TECNOLOGIA & CULTURA

Revista do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
Cefet/RJ | Edição Especial | 2025

TECNOLOGIA & CULTURA

CEFET/RJ - CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA

Expediente

Ministério da Educação (MEC)
Secretaria de Educação Profissional e Tecnológica (Setec)

CEFET/RJ - CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA

TECNOLOGIA & CULTURA - Revista do Cefet/RJ
Edição Especial. 2025
Edição eletrônica disponível em: https://www.cefet-rj.br/index.php/revista-tecnologia-cultura
Av. Maracanã, 229 - Rio de Janeiro/RJ
CEP 20.271-110
Telefone geral: (21) 2566-3022 r. 3160
Telefax: (21) 2284-6021
http://www.cefet-rj.br
E-mail: revista@cefet-rj.br

Diretor-geral
Mauricio Saldanha Motta

Vice-diretor
Gisele Maria Ribeiro Vieira

Diretora de Ensino
Dayse Haime Pastore

Diretor de Pesquisa e Pós-graduação
Ronney Arismel Mancebo Boloy

Diretora de Gestão Estratégica
Diego Carvalho

Diretora de Extensão
Renata da Silva Moura

Diretor de Administração e Planejamento
Bianca de França Tempone Felga de Moraes

ISSN 1414-8498

Editoria
Taís Conceição dos Santos

Biblioteca Central
Mariana de Oliveira Caruso Carvalho

Projeto Gráfico/Diagramação
Divisão de Programação Visual (DPROV)

Tecnologia & Cultura. _ Edição Especial 2025) -
Rio de Janeiro : Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2025.
v. : il.; 28 cms.

Semestral
ISSN 1414-8498

I. Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

SUMÁRIO

Apresentação

O Workshop-Escola de Computação e Informação Quântica (WECIQ) é um evento nacional que reúne palestrantes nacionais e internacionais e contempla as áreas de Aplicações da Computação Quântica na Indústria, Computação Quântica e Grafos, Comunicação e Informação Quânticas, Internet Quântica e Aprendizado Quântico de Máquinas.

Seu público-alvo inclui pesquisadores, professores, estudantes de graduação e pós-graduação em Computação, Engenharia Elétrica, Física e Matemática, bem como profissionais e demais interessados em compreender as aplicações da Mecânica Quântica na computação, comunicação e segurança da informação.

O evento tem como objetivos principais: (1) fomentar o desenvolvimento da informação, computação e comunicação quântica no Brasil; (2) ampliar o intercâmbio de ideias entre especialistas, estudantes e profissionais da área; (3) fortalecer a integração entre universidades, centros de pesquisa e empresas; (4) divulgar a produção técnico-científica nacional; e (5) oferecer minicursos voltados tanto para o público especializado quanto para o público geral.

Nesta edição especial da revista Tecnologia & Cultura, são apresentados os principais trabalhos completos oriundos das sessões de comunicação oral do evento. Esperamos que este material sirva como fonte de inspiração, conhecimento e aprofundamento para a comunidade acadêmica e profissional dedicada à computação, comunicação e informação quântica.

A Comissão Organizadora

7º WECIQ

Análise do Desempenho de Geradores Quânticos de Números Aleatórios usando a Disentropia

S. T. de Oliveira, G. L. de Oliveira e R. V. Ramos

***Resumo* — Neste trabalho utilizamos a disentropia da autocorrelação, uma medida de aleatoriedade baseada na função W_q de Lambert-Tsallis, para medir a aleatoriedade de sequências binárias geradas por geradores quânticos de números aleatórios. É mostrado, desta forma, que a disentropia da autocorrelação pode ser utilizada para analisar o desempenho de geradores quânticos de números aleatórios, sendo sua principal vantagem a facilidade de cálculo quando comparada com testes de aleatoriedade do NIST.**

Palavras-Chave* — *Gerador quântico de números aleatórios, aleatoriedade, disentropia, função W_q de Lambert-Tsallis.

***Abstract* — In this work we use the disentropy of the autocorrelation, a randomness measure based on the Lambert-Tsallis W_q function, to measure the randomness of the binary sequences generated by quantum random number generators. Thus, it is shown the disentropy of the autocorrelation can be used to analyze the performance of quantum random number generators, being the easiness of calculation its main advantage when compared to NIST's randomness tests.**

Keywords* — *Quantum random number generator, randomness, disentropy, Lambert-Tsallis W_q function.

I. INTRODUÇÃO

No mundo moderno a análise de dados é uma tarefa crucial que dá suporte à diversas atividades humanas. Por exemplo, dados médicos, sismológicos, econômicos, astronômicos, dentre outros. Em todos esses casos, de forma intencional ou não, os dados estão sempre contaminados com algum tipo de ruído, o que confere alguma aleatoriedade aos dados. A análise desses dados cada vez mais tem sido feita através do uso de algoritmos de aprendizagem profunda de máquina, como redes neurais multicamadas. Desta forma, quando dados reais não são utilizados, o treinamento de algoritmos de aprendizagem de máquina deve contar com uma fonte de ruído que simule os ruídos presentes em dados reais. Isso aumenta a robustez do algoritmo em questão. Portanto, uma fonte de aleatoriedade se faz importante para o treinamento de algoritmos de aprendizagem de máquina. Outra aplicação importante da aleatoriedade surge em protocolos de criptografia. Um grande número deles exige a geração de dados, no caso em questão bits, aleatórios, para garantir a segurança dos protocolos. Pode-se ainda citar a importância da aleatoriedade em loterias e jogos, dentre outros. Portanto, geradores de aleatoriedade, ou fontes de entropia como também são chamados, são importantes e precisam ser corretamente projetados. Basicamente, há dois tipos de geradores de aleatoriedade: geradores pseudoaleatórios, baseados em software e geradores verdadeiramente aleatórios, baseados em propriedades físicas. O primeiro tipo apresenta memória e, portanto, não são completamente confiáveis. Por outro lado, geradores verdadeiramente aleatórios não possuem nenhuma memória e são preferidos em atividades como segurança de dados [1,2]. Nesta classe se destacam os geradores quânticos de números aleatório (QRNG – *quantum random number generators*), cuja aleatoriedade é baseada em alguma propriedade quântica de um sistema físico, como polarização de fótons, ruído de fase em fontes ópticas não coerentes, flutuações do vácuo, dentre outros [3-13].

Em QRNGs reais, os dispositivos utilizados em sua construção podem apresentar imperfeições que podem afetar a qualidade da aleatoriedade gerada. Por exemplo, detectores de fótons com valores diferentes de eficiência quântica e de contagem de escuro ou divisores de feixes que não são perfeitamente balanceados. Por isso, a aleatoriedade produzida deve ser testada para garantir o bom desempenho do QRNG construído. Comumente, testes de aleatoriedade do NIST são utilizados para certificar a aleatoriedade de um QRNG.

Por outro lado, uma importante ferramenta matemática utilizada na análise de aleatoriedade de sinais é a função de autocorrelação (FAC). Para um sinal contínuo $s(t)$ a FAC é definida como sendo

$$R(\tau)=\int_{-\infty}^{\infty} s(t)s^{*}(t-\tau)\,dt\,, \tag{1}$$

enquanto que para um sinal discreto s_t a FAC no atraso k é definida como sendo

$$r_k=\frac{E\left[(s_t-\bar{s})(s_{t+k}-\bar{s})\right]}{\sigma_s^2}=\frac{1}{N}\frac{\sum_{t=1}^{N-k}(s_t-\bar{s})(s_{t+k}-\bar{s})}{\frac{1}{N}\sum_{t=1}^{N}(s_t-\bar{s})^2}. \tag{2}$$

Em (2) $\bar{s}$ e σ_s^2 são, respectivamente, o valor médio e a variância de s_t. Como é usual, o símbolo E indica o valor esperado. Basicamente, a FAC mostra a similaridade de uma função (sinal) com uma versão atrasada(o) da(o) mesma(o). Quanto maior a aleatoriedade de um sinal, mais a FAC deste sinal se aproxima de uma função delta. Uma medida de aleatoriedade baseada na FAC foi recentemente proposta, chamada de disentropia da autocorrelação [14] e utilizada para aumentar a segurança de distribuição quântica de chaves [15], analisar o desempenho de computador quântico baseado em amostragem

Sergio Tahim de Oliveira, , Departamento de Engenharia de Teleinformática, UFC, Fortaleza-Ce e-mail: sergiotahim@gmail.com; Glaucionor Lima de Oliveira, Departamento de Engenharia de Teleinformática, UFC, Fortaleza-Ce email: glau@ifce.edu.br; Rubens Viana Ramos, Departamento de Engenharia de Teleinformática, UFC, Fortaleza-Ce, e-mail: rubens.ramos@ufc.br.

de Gaussiana de bósons [16], e na análise de sinais astronômicos [17]. Nesta direção, o presente trabalho apresenta o uso da disentropia da autocorrelação na análise da aleatoriedade gerada por QRNGs.

Este trabalhado está dividido da seguinte forma: na Seção II é feita uma revisão da função de Lambert-Tsallis e da disentropia; Na Seção III, usando a disentropia da autocorrelação, é feita a análise da aleatoriedade de sequências binárias geradas por geradores pseudoaleatórios e por QRNGs. Por fim as conclusões são descritas na Seção IV.

II. A Função Wq de Lambert-Tsallis e a Disentropia

A função $W(z)$ de Lambert é uma função matemática elementar que tem sido utilizada em diferentes áreas da matemática, física e ciência da computação [18-21]. A função W de Lambert é definida como sendo a solução de

$$W(z)e^{W(z)} = z. \tag{3}$$

Tomando o logaritmo em ambos os lados de (3) obtem-se

$$\log(z) = W(z) + \log\left[W(z)\right]. \tag{4}$$

Portanto, a entropia pode ser escrita como

$$S = -\sum_i p_i \log(p_i) = -\sum_i p_i W(p_i) - \sum_i p_i \log\left[W(p_i)\right]. \tag{5}$$

na qual $\{p_1, p_2, \ldots, p_n\}$ é uma distribuição discreta de probabilidade. O termo

$$D = \sum_i p_i W(p_i) \tag{6}$$

é chamado de disentropia. Quando a disentropia é mínima a entropia é máxima e vice-versa. A eq. (6) é a disentropia relacionada à entropia de Boltzmann-Gibbs. A eq. (3) pode ser modificada para a forma

$$R_2(z)2^{R_2(z)} = z, \tag{7}$$

cuja solução é

$$R_2(z) = \log_2(e)W\left(\frac{z}{\log_2(e)}\right) \tag{8}$$

e, neste caso, a disentropia relacionada à entropia de Shannon é

$$D = \sum_i p_i R_2(p_i). \tag{9}$$

A eq. (3) pode ainda ser modificada fazendo a troca da função exponencial pela função q-exponencial de Tsallis [22]:

$$W_q(z)e_q^{W_q(z)} = z, \tag{10}$$

sendo a função q-exponential de Tsallis dada por

$$e_q^z = \begin{cases} e^z & q = 1 \\ \left[1+(1-q)z\right]^{1/(1-q)} & q \neq 1 \;\&\; 1+(1-q)z \geq 0 \\ 0 & q \neq 1 \;\&\; 1+(1-q)z < 0 \end{cases}. \tag{11}$$

A função $W_q(z)$ é chamada de função de Lambert-Tsallis [23]. É possível encontrar a forma analítica de W_q para alguns pouco casos sendo a mais simples delas obtida quando $q = 2$: $W_2(z) = z/(1+z)$ definida for $z > -1$. Por outro lado, para $q = 3/2$ tem-se:

$$W_{3/2}^{\pm}(z) = \frac{2(z+1) \pm 2\sqrt{2z+1}}{z}, \quad z > -1/2 \tag{12}$$

Em geral, $W_q(z)$ tem que ser calculada numericamente. Por exemplo, pode-se usar o método de Halley para calcular $W_q(z)$:

$$w_q(j+1) = w_q(j) - \frac{A}{B - \frac{AC}{2B}} \tag{13}$$

$$A = w_q(j)e_q^{w_q(j)} - z \tag{14}$$

$$B = e_q^{w_q(j)} + w_q(j)e_{2-\frac{1}{q}}^{qw_q(j)} \tag{15}$$

$$C = 2e_{2-\frac{1}{q}}^{qw_q(j)} + \frac{w_q(j)}{q}e_{\frac{3-2/q}{2-1/q}}^{(2q-1)w_q(j)}. \tag{16}$$

Por exemplo, na Fig. 1 pode-se ver a curva de $W_{3/2}(z)$ versus z.

Pode-se mostrar que o ponto de ramificação de $W_q(z)$ ($dW_q(z)/dz = \infty$) é dado pelo ponto $\{z_b = \exp_q(q\text{-}2)/(q\text{-}2), W_q(z_b) = 1/(q\text{-}2)\}$. Mais detalhes sobre a função W_q podem ser encontrados em [24-28].

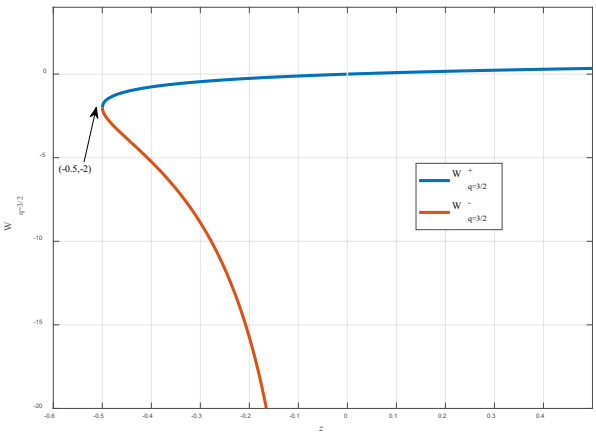


Fig.1. $W_{q=3/2}(z)$ versus z.

Tomando a função q-logaritmo em ambos os lados da eq. (10), obtém-se

$$\log_q(z) = W_q(z) + \log_q\left[W_q(z)\right] + (1-q)W_q(z)\log_q\left[W_q(z)\right]. \quad (17)$$

na qual

$$\log_q(z) = \begin{cases} \log(z) & x>0 \,\&\, q=1 \\ \dfrac{x^{(1-q)}-1}{1-q} & x>0 \,\&\, q\neq 1 \\ \text{indefinida} & x\leq 0 \end{cases}. \quad (18)$$

Portanto, a q-entropia de Tsallis [22] pode ser escrita como

$$S_T = -\sum_i p_i^q \log_q(p_i) = -\sum_i p_i^q W_q(p_i) - \sum_i p_i^q \log_q\left[W_q(p_i)\right] - (1-q)\sum_i p_i^q W_q(p_i)\log_q\left[W_q(p_i)\right]. \quad (19)$$

O termo

$$D_q = \sum_i p_i^q W_q(p_i) \quad (20)$$

é a disentropia relacionada à entropia de Tsallis.

III. A Disentropia da Função de Autocorrelação e a Aleatoriedade de Sequências Binárias

Aproveitando as propriedades da FAC e da disentropia, a medida de aleatoriedade chamada disentropia da autocorrelação foi recentemente proposta em [14]. Para um sinal discreto s_t ela é dada simplesmente por $D_{q=2}(R(s_t))$, ou seja,

$$D_2 = \sum_{n=1}^{N} \frac{r_n^3}{r_n+1}, \quad (21)$$

na qual r_n é o n-ésimo valor da FAC de s_t. Para um sinal com máxima aleatoriedade, como um ruído branco, a FAC é uma função delta e o valor de D_2 em (21) é igual a 0.5. Portanto, quanto mais próximo de 0.5, mais aleatório é o sinal considerado. A Tabela 1 e a Fig. 2 a seguir mostram cinco exemplo, nos quais um sinal quadrado é contaminado com ruído Gaussiano cada vez mais intenso.

Tabela I – Sinal quadrado com diferentes níveis de ruído e seus respectivos valores de aleatoriedade calculados com uso da disentropia da autocorrelação: $N(x,y)$ – ruído Gaussiano com média igual a x e variância igual a y.

	$s(t)$ = onda quadrada	Aleatoriedade - $D_2(R(s(t)))$
I	$s(t)$	-29054.1802
II	4+ $s(t)$ +N (0,0.25)	-7521.519
III	8 + $s(t)$+N (0,0.5)	-830.401
IV	15 + $s(t)$+N (0,1)	-15.482
V	25+ N (0,1)	0.49995

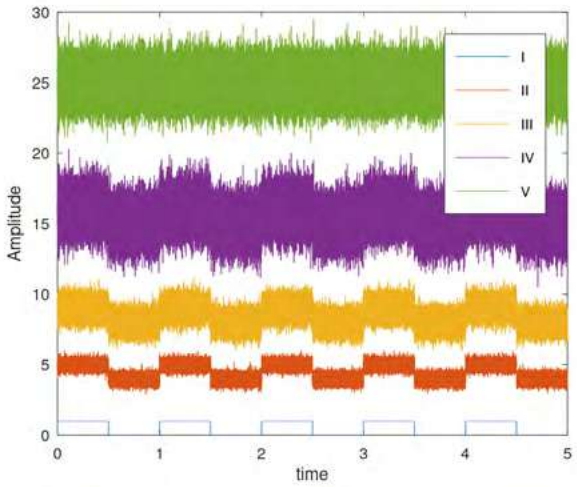


Fig. 2. Onda quadrada contaminada com ruído Gaussiano de diferentes valores de variância (V – ruído Gaussiano puro).

Como pode ser observado na Fig. 2 e na Tabela I, quanto maior o nível de ruído (valor da variância) mais próximo do valor 0.5 estará a disentropia da autocorrelação.

Para calcular a aleatoriedade de sequências binárias, deve-se lembrar que a FAC capta a presença de memória no sinal. Portanto, sequências de bits obtidas por um processo físico sem memória serão consideradas pela FAC como sendo maximamente aleatório, mesmo que o número total de '0's e '1's sejam muito diferentes. Em outras palavras, uma sequência binária obtida com uso de uma moeda honesta e outra sequência binária obtida com uso de uma moeda viciada, vão possuir a mesma FAC, uma função delta e, portanto, o mesmo valor de D_2 = 0.5. Para evitar este problema, somamos uma função determinística f (neste trabalho uma função seno) à sequência binária b a ser testada. Neste caso, quanto maior a aleatoriedade da sequência b maior o apagamento da memória da função f e, portanto, menor o valor da disentropia da autocorrelação do conjunto $b+f$.

Usando o gerador pseudoaleatório do software OCTAVE, 5.000 sequências de 1.000.000 de bits foram geradas, e classificadas como b_1, b_2 e b_3. Para a sequências b_1 os bits '0' e '1' foram escolhidos com a mesma probabilidade $p_0 = p_1 = 0.5$. Para sequências b_2 os bits '0' e '1' foram escolhidos com probabilidades $p_0 = 0.7$ e $p_1 = 0.3$. Por fim, para sequências b_3 os bits '0' e '1' foram escolhidos com probabilidades $p_0 = 0.9$ e $p_1 = 0.1$. Os histogramas dos valores do cálculo da aleatoriedade usando a disentropia para estas sequências, podem ser vistos na Fig. 3.

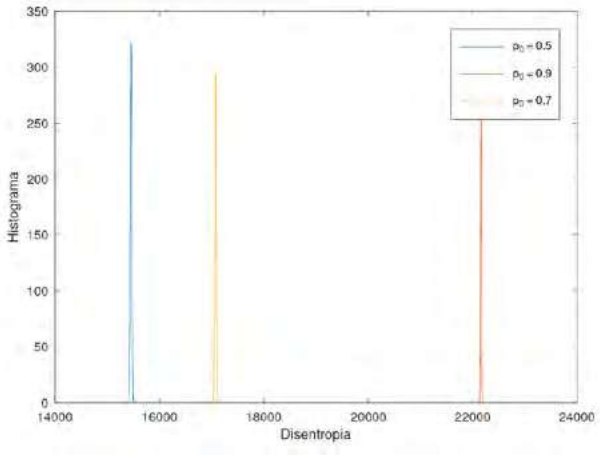


Fig. 3. Histogramas das aleatoriedades calculadas pela disentropia da autocorrelação das sequências b_1, b_2 e b_3 (5.000 sequências de 1000.000 de bits) obtidas com um gerador pseudoaleatório.

Como pode ser observado na Fig. 3, as sequências b_1 (maximamente aleatória), b_2 (com viés) e b_3 (com forte viés) são completamente distinguidas pela disentropia da autocorrelação. Os valores médios da disentropia para as sequências b_1, b_2 e b_3 são, respectivamente: 15462.1837 (p_0 = 0.5), 17074.1018 (p_0 = 0.7) e 22160.5959 (p_0 = 0.9). A disentropia da função f é 22842.7428.

A Fig. 4 mostra o mesmo para 1.000 sequências binárias de 150.000 bits.

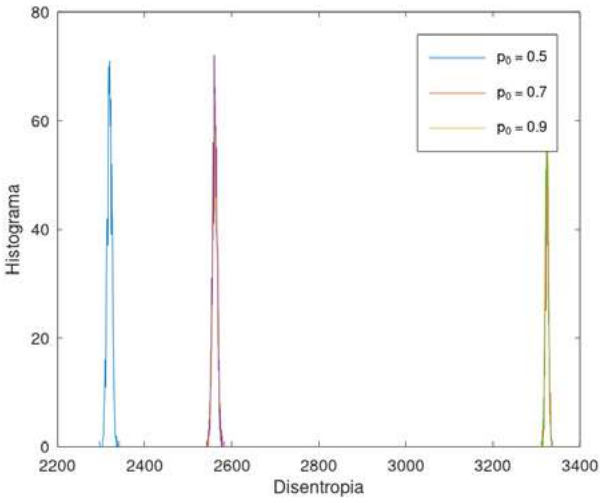


Fig. 4. Histogramas das aleatoriedades calculadas pela disentropia da autocorrelação das sequências b_1, b_2 e b_3 (1.000 sequências de 150.000 bits) obtidas com um gerador pseudoaleatório.

Os valores médios da disentropia para as sequências b_1, b_2 e b_3 são, respectivamente: 2319.3898 (p_0 = 0.5), 2561.2481 (p_0 = 0.7) e 3324.32 (p_0 = 0.9). A disentropia da função f neste caso é 3426.4097.

Usando dez sequências de 150.000 bits, geradas por um QRNG real obtidos no sítio http://qrng.ethz.ch/live/, os valores da disentropia obtidos são mostrados na Tabela II.

Tabela II. Disentropia da autocorrelação de dez sequências binárias de 150.000 bits cada. Sequências obtidas no site **http://qrng.ethz.ch/live/.**

QRNG				
2322.8445	2309.9544	2318.5914	2320.5676	2315.9882
2318.5725	2320.5387	2325.5954	2312.7469	2306.9587

IV. CONCLUSÕES

A disentropia da autocorrelação é uma medida confiável de aleatoriedade e pode ser utilizada na análise de desempenho de geradores quânticos de números aleatórios. Sua grande vantagem é o fácil e rápido cálculo, além de permitir fazer uma gradação de diferentes níveis de aleatoriedade. As sequências binárias reais obtidas a partir do sítio disponível na internet apresentaram valores de disentropia compatíveis com os valores esperados para uma sequência binária maximamente aleatória ($p_0 = p_1 = 0.5$). Portanto, a disentropia da autocorrelação é uma ferramenta que pode ser utilizada para detectar, de forma não invasiva, o mau comportamento de dispositivos utilizados no QRNG que, com o envelhecimento, podem mudar suas características e introduzir um viés na sequência binária gerada.

AGRADECIMENTOS

Este trabalho foi parcialmente financiado pelas agências CNPq (309374/2021-9) e CAPES (001).

REFERÊNCIAS

[1] J. Bouda, M. Pivoluska, M. Plesch, C. Wilmott, "Weak randomness seriously limits the security of quantum key distribution", *Phys. Rev. A*, v. 86, pp. 062308/1-5, 2012.
[2] H. W. Li, Z. Q. Yin, S. Wang, Y. J. Qian, W. Chen, G. C. Guo, Z. F. Han, "Randomness determines practical security of BB84 quantum key distribution", *Sci. Rep.*, v. 5, pp. 16200/1-8, 2015.
[3] R. Serrano, C. Duran, T.-T. Hoang, M. Sarmiento, K.-D. Nguyen, A. Tsukamoto, K. Suzaki, "A fully digital true random number generator with entropy source based in frequency collapse", *IEEE Access*, v. 9, pp. 105748-105755, 2021.
[4] F. Monet, J.-S. Boisvert, R. Kashyap, "A simple high-speed random number generator with minimal post-processing using a random Raman fiber laser", *Sci. Rep.*, v. 11, pp. 13182/1-8, 2021.
[5] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, "A fast and compact quantum random number generator", *Rev. Sci. Instrum*. V. 71, no. 4, pp. 1675-1680, 2000.
[6] T. Gehring, C. Lupo, A. Kordts, D. S. Nikolic, N. Jain, T. Rydberg, T. B. Pedersen, S. Pirandola, U. L. Andersen, "Homodyne-based quantum random number generator at 2.9 Gbps secure against quantum side-information", *Nature Comm.*, v. 12, pp. 605/1-11, 2021.
[7] Y. Zhang, H.-P. Lo, A. Mink, T. Ikuta, T. Honjo, H. Takesue, W. J. Munro, "A simple low-latency real-time certifiable quantum random number generator", *Nature Comm.*, v. 12, pp. 1056/1-8, 2021.
[8] B. Qi, "True randomness from an incoherent source", *Rev. Sci. Instrum.*, v. 88, pp. 113101/1-6, 2017.
[9] J. Liu, J. Yang, Z. Li, Q. Su, W. Huang, B. Xu, H. Guo, "117 Gbits/s quantum random number generation with simple structure", *IEEE Photon. Tech. Lett.*, v. 29, no. 3, pp. 283-286, 2017.
[10]B. Bai, J. Huang, G.-R. Qiao, Y.-Q. Nie, W. Tang, T. Chu, J. Zhang, J.-W. Pan, "18.8 Gbps real-time quantum random number generator with a photonic integrated chip", *Appl. Phys. Lett.*, v. 118, pp. 264001, 2021.
[11]C. Bruynsteen, T. Gehring, C. Lupo, J. Bauwelinck, X. Yin, "100-Gbit/s Integrated quantum random number generator based on vacuum fluctuations", *PRX Quantum*, v. 4, pp. 010330/1-11, 2023.

[12]Y.-X. Liu, K.-X. Huang, Y.-M. Bai, Z. Yang, J.-L. Li, “A High-Randomness and High-Stability Electronic Quantum Random Number Generator without Post Processing”, *Chinese Phys. Lett.*, v. 40, pp. 070303/1-5, 2023.
[13]E. de J. L. Soares, F. A. Mendonca and R. V. Ramos, "Quantum Random Number Generator Using Only One Single-Photon Detector," *IEEE Phot. Tech. Lett.*, v. 26, no. 9, pp. 851-853, 2014.
[14] R. V. Ramos, “Estimation of the Randomness of Continuous and Discrete Signals Using the Disentropy of the Autocorrelation”, *SN Compt. Sci.*, v. 2, pp. 254/1-9, 2021.
[15]G. S. Castro, R. V. Ramos, “Enhancing eavesdropping detection in quantum key distribution using disentropy measure of randomness” *Quant. Inf. Process*, v. 21, pp. 79/1-10, 2022.
[16]F.V. Mendes, C. Lima, R. V. Ramos, “Applications of the Lambert–Tsallis Wq function in quantum photonic Gaussian boson sampling”. *Quant. Inf. Process.*, v. 21, pp. 215, 2022.
[17]F. J. L. de Almeida, R. V. Ramos, Disentropy in astronomy, *Eur. Phys. J. Plus*, v. 138, pp. 20/1-10, 2023.
[18]R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey and D. E. Knuth, “On the Lambert W function”, *Adv. in Comp. Math.*, v. 5, pp. 329 – 359, 1996.
[19]S. R. Valluri, D. J. Jeffrey, R. M. Corless, “Some applications of the Lambert W function to Physics”, *Can. J. of Phys.*, v. 78, no. 9, pp. 823-831, 2000.
[20]F. C.-Blondeau and A. Monir, “Numerical evaluation of the Lambert W function and application to generation of generalized Gaussian noise with exponent ½”, *IEEE Trans. on Signal Processing*, v. 50, no. 9, pp. 2160-2165, 2002.
[21]K. Roberts, S. R. Valluri, Tutorial: “The quantum finite square well and the Lambert W function”, *Can. J. of Phys.*, v. 95, no. 2, pp. 105-110, 2017.
[22]C. Tsallis, “Possible generalization of Boltzmann-Gibbs statistics”, *J. Stat. Phys.*, v. 52, pp. 479, 1988.
[23]G. B. da Silva, R. V. Ramos, “The Lambert–Tsallis Wq function”, *Physica A*, v. 525, pp. 164, 2019.
[24] J. S. de Andrade, K. Z. Nobrega, R. V. Ramos, “Analytical solution of the current-voltage characteristics of circuits with power-law dependence of the current on the applied voltage using the Wq de Lambert-Tsallis function”, *IEEE Trans. Circuits Syst. II Express Briefs*, v. 69, n2022.
[25]J. R. da Silva, R. V. Ramos, “Applications of the Lambert–Tsallis Function in X-Ray Free Electron Laser”, *IEEE Trans. on Plasma Sci.*, v. 50, no. 10, pp. 3578-3582, 2022.
[26]R. L. C. Damasceno, J. S. Andrade, R.V. Ramos, “Applications of the Lambert–Tsallis Wq function in QKD”, *J. Opt. Soc. Am. B*, v. 40, no. 9, pp. 2280-2286, 2023.
[27]R. V. Ramos, “The Rq,Q function and the q-diode”, Physica A, no. 556, p. 12485/1-9, 2020.
[28]R. V. Ramos, “Disentropy of the Wigner function”, J. of Opt. Soc. of Am. B, 36, 8 2244, 2019.

7º WECIQ

Simulação do Impacto do Espalhamento Raman Espontâneo na Taxa de Transmissão em Sistemas de QKD em Redes Ópticas Passivas

J. S. de Andrade e R. V. Ramos

***Resumo* — Neste trabalho utilizamos a função W_q de *Lambert-Tsallis* e simulações numéricas para analisar a taxa de transmissão de bits seguros de um protocolo de QKD em uma rede óptica passiva, com dados clássicos e quânticos coexistindo, quando a potência óptica dos sinais clássicos e o número de usuários variam. Observamos que um aumento da potência óptica dos sinais clássicos tem que ser compensada com o aumento do número de usuários para que a distribuição quântica de chaves possa ser realizada.**

Palavras-Chave* — *Distribuição quântica de chaves, redes ópticas, espalhamento Raman espontâneo.

***Abstract* — In this work we use the *Lambert-Tsallis* W_q function and numerical simulations to analyze the secure bit rate of a QKD protocol in a passive optical network, with coexistence of classical and quantum data, when the optical power of classical data and the number of users change. We observe that an increase of the classical optical power must be compensated by an increase of the number of users to permit the realization of QKD.**

Keywords*— *Quantum key distribution, optical networks, spontaneous Raman scattering.

I. INTRODUÇÃO

A tecnologia quântica, já comercialmente disponível, conhecida como Distribuição Quântica de Chaves (*quantum key distribution* - QKD), possibilita o estabelecimento seguro de uma chave, uma sequência aleatória de bits, entre usuários espacialmente distantes conhecidos como Alice e Bob [1-12]. A chave distribuída é utilizada em protocolos de criptografia, permitindo a transmissão segura de informações entre Alice e Bob através do protocolo de chave simétrica *one-time pad*, no qual a chave é utilizada uma única vez e a codificação (decodificação) consiste na operação lógica XOR (ou-exclusivo) entre chave e mensagem (texto cifrado). Entretanto, a implementação de QKD em redes ópticas reais enfrenta desafios consideráveis. Um desses desafios é a coexistência de dados clássicos e quânticos na mesma rede óptica [13-18]. Neste caso em que dados clássicos (pulsos ópticos intensos) e quânticos (pulsos ópticos fracos) percorrem a mesma fibra óptica, o espalhamento Raman espontâneo surge como o principal ruído, limitando o comprimento máximo do canal. Note-se que, para ser amplamente adotada, uma configuração óptica de QKD precisa ser flexível, reconfigurável e escalável. Essas características podem ser alcançadas ao se realizar QKD em redes ópticas já instaladas, integrando dados quânticos e clássicos na mesma fibra óptica. No entanto, devido à substancial diferença de potência óptica utilizada pelos protocolos de comunicação quântica e clássica, a coexistência de sinais de dados quânticos e clássicos na mesma fibra óptica pode prejudicar o protocolo quântico.

A presença de amplificadores ópticos na rede óptica impede a realização de QKD na janela de 1550 nm devido ao intenso ruído de emissão espontânea amplificada nessa parte do espectro. Em tal situação, o protocolo de QKD poderia ser implementado em 1310 nm. Entretanto, devido à elevada perda óptica na fibra nesse comprimento de onda, a distância entre o transmissor e o receptor é severamente limitada. Para alcançar distâncias maiores, é necessário executar o protocolo de QKD em 1550 nm e os dados clássicos em outro comprimento de onda, como 1310 nm. A colocação de dados quânticos e clássicos na janela de 1550 nm é possível se uma fibra multinúcleos for utilizada, embora essas fibras ainda sejam caras e apresentem diafonia entre diferentes núcleos, o que deve ser considerado. Outra opção é o uso de redes ópticas passivas (*passive optical network* - PON), onde amplificadores ópticos não são empregados. Em todos esses casos, o *crosstalk* linear proveniente de (de)multiplexadores e filtros ópticos imperfeitos, juntamente com vários efeitos não lineares, como mistura de quatro ondas (*four-wave mixing* - FWM) e os espalhamentos de *Brillouin*, *Rayleigh* e *Raman*, dificultam a integração eficiente de dados quânticos e clássicos [13-15].

Embora seja possível evitar a FWM e os espalhamentos *Brillouin* e *Rayleigh* alocando o canal quântico de forma espectralmente distante dos canais de dados clássicos, o espalhamento *Raman* espontâneo (*Spontaneous Raman Scattering* - SRS) apresenta uma grande largura espectral e, portanto, pode não ser completamente evitado. Assim, o desafio mais significativo na coexistência de dados quânticos e clássicos na mesma fibra óptica é a geração de falsas contagens causadas pelo espalhamento *Raman* espontâneo [13,16-18]. O SRS aumenta a taxa de erro quântico (*quantum bit error rate* - QBER), reduzindo a taxa de transmissão de bits seguros da chave. Desta forma, um projeto cuidadoso de redes ópticas que suportem serviços quânticos e clássicos deve considerar a geração do SRS e o impacto do mesmo em protocolos de QKD.

Neste contexto, o presente trabalho utiliza a função de Lambert-Tsallis e simulações numéricas na análise da taxa de transmissão de bits seguros do protocolo de QKD BB84 com dois estados iscas, em uma topologia de rede óptica passiva chamada configuração *downstream* [13,17,18] com coexistência

Joacir Soares de Andrade, Departamento de Telemática, IFCE, Fortaleza-Ce, e-mail: joacirandrade@ppget.ifce.edu.br; Rubens Viana Ramos, Departamento de Engenharia de Teleinformática, UFC, Fortaleza-Ce, e-mail: rubens.ramos@ufc.br.

de dados quânticos e clássicos, quando a potência óptica do sinal clássico e o número de usuários variam.

Este trabalho está dividido da seguinte forma: na Seção 2 é feita uma revisão da função de *Lambert-Tsallis*; Na Seção 3 o efeito do SRS no protocolo de QKD é descrito e as simulações numéricas são realizadas. Por fim as conclusões são descritas na Seção 4.

II. A Função de Lambert-Tsallis

A função W_q de *Lambert-Tsallis* pode ser usada para resolver algumas equações não lineares nas quais as variáveis independente e dependente estão relacionadas por uma lei de potência. Ela é definida como a solução da equação [19]

$$W_q(z)\, e_q^{W_q(z)} = z \tag{1}$$

Em (1), a q-exponencial de *Tsallis* é dada por [20-21]

$$e_q^{\,z} = \left[1+(1-q)z\right]^{1/(1-q)} \quad \text{for } q \neq 1 \;\&\; 1+(1-q)z \geq 0 \tag{2}$$

Além disso, $e_q^z = 0$ quando $1+(1-q)z < 0$. Obviamente, $\lim_{q\to 1} e_q^z = e^z$, portanto, $\lim_{q\to 1} W_q(z) = W(z)$, sendo $W(z)$ a função de Lambert [22-23]. É possível encontrar a forma analítica de W_q para alguns poucos casos [19], sendo a mais simples delas obtida quando $q = 2$, $W_2(z) = z/(1+z)$ definida para $z > -1$. No caso geral, $W_q(z)$ tem que ser calculada numericamente. Por exemplo, usando o método de *Halley* para calcular $W_q(z)$ tem-se:

$$w_q(j+1) = w_q(j) - \frac{A}{B - \dfrac{AC}{2B}} \tag{3}$$

$$A = w_q(j)\, e_q^{w_q(j)} - z \tag{4}$$

$$B = e_q^{w_q(j)} + w_q(j)\, e_{2-\frac{1}{q}}^{q w_q(j)} \tag{5}$$

$$C = 2 e_{2-\frac{1}{q}}^{q w_q(j)} + \frac{w_q(j)}{q}\, e_{\frac{3-2/q}{2-1/q}}^{(2q-1) w_q(j)}. \tag{6}$$

Na Fig. 1 podem-se ver as curvas de $W_{3/4}(z)$ ($q < 1$), $W_{5/4}(z)$ ($q > 1$) e $W(z)$ ($q = 1$) versus z. Pode-se mostrar que o ponto de ramificação de $W_q(z)$ (ponto onde $dW_q(z)/dz = \infty$) é dado pelos pontos $z_b = \exp_q(q\text{-}2)/(q\text{-}2)$, $W_q(z_b) = 1/(q\text{-}2)$. Mais detalhes sobre a função W_q podem ser encontrados em [24-27].

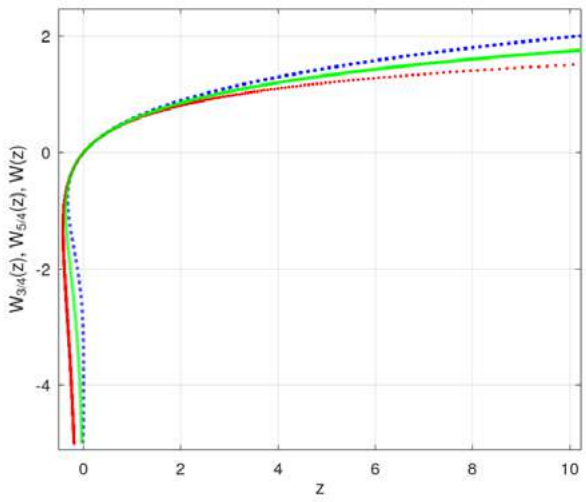


Fig. 1. $W_{3/4}(z)$ (linha pontilhada azul), $W_{5/4}(z)$ (linha pontilhada vermelha) e $W(z)$ (linha contínua verde) versus z.

III. Impacto do SRS em QKD em Redes PON

Na comunicação clássica, o sinal *downstream* do terminal de linha óptica (*optical line terminal* - OLT) no escritório central é enviado a todos os usuários por um divisor de feixes, e o sinal upstream de apenas uma unidade de rede óptica (*Optical network unit* - ONU) colocada no nó do usuário é transmitido para OLT em cada intervalo de tempo [28]. No caso de uma rede de acesso quântico (*quantum access network* - QAN) usando a estrutura PON, pode-se ter duas configurações: 1) *Downstream*, em que o receptor QKD é colocado nos nós dos usuários [29] e 2) *upstream*, em que o transmissor QKD é colocado nos nós dos usuários [17]. Como a configuração *downstream* tem menos ruído SRS do que a configuração *upstream* e a taxa de geração segura não depende do número de usuários, nesta trabalho consideraremos apenas a configuração *downstream* $1 \times N$: uma OLT com transmissor QKD e N ONU's, cada uma com seu receptor QKD. Entre a OLT e a ONU estão a fibra alimentadora, com comprimento L_F, um divisor de potência passivo $1 \times N$ (alimentador único) e as fibras de descida com comprimento L_D << L_F. Assim como em [13], consideramos que os fótons de ruído SRS são gerados principalmente pelo sinal OLT. Além disso, consideramos que os dados clássicos são transmitidos em 1310 nm enquanto que os dados quânticos são transmitidos em 1550 nm. O sinal OLT gera fótons de ruído SRS tanto na fibra alimentadora (*feed* - S_F) quanto na fibra de descida (*drop* - S_D). Os valores de S_F e S_D são dados pelas equações [16]

$$S_F = \left\{ P\beta \Big/ \left[N(\alpha_q - \alpha_c) \right] \right\} \left(e^{-\alpha_c L_F} - e^{-\alpha_q L_F} \right) e^{-\alpha_q L_D} \tag{7}$$

$$S_D = \left\{ P\beta \Big/ \left[N(\alpha_q - \alpha_c) \right] \right\} \left(e^{-\alpha_c L_D} - e^{-\alpha_q L_D} \right) e^{-\alpha_c L_F} \tag{8}$$

nas quais P é a potência de lançamento do sinal OLT, N é a taxa de divisão do divisor de potência, β é o coeficiente SRS, α_c (α_q) é a perda de fibra no comprimento de onda dos dados clássicos (quânticos). Consideramos o protocolo BB84 em QKD com dois

estados isca [30]. A taxa de chave segura do protocolo é limitada inferiormente por

$$R = q\left\{Q_1\left[1-H_2\left(e_1\right)\right]-f_{ec}Q_\mu H_2\left(e_\mu\right)\right\} \quad (9)$$

sendo $q = 1/2$ para o caso do protocolo BB84, f_{ec} é a eficiência da correção de erros, $H_2(x) = -x\log_2(x)-(1-x)\log_2(1-x)$, Q_μ e e_μ são o ganho geral e taxa de erro de bits quânticos (*qubit error rate* – QBER) do estado do sinal, enquanto Q_1 e e_1 são o ganho e QBER dos estados de um fóton do sinal. Assumindo o estado do sinal QKD com número médio de fótons μ e dois estados iscas com números médios de fótons v ($< \mu$) e 0, a eq. (9) é calculada usando [15, 16]

$$Q_{\mu(\nu)} = 1-\left(1-Y_0\right)\exp\left(-\mu\left(\nu\right)\eta_B e^{-\alpha_q L-10\log_{10}(N)}\right) \quad (10)$$

$$e_{\mu(\nu)} = e_d+\left[\left(1/2-e_d\right)Y_0\right]/Q_{\mu(\nu)}, \quad (11)$$

$$Q_1 = \frac{\mu^2 e^{-\mu}}{\mu\nu-\nu^2}\left(Q_\nu e^\nu - Q_\mu e^\mu \frac{\nu^2}{\mu^2}-\frac{\mu^2-\nu^2}{\mu^2}Y_0\right), \quad (12)$$

$$e_1 = \left(e_\nu Q_\nu e^\nu - 0.5Y_0\right)/\left(Y_1\nu\right), \quad (13)$$

$$Y_1 = \frac{\mu}{\mu\nu-\nu^2}\left(Q_\nu e^\nu - Q_\mu e^\mu \frac{\nu^2}{\mu^2}-\frac{\mu^2-\nu^2}{\mu^2}Y_0\right), \quad (14)$$

$$Y_0 = 2p_{dark}+p_R\left(L\right). \quad (15)$$

Em (10)-(11) o comprimento do canal é igual a L e η_b é a eficiência quântica do detector de fótons únicos do receptor. O parâmetro e_d representa os erros de desalinhamento, visibilidade não unitária do interferômetro e modulação imperfeita, p_{dark} é a taxa de contagem de escuro dos detectores de fótons únicos e $p_R(L)$ é probabilidade de haver uma detecção causada por fótons gerados pelo SRS na fibra óptica sendo dada por [18]

$$p_R\left(L\right) = \left\{\left[S_F\left(L\right)+S_D\left(L\right)\right]\Delta f\Delta t\eta_B\right\}/\left(hf\right). \quad (16)$$

Em (16) h é a constante de *Planck*, f é a frequência óptica do sinal quântico, Δf é a largura de banda de recepção do canal quântico e Δt é o intervalo de tempo efetivo em que o detector está apto a ter uma deteção (largura dos pulsos de gatilho do detector de fótons).

A partir de um valor para p_R, utiliza-se (16) para obter o valor de $S_F + S_D$. O valor de $S_F + S_D$, por sua vez, é utilizado em (7) + (8). Por fim, o comprimento do canal, $L = L_D + L_F$, é obtido usando a função W_q de *Lambert-Tsallis* para inverter (7) + (8). O resultado é ($\alpha_q < \alpha_c$) [27]

$$L = \frac{1}{\alpha_q-\alpha_c}\ln\left(\frac{\alpha_q}{\alpha_q-\alpha_c}W_{1-\frac{\alpha_q}{\alpha_c-\alpha_q}}\left[\frac{\alpha_q-\alpha_c}{\alpha_q}\left(-z\right)^{\frac{\alpha_c-\alpha_q}{\alpha_q}}\right]\right) \quad (17)$$

$$z = \frac{N\left(S_F+S_D\right)\left(\alpha_q-\alpha_c\right)}{P\beta} = \frac{Nhfp_R\left(\alpha_q-\alpha_c\right)}{\Delta f\Delta t\eta_B P\beta}. \quad (18)$$

Usando o ponto de ramificação de W_q na eq. (17) pode-se encontrar a seguinte relação entre P, N e p_R:

$$\frac{Np_R}{P} \geq \frac{\Delta f\Delta t\eta_B\beta}{hf}\frac{\left[1-\left(\alpha_q/\alpha_c\right)\right]}{\left(\alpha_c-\alpha_q\right)}\left(\frac{\alpha_q}{\alpha_c}\right)^{\frac{\alpha_q}{\alpha_c-\alpha_q}}. \quad (19)$$

Em outras palavras, se a eq. (19) não for satisfeita, não será possível encontrar um valor para L. De acordo com (9)-(15), R é máximo quando Y_0 é mínimo (quanto menor o ruído, maior a taxa de transmissão). O valor mínimo de Y_0 é alcançado quando p_R é mínimo e, conforme (19), o valor mínimo de p_R é [27]

$$p_R^{\min} = \frac{P\Delta f\Delta t\eta_B\beta}{Nhf}\frac{\left[1-\left(\alpha_q/\alpha_c\right)\right]}{\left(\alpha_c-\alpha_q\right)}\left(\frac{\alpha_q}{\alpha_c}\right)^{\frac{\alpha_q}{\alpha_c-\alpha_q}}. \quad (20)$$

Portanto, substituindo (20) em (17)-(18), o valor ótimo de L é dado por

$$L^{opt} = \frac{1}{\alpha_q-\alpha_c}\ln\left(\frac{\alpha_q}{\alpha_q-\alpha_c}W_{1-\frac{\alpha_q}{\alpha_c-\alpha_q}}\left[\frac{\alpha_q-\alpha_c}{\alpha_c}\left[\frac{\alpha_c-\alpha_q}{\alpha_c}\right]^{\frac{\alpha_c-\alpha_q}{\alpha_q}}\right]\right). \quad (21)$$

Como se pode notar, quanto menor o valor de α_c maior é o valor de L_{opt}. Isso acontece porque será necessário mais comprimento de fibra para atenuar o sinal clássico (o que diminui o SRS) ao nível aceitável.

Para analisar a variação da taxa de transmissão de bits seguros da chave quando a potência óptica clássica e o número de usuários variam, simulamos numericamente a eq. (9). Em nossa simulação usamos $\alpha_c = 0.48$ dB/km (1310 nm), $\alpha_q = 0.27$ dB/km (1550nm), $\eta_B = 0.15$, $P = \{0.25, 0.5, 0.6\}$ mW, $N = \{4, 8, 16\}$, hf é a energia do fóton em 1550 nm, $p_{dark} = 2\times10^{-7}$, $\Delta t = 1$ ns, $\Delta f = 100$ GHz, $f_{ec} = 1.2$, $\mu = 0.4$, $v = 0.1$, $e_d = 0.02$ e $\beta = 7\times10^{-9}$nm^{-1}. As perdas de inserção dos divisores de potência são 6.2 dB, 9.2 dB e 12.7 dB para 1 × 4, 1 × 8, 1 × 16, respectivamente. Não estamos considerando o efeito de *afterpulsing* nos detectores. Os resultados das simulações podem ser vistos nas Figs. 2 ($P = 0.25$ mW), 3 ($P = 0.5$ mW) e 4 ($P = 0.6$ mW).

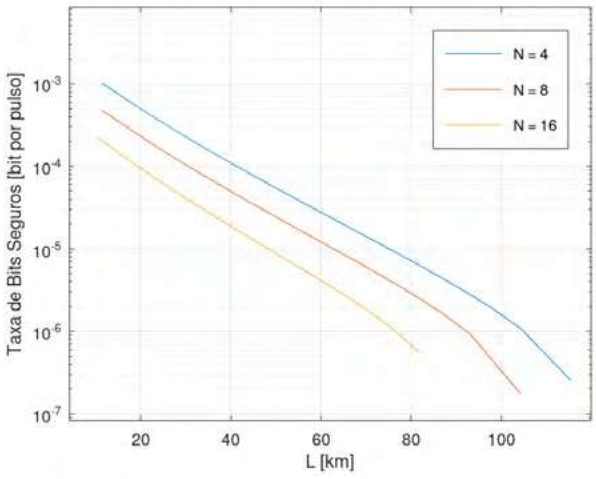


Fig. 2. Taxa de bits seguros versus L (comprimento do canal). $\alpha_c = 0.48$ dB/km (1310 nm), $\alpha_q = 0.27$ dB/km (1550nm), $P = 0.25$ mW.

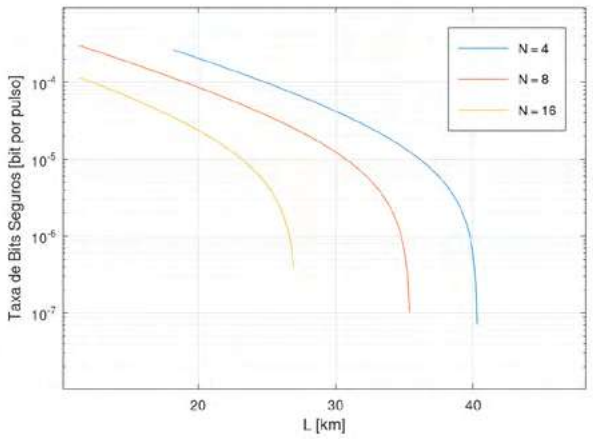


Fig. 3. Taxa de bits seguros versus L (comprimento do canal). $\alpha_c = 0.48$ dB/km (1310 nm), $\alpha_q = 0.27$ dB/km (1550nm), $P = 0.5$ mW.

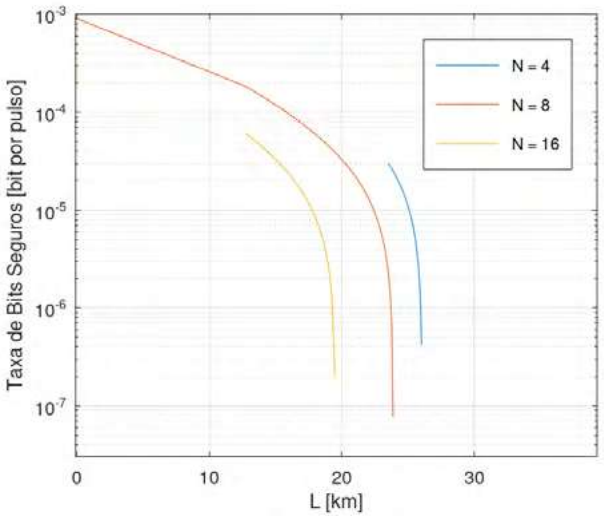


Fig. 4. Taxa de bits seguros versus L (comprimento do canal). $\alpha_c = 0.48$ dB/km (1310 nm), $\alpha_q = 0.27$ dB/km (1550nm), $P = 0.6$ mW.

Como pode ser obeservado nas Figs. 2, 3 e 4, quanto maior o valor de P, maior o valor de p_R e menores serão a taxa R e o comprimento L do canal. Nota-se que, comprimentos menores de fibra (menor atenuação do sinal clássico) devem ser compensados com aumento do número de usuários para que o valor de p_R atinja um valor que permita a realização do protocolo. O mesmo ocorre se ao invés da diminuição do comprimento do canal, tivermos o aumento da potência P. De fato, a taxa R é maior para o valor de N que minimiza $[\alpha_q L(N)+10\log_{10}(N)]$. Para ter alta taxa de chave segura são necessários enlaces curtos (mais fótons do sinal quântico chegarão ao detector), porém, neste caso o SRS aumenta (os sinais clássicos são menos atenuados) o que aumenta o QBER diminuindo a taxa de chave segura. Por exemplo, para uma potência P de 0.7 mW não é possível realizar QKD com $N = 4$ pois o sinal clássico não sofrerá a atenuação necessária para que o p_R atinja o valor que permita a realizacao de QKD.

IV. Conclusões

A realização de QKD em redes ópticas passivas com multiusuários e com coexistência de dados clássicos e quânticos, requer um projeto que faça o correto balanceamento entre o número de usuários e a potência óptica clássica utilizada, de forma a permitir a realização do protocolo de QKD. A eq. (19) mostra a relação entre número de usuários e potência óptica que deve ser obedecida. Perceba-se que são cruciais na eq. (19) os valores de atenuação dos sinais quântico (α_q) e clássico (α_c). Portanto, uma escolha apropriada dos comprimentos de onda utilizados também é fundamental. Por exemplo, utilizar o comprimento de onda de 1480 nm para o sinal clássico resulta em um valor menor de α_c, que deve ser compensado com um aumento do comprimento do canal, o que por sua vez causará um aumento da perda em 1550 nm diminindo a taxa de transmissão.

Agradecimentos

Este trabalho foi parcialmente financiado pelas agências CNPq (309374/2021-9) e CAPES (001).

Referências

[1] H.-K. Lo, M. Curty, K. Tamaki, “Secure quantum key distribution”, *Nature Photonics*, v. 18, pp. 595-604, 2014.

[2] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, M. Peev, “The security of practical quantum key distribution”, *Rev. Mod. Phys.*, v. 81, pp. 1301–1350, 2009.

[3] K. Inoue, “Differential phase-shift quantum key distribution systems”, *IEEE Sel. Top. in Quant. Elec.*, v. 21, no. 3, pp. 6600207, 2015.

[4] Y.-P. Li, W. Chen, F.-X. Wang, Z.-Q. Yin, L. Zhang, H. Liu, S. Wang, D.-Y. He, Z. Zhou, G.-C. Guo, Z.-F. Han, “Experimental realization of a reference-frame independent decoy BB84 quantum key distribution based on Sagnac interferometer”, *Optics Letters*, v. 44, no. 18, pp. 4523-4526, 2019.

[5] C. H. Bennett, G. Brassard, “Quantum Cryptography: Public Key Distribution and Coin Tossing” in Proc. IEEE International Conference on Computers, Systems and Signal Processing. pp. 175–179 (IEEE Press, New York, 1984).

[6] C. H. Bennet, “Quantum cryptography using any two non-orthogonal states”, *Phys. Rev. Lett.*, v. 68, pp. 3121, 1992.

[7] K. Inoue, E. Waks, Y. Yamamoto, “Differential-phase-shift quantum key distribution using coherent light”, *Phys. Rev. A*, v. 68, no. 2, pp. 022317, 2003.

[8] D. Stucki, N. Brunner, N. Gisin, V. Scarani, H. Zbinden, “Fast and simple one-way quantum key distribution”, *Appl. Phys. Lett.*, v. 87, no. 19, pp. 194108, 2005.

[9] B.-H. Li, Y.-M. Xie, Z. Li, C.-X. Weng, C.-L. Li, H.-L. Yin, and Z.-B. Chen, “Long-distance twin-field quantum key distribution with entangled sources”, *Opt. Lett.*, v. 46, no. 22, pp. 5529, 2021.

[10] H.-K. Lo, M. Curty, B. Qi, “Measurement-device-independent quantum key distribution”, *Phys. Rev. Lett.*, v. 108, pp. 130503, 2012.

[11] P. V. P. Pinheiro, R. V. Ramos, “Two-layer quantum key distribution”, *Quantum Inf Process*, v. 14, pp. 2111, 2015.

[12] G. L. de Oliveira, R. V. Ramos, “Quantum-chaotic cryptography”, *Quantum Inf Process*, v. 17, pp. 40, 2018.

[13] S. Aleksic, F. Hipp, D. Winkler, A. Poppe, B. Schrenk, G. Franzl, “Perspectives and limitations of QKD integration in metropolitan area networks”, *Opt. Express*, v. 23, no. 8, pp. 10359-10373, 2015.

[14] A. Bahrami, A. Lord, T. Spiller, “Quantum key distribution integration with optical dense wavelength division multiplexing: a review”, *IET Quantum Commun.*, v. 1, no. 1, pp. 9-15, 2020.

[15] I. Vorontsova, R. Goncharov, A. Tarabrina, F. Kiselev, V. Egorov, “Theoretical analysis of quantum key distribution systems when integrated with a DWDM optical transport network”, arXiv:2209.15507, 2022.

[16] K. A. Patel, J. F. Dynes, I. Choi, A.W. Sharpe, A. R. Dixon, Z. L. Yuan, R.V. Penty, A. J. Shields, “Coexistence of High-Bit-Rate Quantum Key Distribution and Data on Optical Fiber”, *Phys. Rev. X*, v. 2, pp. 041010, 2012.

[17] B. Fröhlich, J. F. Dynes, M. Lucamarini, A. W. Sharpe, S. W.-B. Tam, Z. Yuan, A. J. Shields, “Quantum secured gigabit optical access networks”, *Scientific Reports*, v. 5, pp. 18121, 2015.

[18] C. Cai, Y. Sun, Y. Ji, “Intercore spontaneous Raman scattering impact on quantum key distribution in multicore fiber”, *New J. Phys.*, v. 22, pp. 083020, 2020.

[19] G. B. da Silva, R. V. Ramos, “The Lambert–Tsallis Wq function”, *Physica A*, v. 525, pp. 164, 2019.

[20] E. M. F. Curado, C. Tsallis, “Generalized statistical mechanics: connection with thermodynamics”, *J. Phys. A*, v. 24, pp. L69, 1991. [Corrigenda: v. 24, pp. 3187, 1991 and v. 25, pp. 1019, 1992].

[21] C. Tsallis, “Possible generalization of Boltzmann-Gibbs statistics”, *J. Stat. Phys.*, v. 52, pp. 479, 1988.

[22] 1. R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey and D. E. Knuth, “On the Lambert W function”, *Advances in Comput. Math.*, v. 5, pp. 329 – 359, 1996.

[23] S. R. Valluri, D. J. Jeffrey, R. M. Corless, “Some applications of the Lambert W function to Physics”, Canadian J. of Phys., v. 78, no. 9, pp. 823-831, 2000.

[24] F.V. Mendes, C. Lima, R. V. Ramos, “Applications of the Lambert–Tsallis Wq function in quantum photonic Gaussian boson sampling”. *Quant. Inf Process.*, v. 21, pp. 215, 2022.

[25] J. S. de Andrade, K. Z. Nobrega, R. V. Ramos, “Analytical solution of the current-voltage characteristics of circuits with power-law dependence of the current on the applied voltage using the Wq de Lambert-Tsallis function”, *IEEE Trans. Circuits Syst. II Express Briefs*, 2021.

[26] J. R. da Silva, R. V. Ramos, “Applications of the Lambert–Tsallis Function in X-Ray Free Electron Laser”, *IEEE Trans. on Plasma Sci.*, v. 50, no. 10, pp. 3578-3582, 2022. doi: 10.1109/TPS.2022.3205545.

[27] R. L. C. Damasceno, J. S. Andrade, R.V. Ramos, “Applications of the Lambert–Tsallis Wq function in QKD”, *J. Opt. Soc. Am. B*, v. 40, no. 9, pp. 2280-2286, 2023.

[28] .B.-X. Wang, S.-B. Tang, Y. Mao, W. Xu, M. Cheng, J. Zhang, T.-Y. Chen, J.-W. Pan, “Practical quantum access network over a 10 Gbit/s Ethernet passive optical network”, *Opt. Express*, v. 29, no. 23, pp. 38582/1-9, 2021.

[29] P. D. Townsend, “Quantum cryptography on multiuser optical fibre networks” *Nature*, v. 385, pp. 47–49, 1997.

[30] X. Ma, B. Qi, Y. Zhao, and H.-K. Lo, “Practical decoy state for quantum key distribution”, *Phys. Rev. A*, v. 72, pp. 012326/1-15, 2005.

7º WECIQ

Simulação de Circuitos Quânticos Ópticos

V. F. Guedes, F. A. Mendonça e R. V. Ramos

***Resumo* — Neste trabalho apresentamos os resultados de um simulador de circuitos quânticos ópticos. Os circuitos quânticos considerados são compostos por divisores de feixes, moduladores de fase e contadores de fótons. Além disso, estados coerentes, estados de Fock e estados comprimidos da luz são utilizados. Os resultados apresentados demonstram que o simulador produzido é uma ferramenta útil na análise de circuitos para geração condicional de estados quânticos e de circuitos de amostragem Gaussiana de bósons.**

Palavras-Chave* — *Computação quântica, óptica quântica, simulação numérica, amostragem Gaussiana de bósons.

***Abstract* — In this work we present the results of an optical quantum circuit simulator. The quantum circuits considered are composed by beam splitters, phase modulators and photon counters. Furthermore, coherent, Fock and squeezed states are used. The presented results demonstrate that the simulator is a useful tool for analysis of conditional quantum state generation**

Keywords* — *Quantum computing, quantum optics, numerical simulation, Gaussian boson sampling.

I. Introdução

Computadores quânticos ainda estão em estágio inicial de desenvolvimento, muitos desafios tecnológicos ainda não foram superados e, por isso, computadores quânticos ainda não estão disponíveis em larga escala. Assim, a computação quântica ainda conta com simuladores para o estudo, desenvolvimento e análise de algoritmos quânticos. De fato, há atualmente um número significativo simuladores disponíveis como o QISKIT, Cirq, QUIRK, QUBIT4MATLAB dentre muitos outros [1]. A maioria destes simuladores lida com qubits, ou seja, dois estados quânticos ortogonais representam os bits lógicos '0' e '1'. As portas quânticas como CNOT e portas de um qubit processam esses qubits. Outra forma de computação quântica sem a utilização de bits lógicos é possivel. Nesta, a distribuição de números de fótons ou os valores das quadraturas são utilizados para computação. É uma forma menos convencional, mas ainda útil e poderosa. Um exemplo é a amostragem Gaussiana de bósons [2-4]. Neste caso, circuitos ópticos construídos com divisores de feixes, moduladores de fase e contadores de fótons são utilizados. Um exemplo de simulador de circuitos quanto-ópticos é o *strawberry fields*, da empresa Xanadu, disponível em https://strawberryfields.ai/.

Nesta direção, o presente trabalho apresenta resultados de um simulador de circuitos quânticos ópticos. A geração condicional de estados quânticos da luz e a amostragem Gaussiana de bósons são considerados.

Este trabalhado está dividido da seguinte forma: na Seção II é feita uma revisão dos conceitos de óptica quântica necessários ao entendimento do trabalho; Na Seção III, o simulador de circuitos quânticos ópticos é utilizado na análise de circuitos quânticos com dois, três e quatro estados. Por fim as conclusões são descritas na Seção IV.

Vitor Ferreira Guedes, Departamento de Engenharia de Teleinformática, UFC, Fortaleza-Ce e-mail: vitorfergue@gmail.com; Fábio Alencar Mendonça, Departamento de Telemática, IFCE, Fortaleza-Ce email: fabioalencar@ifce.edu.br; Rubens Viana Ramos, Departamento de Engenharia de Teleinformática, UFC, Fortaleza-Ce, e-mail: rubens.ramos@ufc.br.

II. Conceitos Básicos de Óptica Quântica em Dimensão Finita

Um estado coerente pode ser expandido na base de estados número $\{|0\rangle...|s\rangle\}$, como sendo

$$|\alpha\rangle = \sum_{n=0}^{s} c_n |n\rangle \tag{1}$$

sendo $s + 1$ a dimensão do espaço finito de Hilbert. Usando o operador deslocamento de Glauber tem-se que:

$$|\alpha\rangle = e^{\alpha a^\dagger - \alpha^* a}|0\rangle. \tag{2}$$

O estado número é representado pelo vetor

$$|n\rangle = (0 \quad \dots \quad 1 \quad \dots \quad 0)^T \tag{3}$$

no qual somente um elemento não nulo ocorre na posição $(n+1)$. Os operadores aniquilação e criação, neste espaço finito, são dados, respectivamente, por:

$$\hat{a} = \sum_{n=1}^{s} \sqrt{n}|n-1\rangle\langle n| \text{ e } \hat{a}^\dagger = \sum_{n=1}^{s} \sqrt{n}|n\rangle\langle n-1|, \tag{4}$$

com limites inferior e superior dados por $\hat{a}|0\rangle = 0$ e $\hat{a}^\dagger|s\rangle = 0$. Os estados comprimidos, por sua vez, são obtidos através da operação $|S\rangle = S_{r\phi}|0\rangle$ sendo $S_{r\phi}$ operador dado por

$$S_{r\phi} = e^{\frac{1}{2}\left[\alpha^*(a)^2 - \alpha\left(a^\dagger\right)^2\right]}. \tag{5}$$

com $\alpha = re^{i\phi}$. Por fim, divisores de feixes e modulador de fase são matematicamente descritos, respectivamente, pelos operadores

$$U_{BS} = e^{\theta\left(a^\dagger b - ab^\dagger\right)} \tag{6}$$

$$U_\phi = e^{i\phi a^\dagger a}. \tag{7}$$

III. SIMULAÇÃO DE CIRCUITOS QUÂNTICOS ÓPTICOS

O primeiro circuito a ser considerado é o interferômetro de Mach-Zehnder mostrado na Fig. 1.

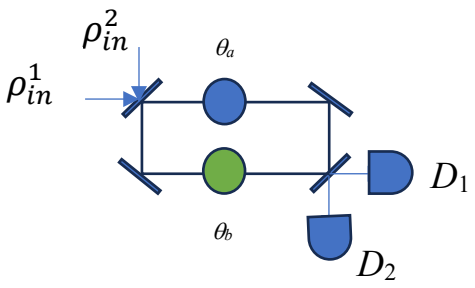


Fig. 1. Interferômetro de Mach-Zehnder. D_1 e D_2 são detectores.

Na primeira simulação, mostrada na Fig. 2, tem-se $\rho_{in}^1 = |\alpha\rangle\langle\alpha|$ (estado coerente) com $\alpha = 3$ e $\rho_{in}^2 = |0\rangle\langle 0|$. Na segunda simulação, mostrada na Fig. 3, tem-se $\rho_{in}^1 = |S\rangle\langle S|$ (estado comprimido) com $r = 3$, $\phi = 0$ rad, $\rho_{in}^2 = |0\rangle\langle 0|$. Três situações são consideradas: θ_a-$\theta_b \in \{0, \pi/2, \pi\}$ rad.

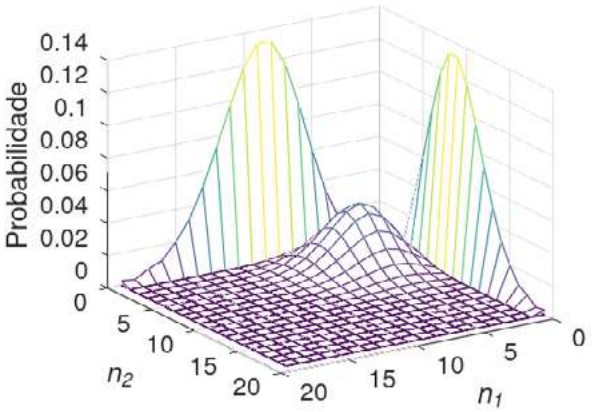


Fig. 2. Distribuições do número de fótons nas saídas do Mach-Zehnder da Fig. 1 quando $\rho_{in}^1 = |\alpha\rangle\langle\alpha|$ com $\alpha = 3$, $\rho_{in}^2 = |0\rangle\langle 0|$ e θ_a-$\theta_b \in \{0, \pi/2, \pi\}$ rad.

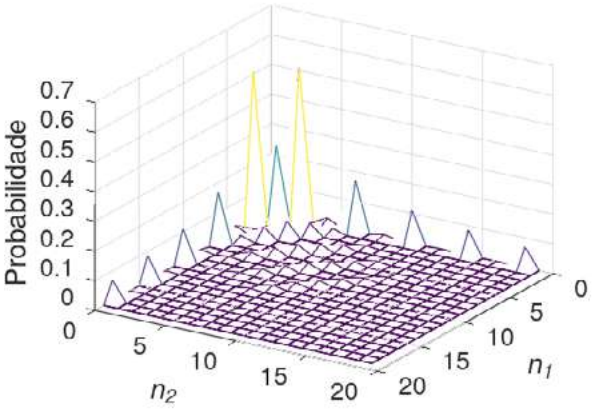


Fig. 3. Distribuições do número de fótons nas saídas do Mach-Zehnder da Fig. 1 quando $\rho_{in}^1 = |S\rangle\langle S|$ com $r = 3$, $\phi = 0$ rad, $\rho_{in}^2 = |0\rangle\langle 0|$ e θ_a-$\theta_b \in \{0, \pi/2, \pi\}$ rad.

As distribuições do número de fótons podem ser vistas nas Fig. 2 e 3, respectivamente. Como esperado, nas situações em que θ_a-$\theta_b \in \{0, \pi\}$ rad tem-se perfeita interferência construtiva em uma saída e destrutiva ($n_1 = 0$ ou $n_2 = 0$) na outra. Quando θ_a-$\theta_b = \pi/2$ rad os fótons se distribuem com a mesma probabilidade pelas duas saídas (curva que passa pela reta diagonal do piso do gráfico).

Um esquema óptico muito útil na geração de estados quânticos é mostrado na Fig. 4 [5-8].

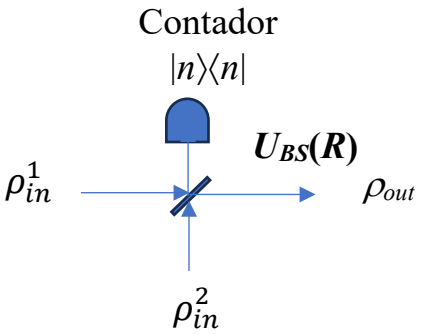


Fig. 4. Esquema básico para a geração de estados quânticos condicionado ao número de fótons medido em uma das saídas.

Basicamente, o estado quântico na saída (ρ_{out}) depende do estado quântico na entrada (ρ_{in}), da reflectância do divisor de feixes e do número de fótons medidos na outra saída. Aqui, consideramos uma versão ampliada do esquema na Fig. 4. Como mostrado na Fig. 5, dois divisores de feixes são utilizados, além de um modulador de fase. Uma das saídas é medida.

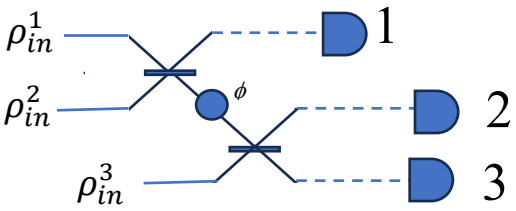


Fig. 5. Esquema ampliado com dois divisores de feixes para geração condicional de um estado quântico bipartite.

A Fig. 6 (7 e 8) a seguir mostra a distribuição do número de fótons nas saídas 2 e 3 (1 e 3, 1 e 2) condicionada à medição de um fóton na saída 1 (2, 3). Os parâmetros utilizados são: $\theta = \pi/4$ (eq. 6 – divisores balanceados), $\phi = 0$ (sem modulação de fase), $\rho_{in}^1 = \rho_{in}^3 = |\alpha\rangle\langle\alpha|$ com $\alpha = 2$ e $\rho_{in}^2 = |1\rangle\langle 1|$.

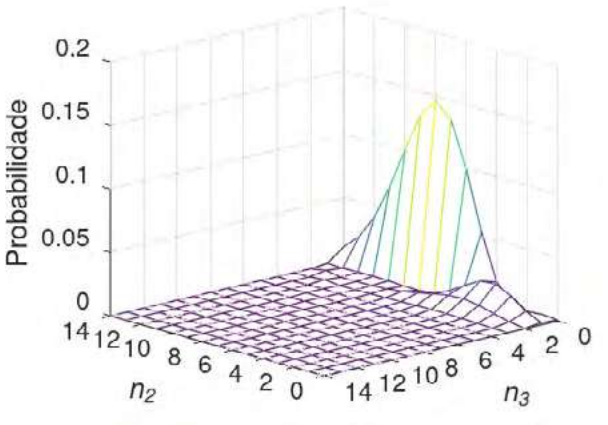


Fig. 6. Distribuição do número de fótons das saídas 2 e 3 do circuito da Fig. 5 condicionada à medição de um fóton na saída 1.

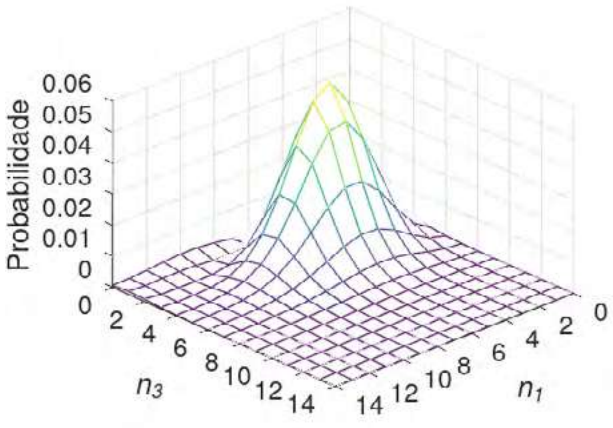


Fig. 7. Distribuição do número de fótons das saídas 1 e 3 do circuito da Fig. 5 condicionada à medição de um fóton na saída 2.

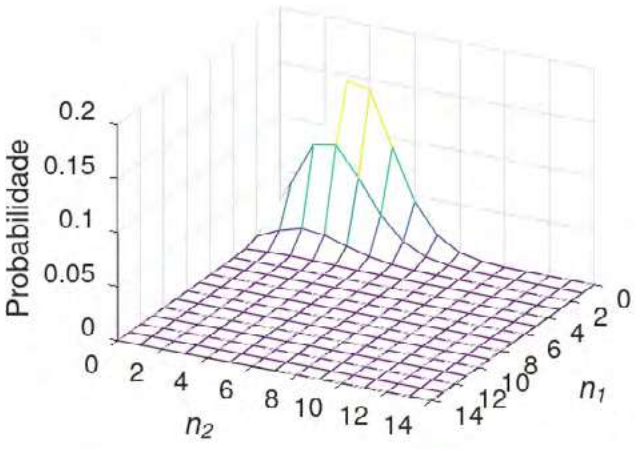


Fig. 8. Distribuição do número de fótons das saídas 1 e 2 do circuito da Fig. 5 condicionada à medição de um fóton na saída 3.

Por fim, a Fig. 9 mostra um circuito com seis divisores de feixes balanceados ($\theta = \pi/4$) e 8 moduladores de fase (os dois primeiros com $\phi = \pi/2$ e os demais com $\phi = \pi/4$). Os estados de entrada são quatro estados comprimidos com $r = 1$ e fase zero.

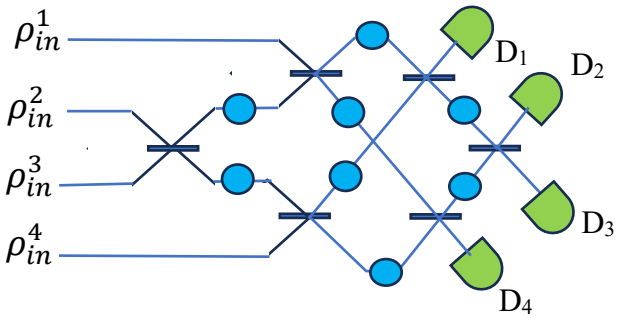


Fig. 9. Circuito óptico para amostragem Gaussiana de bósons com seis divisores de feixes, oito moduladores de fase e quatro contadores de fótons.

Os detectores D_1, D_2, D_3 e D_4 são contadores de fótons. O número de fótons em cada saída pode variar de zero a oito. Portanto, existem 6561 sequências possíveis, sendo a primeira (sequência #1) '0000' (zero fótons medidos nas quatro saídas) e a última (sequência #6561) '8888' (zero fótons medidos nas quatro saídas). A distribuição de probabilidade dessas sequências esta mostrada na Fig. 10.

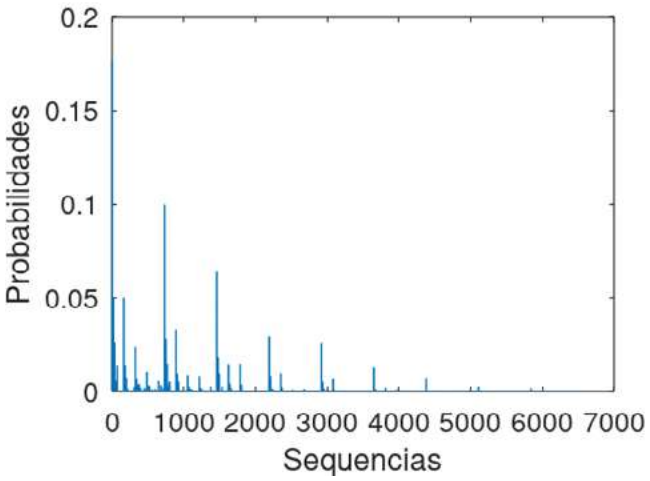


Fig. 10. Distribuição de probabilidade de ocorrência das 6561 sequências obtidas nas saídas do circuito amostrador de bósons da Fig. 9.

Na Fig. 10 pode-se notar o caráter nada suave da distribuição obtida, o que está em acordo com a dificuldade de calculá-la em um computador clássico.

IV. Conclusões

Fugindo do lugar comum de simulações de circuitos quânticos utilizando portas quânticas de qubits, mostramos que um simulador que considere estado quânticos da luz e componentes ópticos simples, é uma ferramenta útil que pode explorar a rica dinâmica da geração de estados quânticos condicionados ao resultado de uma medição e investigar a amostragem Gaussiana de bósons. Além disso, embora não tenhamos discutido nesse trabalho, certamente o simulador

apresentado é útil no desenolvimento de computação quântica com variáveis contínuas se ao invés da contagem de fótons, detecções homódinas forem utilizadas.

AGRADECIMENTOS

Este trabalho foi parcialmente financiado pelas agências CNPq (309374/2021-9) e CAPES (001).

REFERÊNCIAS

[1] https://thequantuminsider.com/2022/06/14/top-63-quantum-computer-simulators-for-2022/ and references there in.
[2] T. R. Bromley, J. M. Arrazola, S. Jahangiri, J. Izaac, N. Quesada, A. D. Gran, M. Schuld, J. Swinarton, Z. Zabaneh, N. Killoran, "Applications of near-term photonic quantum computers: software and algorithms", *Quantum Sci Tech*., v. 5, pp. 034010, 2020.
[3] D. J. Brod, E. F. Galvão, A. Crespi, R. Osellame, N. Spagnolo, F. Sciarrino, "Photonic implementation of boson sampling: a review", *Adv Photon*, v. 1, no. 3, pp. 034001, 2019.
[4] F. V. Mendes, C. Lima, R. V. Ramos, "Applications of the Lambert–Tsallis Wq function in quantum photonic Gaussian boson sampling", *Quant. Inf. Process*., v. 21, pp. 215, 2022.
[5] M. Dakna, L. Knoll, D.-G. Welsch, "Quantum state engineering using conditional measurement on a beam splitter", *The European Phys. J. D - Atomic, Molecular, Optical and Plasma Physics*, v. 3, pp. 295, 1998.
[6] C. Yang, F.-L. Li, "Nonclassicality of photon-subtracted and photon-added- then-subtracted Gaussian states", *J. Opt. Soc. Am. B*, v. 26, no. 4, pp. 830, 2009.
[7] V. Parigi, A. Zavatta, M. Bellini, "Manipulating thermal light states by the controlled additon and subtraction of single photons", *Laser Phys. Lett.*, v. 5, no. 3, pp. 246, 2008.
[8] P. V. P. Pinheiro, R. V. Ramos, "Quantum communication with photon-added coherent states", *Quant. Inf. Process*., v. 12, pp. 537–547, 2013.

7º WECIQ

Efficient Computation of the Wave Function $\psi_n(x)$ using Hermite Coefficient Matrix in Python

Matheus Cordeiro, Italo Bezerra and Hilma Vasconcelos

***Abstract*— With the acceleration of quantum hardware development each year, the demand for fast and accurate Quantum Computing Simulation tools has grown significantly. This growth is largely due to the challenges in accessing real quantum hardware. In this context, our work aims to gain a computational advantage in calculating the wave function of a Quantum Harmonic Oscillator. We achieve this through a hybrid strategy that relies partially on the efficient and precise calculation of the Hermite polynomial using a coefficient matrix, combined with the use of a Python Just-In-Time (JIT) compilation optimizer.**

Keywords*— *Wave Function, Hermite Polynomials, Quantum Harmonic Oscillator, JIT, Python.

I. Motivation and Theoretical Foundations

The simplicity and convenience offered by the Python programming language [1] have established it as a standard tool for Classical Quantum Computing Simulation. Python excels in building highly abstract ideas due to its high-level nature. Open access to source code promotes reproducibility of computational projects and a greater exchange of ideas.

The development of specific libraries for quantum computing has facilitated the use of Python language in research in this field. For example, *Piquasso* [2] is an open-source Python library developed by the Budapest Quantum Computing Group, which focuses on the simulation of photonic quantum computers. This library differs from the others because it offers simulations that use the *TensorFlow* and the *JAX* libraries on the backend. These are two Python libraries specifically designed for Deep Learning applications. The use of these two Python libraries by *Piquasso* is related to matrix differentiation in problems involving Continuous-Variable Quantum Neural Networks (CVQNN). In addition to the library for direct use in Python, *Piquasso* also provides a drag-and-drop interface, that is code-free and easy to handle, for simulating photonic circuits: Piquasso Dashboard.

Spinsim [3] is a simulation package for spin-1/2 and spin-1 quantum systems under time-dependent control. Developed in Python, this package has been optimized for GPU to handle geometric integration calculations, which demand substantial computational resources due to the magnitude of the calculations. This optimization is achieved using the *Numba* library [4] in Python. *Numba* has a Just-In-Time (JIT) optimizer based on the LLVM compilation infrastructure, which was developed in C++ and was specifically designed to enhance both compilation and execution times. With *Numba*, we can get runtime translation of Python code into optimized machine code. The geometric integration provided by this package outperforms `sesolve` from *QuTip*, `NDSolve` from *Mathematica* and `ivp_solve` from *Scipy* [5].

A. The Wave Function of a Quantum Harmonic Oscillator

Solving the Schrödinger Equation is a fundamental task in the field of Quantum Mechanics, as its solution, the wave function, mathematically describes the quantum state of one or more particles. The wave function provides a non-deterministic description of the physical system, which implies the need to deal with probabilities [8].

From the same perspective, the quantum harmonic oscillator is one of the most relevant models in Quantum Mechanics. Its wave function describes all harmonic potentials, which is essential for representing vibrational states of varying energies [8]. This model is applied to the description of systems in Quantum Optics, where each vibrational state is associated with a quantum number n, corresponding to the number of $n = 0$ photons in that state. For instance, for, we have the vacuum state, representing the absence of photons [9].

For the one-dimensional harmonic oscillator, the wave function can be derived from the time-independent Schrödinger Equation, where $V(x) = (m\omega^2 x^2/2)$ represents the potential energy of the quantum harmonic oscillator [10]:

$$E\psi(x) = -\left(\frac{\hbar^2}{2m}\right)\frac{d^2}{dx^2}\psi(x) + \left(\frac{m\omega^2 x^2}{2}\right)\psi(x). \quad (1)$$

where E represents the total energy of the system, $\hbar$ is the reduced Planck constant, m is the particle's mass, x is the position, ω is the angular frequency of the harmonic potential, and ψ is the one-dimensional, time-independent wave function to be found. Solving this equation yields the following result [11]:

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}}\left(\frac{m\omega}{\pi\hbar}\right)^{\frac{1}{4}} e^{-\frac{m\omega x^2}{2\hbar}} H_n\left(\sqrt{\frac{m\omega}{\hbar}}x\right),$$
$$n = 0, 1, 2, 3, \ldots \quad (2)$$

In Eq. (2), H_n is the Hermite polynomial of degree n. Thus, $\psi_n(x)$ is considered a Hermite function. This wave function, like any other wave function, has the following properties: it is a solution to the Schrödinger Equation, it is normalizable

Matheus Cordeiro, Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará (UFC), Fortaleza-CE, e-mail: matheuscord@alu.ufc.br; Italo Bezerra, Universidade Estadual do Ceará (UECE), Iguatu-CE, e-mail: italop@hotmail.com; Hilma Vasconcelos, Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará (UFC), Fortaleza-CE, e-mail: hilma@ufc.br. Este trabalho foi parcialmente financiado pela CAPES (88887.834371/2023-00).

($\int_{-\infty}^{\infty} |\psi_n(x)|^2 = 1$), and it is continuous in x, as well as its derivative [8].

Fig. 1 gives a better understanding of the behavior of the wave function for different values of n. In this graph, we can see the energy levels of the system changing in discrete energy increments as n increases [10]. In Quantum Optics, this change in energy levels is linked to the absorption or emission of photons in a system, altering the value of n and consequently the degree of the Hermite polynomial in Eq. (2) [9].

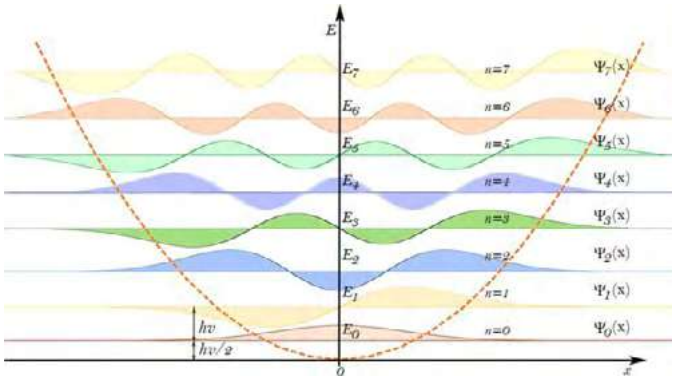


Fig. 1. Wave functions and energies for different n [12].

For large values of n and x, achieving efficient and accurate computation of the Hermite polynomial becomes essential for a reliable modeling of a Quantum Harmonic Oscillator. This work aims to present a solution to enhance the efficiency of Hermite polynomial calculations. In the next section, we will discuss the methods used for computing the Hermite polynomial, as well as well-known Python libraries for performing this task.

B. Calculating the Hermite Polynomial

The Hermite polynomial can be explicitly defined by Rodrigues' formula as follows [13]:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}. \quad (3)$$

These polynomials obey the following recurrence relations [13]:

$$\begin{cases} H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \\ H_0(x) = 1, \\ H_1(x) = 2x. \end{cases} \quad (4)$$

We can also represent the Hermite polynomial as a sum in the following manner [13]:

$$H_n(x) = n! \sum_{l=0}^{\left\lfloor \frac{n}{2} \right\rfloor} \frac{(-1)^l (2x)^{n-2l}}{l!(n-2l)!}. \quad (5)$$

We can also work with calculations for the multidimensional Hermite polynomial $G_A^k(b)$ [14], expanding the exponential function of a quadratic polynomial into a Taylor series:

$$K_A(x,b) = \exp\left(x^T b + \frac{1}{2} x^T A x\right) = \sum_{k \geq 0} \frac{G_A^k(b)}{k!} x^k. \quad (6)$$

Where $K_A(x, b)$ is called the generating function for multi-dimensional Hermite polynomials, A is a $h \times h$ symmetric matrix, b is a vector of dimension h containing values, x is a vector of dimension h containing variables, and k is a vector of dimension h containing indices.

The approaches mentioned before (Rodrigues' formula - Eq. (3), recurrence - Eq. (4), and finite power series - Eq. (5)), have their drawbacks when calculating Hermite polynomials for large values of n. Some may be too slow, while others might be too inaccurate. In this context, programmers often employ different strategies to overcome these challenges, even within the programming language itself. For example, Python's *Scipy* library offers two libraries for computing the Hermite polynomial: `scipy.special.hermite` and `scipy.special.eval_hermite`. You can find the source code for each of them at: **hermite** and **eval_hermite**. The first function uses Eq. (3), working with a code that finds the roots of the Hermite polynomial. The second implements an iterative form of the recurrence defining the probabilistic Hermite polynomial, similar to Eq. (4), and relies on the following relationship to obtain the Hermite polynomial [13]:

$$H_n(x) = 2^{\frac{n}{2}} He_n\left(\sqrt{2}x\right). \quad (7)$$

where He_n is the probabilistic Hermite polynomial. This function is implemented in *Cython* [15], which is a hybrid language between C and Python, and is more efficient than a function implemented in standard Python.

On the other hand, *Numpy* [16] offers a function called `numpy.polynomial.hermite.hermval`, whose source code can be found at **hermval**. This function returns a Hermite series defined as follows:

$$p(x) = c_0 H_0(x) + c_1 H_1(x) + \cdots + c_n H_n(x). \quad (8)$$

With this function, it's possible to obtain only the value of $H_n(x)$ by setting the following coefficient vector as the input to the function: $c_n = [0,0,\ldots,1]$. The calculation of this sum is performed using an iterative algorithm for the recurrence in Eq. (4), where the recursive nature of using previous values to obtain current values is exploited.

The *Mr Mustard* library [17], developed by the photonic quantum computing company Xanadu, uses a strategy that is currently regarded as one of the most efficient available. This strategy is utilized in the `oscillator_eigenstate` function. This library adapts Eq. (6) so efficiently in its implementation that the execution time of this function remains virtually unchanged even with an increase in the number of photons n.

In this work we implement two strategies to compute the wave function: one using `eval_hermite` from *Scipy* and a hybrid strategy that uses Hermite coefficient matrix in conjunction with `eval_hermite`. The resulting wave functions computed using these two strategies are compared to the wave functions calculated using `oscillator_eigenstate` from *Mr Mustard*. The runtime efficiency of our two implementations is then compared to demonstrate the computational advantage of using the Hermite coefficient matrix.

II. Methodology: Wave Function with Hermite Coefficient Matrix

The increase in efficiency when using our hybrid strategy arises from storing the coefficients of the Hermite polynomials in a pre-fixed matrix for use during the calculation. Thus, the Hermite polynomial is calculated with just a simple linear algebra operation. The Hermite polynomials varies with n according to Table I [13].

TABLE 1. Hermite polynomials up to $n = 6$.

n	$H_n(x)$
0	1
1	$2x$
2	$4x^2 - 2$
3	$4x^3 - 12x$
4	$16x^4 - 48x^2 + 12$
5	$32x^5 - 160x^3 + 120x$
6	$64x^6 - 480x^4 + 720x^2 - 120$

The coefficient matrix for n = 6 is defined as follows:

$$C_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 4 & 0 & -12 & 0 \\ 0 & 0 & 16 & 0 & -48 & 0 & 12 \\ 0 & 32 & 0 & -160 & 0 & 120 & 0 \\ 64 & 0 & -480 & 0 & 720 & 0 & -120 \end{bmatrix}$$

The i-th row of this coefficient matrix has the coefficients related to Hermite polynomials of degree i, which are obtained by taking the inner product of this row with a vector x_i^p : $\mathrm{H_i(x)} = \mathrm{x_i^p} \cdot \mathrm{C_n[i]}$. Fig. 2 illustrates the idea just described. In this illustration, to calculate the Hermite polynomial of degree *i*, where $\mathrm{i} < \mathrm{n} + 1$, we must construct a coefficient matrix with a degree always greater than the intended working degree.

Initially, the coefficient matrix was constructed using *Sympy* library from Python. This approach failed to outperform the efficiency of the *Scipy* strategy implemented in the `eval_hermite` function in terms of speed. However, by using *Numba*, the computation of the wave function that uses the coefficient matrix to calculate the Hermite polynomial was faster compared to using *Scipy*'s `eval_hermite`.

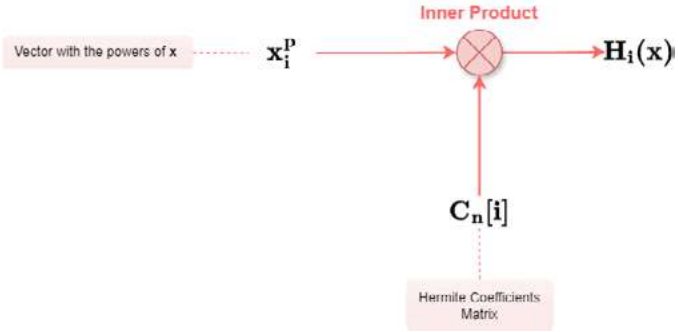


Fig. 2. Computing the Hermite polynomial using the coefficient matrix.

However, it was necessary to replace the factorial in the wave function by the gamma function, where $\mathrm{n!} = \Gamma(\mathrm{n}+1)$. Using the factorial in Numba introduces inaccuracy from n equal to 21 onwards. Additionally, the use of *Numba*, through the `@jit` decorator, on the wave function calculation, introduced inaccuracy for values of n greater than 60 when compared with the strategy using *Scipy* to calculate of the Hermite polynomial.

So, the best strategy we found was a hybrid calculation for the wave function that combines speed and accuracy:

- For $\mathrm{n} \leq 60$, we make use of the coefficient matrix for computing the Hermite polynomial along with the *Numba* library optimization - $\psi_{C_{matrix}}(x)$.
- For $\mathrm{n} > 60$, we make use of the `eval_hermite` function from *Scipy* for computing the Hermite polynomial - $\psi_{\mathrm{scipy}}(\mathrm{x})$.

The complete implementation of this algorithm can be found in the following GitHub repository: CoEfficients-Matrix-Wavefunction .

III. Results and Discussions

In order to compare the efficiency of this hybrid strategy, several speed tests were conducted, comparing this strategy with one that solely utilizes *Scipy* and with the *Mr Mustard* strategy, used as a reference.

Initially, the execution times of the three strategies were calculated, in milliseconds, over the course of 1,000 tests. This is the time it takes to fill out a matrix Ψ_{nx}, with $\mathrm{n} = 50$ and $\mathrm{x} = 20$. This experiment evaluates only the portion of the hybrid algorithm where $\psi_{\mathrm{hybrid}} = \psi_{\mathrm{C_{matrix}}}$. The result of this initial experiment is shown in Fig. 3, where $\mathrm{t_{avg}}$ is the average execution time. From Fig. 3 we can see that the hybrid algorithm (orange) takes less time when compared to the algorithm that uses only *Scipy* (blue). It is clear that our hybrid strategy has better performance than using solely *Scipy*.

We highlight the difference in time levels between the algorithm belonging to the *Mr Mustard* library (green) and both the hybrid algorithm and the algorithm that solely employs *Scipy*. On average, *Mr Mustard* manages to be around 40 times faster than the hybrid algorithm and about 60 times faster than the algorithm using only *Scipy*.

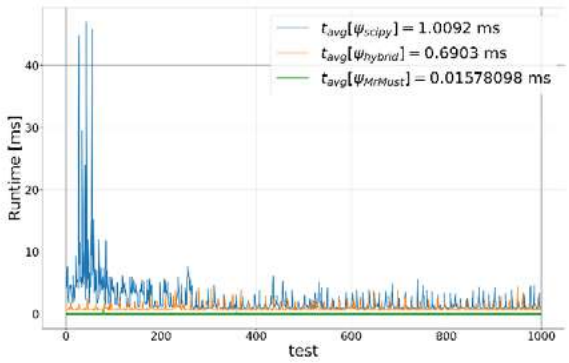


Fig. 3. Speed test filling out the matrix Ψ_{nx} , where $n = 50$ and $x = 20$.

The second experiment was identical to the previous one, but for n = 100. And now, we also evaluate the hybrid algorithm in its entirety. The outcome of this second experiment is shown in Fig. 4. As we can see, the hybrid algorithm consistently maintains a lower time level compared to the algorithm that relies only on *Scipy*. This observation is confirmed by the average execution times of each strategy over 1,000 tests, as indicated in the figure.

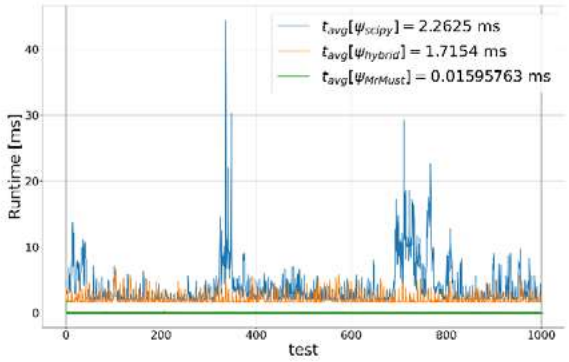


Fig. 4. Speed test of filling out the matrix Ψ_{nx}, where $n = 100$ and $x = 20$, evaluating the hybrid algorithm in its entirety.

In Fig. 5, we have a speed experiment for a fixed value of x, set at 50.0, aiming to evaluate how the execution time of each function evolves when n increases. In this experiment, we compare the smallest execution times among all the strategies, in milliseconds, for 100 tests conducted at each n. We opt for the smallest execution time because the operating system often interferes with the execution time with some random requests that affect the average execution time, despite using filters based on the median to remove outliers.

This behavior is clearly evident in the high peaks present in the previous graphs. Therefore, in this experiment, we seek to analyze, at least, how much time each strategy takes for each value of n, thereby also smoothing out the execution time curve. As we can observe in Fig. 5, the hybrid algorithm (orange) has higher efficiency compared to the algorithm using only *Scipy* (blue), showing an increase in the area of runtime efficiency with increasing values of n. This area is a graphical representation of the computational advantage that the hybrid algorithm holds over the one relying on *Scipy*.

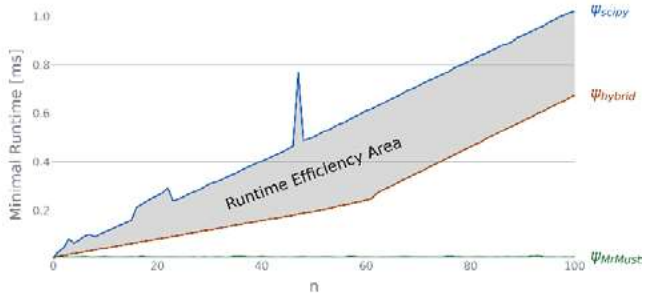


Fig. 5. Speed test of filling out the matrix Ψ_{nx} for a fixed value of x = 50.0 and varying n.

In Fig. 6, 1,000 tests are performed for populating the matrix $\Psi_{n|20}$ for each value of n. In other words, the number of columns in the matrix is fixed and equal to 20, while the number of rows is incremented from 0 to n, consistently conducting 1,000 tests for each n. Furthermore, the version of the hybrid strategy that accepts a vector instead of a numerical value was utilized. The runtime efficiency region is also visible in Fig. 6, demonstrating that the hybrid algorithm once again outperforms the algorithm solely utilizing *Scipy*.

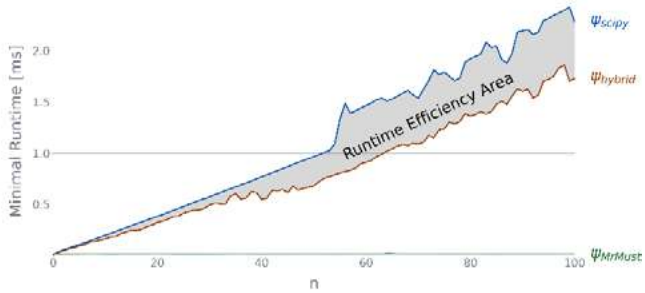


Fig. 6. Speed test of filling out the matrix Ψ_{nx}.

IV. CONCLUSION

The development of a precise and efficient method for calculating a wave function is highly valuable for modeling quantum systems and for executing quantum algorithms on classical computers. This was the purpose of this work, where a hybrid technique was implemented to provide both accuracy and speed.

This approach could be quite useful for well-established tools, should they choose to make use of it. For instance, it could enhance platforms like the Virtual Lab of *Quantum Flytrap* [18], an IDE for Quantum Computing without the use of code, or even *Strawberry Fields* [19], a Python library for designing, optimizing, and utilizing photonic quantum computers.

In a future work, this module currently available on GitHub will be converted into a Python package, bringing greater ease of use for its resources through package managers like *Pip* or *Conda*.

ACKNOWLEDGMENTS

We thank S. G. for his invaluable contributions to this project. M. C. thanks Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Programa de Pós-graduação em Engenharia de Teleinformática (PPGETI) for financial support.

REFERENCES

[1] Python Software Foundation, *Python 3.9.1 Documentation*, available at: https://docs.python.org/3.9/

[2] Budapest Quantum Computing Group, *Piquasso*, [Software]. Available at: https://github.com/Budapest-Quantum-Computing-Group/piquasso

[3] A. Tritt, J. Morris, J. Hochstetter, R.P. Anderson, J. Saunderson, and L.D. Turner, *Spinsim: A GPU optimized python package for simulating spin-half and spin-one quantum systems*, Computer Physics Communications, vol. 287, p. 108701, 2023. DOI:10.1016/j.cpc.2023.108701

[4] Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A llvm-based python jit compiler. *In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC* (pp. 1–6).

[5] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods, vol. 17, pp. 261–272, 2020. DOI:10.1038/s41592-019-0686-2

[6] T. Nguyen and A. J. McCaskey, Extending Python for Quantum-classical Computing via Quantum Just-in-time Compilation, ACM Transactions on Quantum Computing, vol. 3, no. 4, Art. no. 24, Jul. 2022. DOI:10.1145/3544496

[7] X. Fu et al., Quingo: A Programming Framework for Heterogeneous Quantum-Classical Computing with NISQ Features, ACM Trans. Quant. Comput., vol. 2, no. 4, p. 19, 2021. DOI:10.1145/3483528

[8] Beiser, A. (2003). *Concepts of Modern Physics*. 6th ed. McGraw Hill.

[9] Leonhardt, U. (2010). *Essential Quantum Optics: From Quantum Measurements to Black Holes*. 1st ed. Cambridge University Press. ISBN: 978-0-521-86978-2 , 978-0-521-14505-3 .

[10] Basdevant, J.-L. (2023). Lectures on Quantum Mechanics. With Problems, Exercises and Solutions. 3rd ed. Springer, Graduate Texts in Physics. ISBN: 9783031176340, 9783031176357.

[11] Bowers, P. L. (2020). *Lectures on Quantum Mechanics: A Primer for Mathematicians*. Cambridge University Press. ISBN: 1108429769, 9781108429764.

[12] Aerts, D., & Beltran, L. (2019). Quantum Structure in Cognition: Human Language as a Boson Gas of Entangled Words. *Foundations of Science*, 25, 755–802. Available at: https://api.semanticscholar.org/CorpusID:203838565

[13] Olver, F., Lozier, D., Boisvert, R., & Clark, C. (2010). *The NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, NY.

[14] Y. Yao, *Automated design of photonic quantum circuits*, PhD dissertation, Institut Polytechnique de Paris, Feb. 2023. Available at: https://theses.hal.science/tel-04071095

[15] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, e K. Smith, "Cython: The best of both worlds*," Computing in Science & Engineering*, vol. 13, no. 2, pp. 31–39, 2011.

[16] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, et al., *Array Programming with NumPy*, Nature, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI:10.1038/s41586-020-2649-2

[17] Xanadu Quantum Technologies, *Mr Mustard*, versão 0.7.3, 2023. [Software]. Available at: https://github.com/XanaduAI/MrMustard

[18] Jankiewicz, K., Migdal, P., & Grabarz, P. (2022). Virtual Lab by Quantum Flytrap: Interactive simulation of quantum mechanics. *In Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (CHI EA '22), New Orleans, LA, USA. Association for Computing Machinery, New York, NY, USA, Article 175, 1–4. DOI: 10.1145/3491101.3519885. Available at: https://lab.quantumflytrap.com/.

[19] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, *Strawberry Fields: A Software Platform for Photonic Quantum Computing, Quantum*, vol. 3, p. 129, 2019. DOI:10.22331/q-2019-03-11-129

7º WECIQ

Quantum Support Vector Regression for Predicting Zeros of the Riemann Zeta Function

Tharso D. Fernandes, Demerson N. Gonçalves and João T. Dias

***Abstract*— The Riemann hypothesis is one of the seven Millennium Prize Problems to be solved. It conjectures that the zeros of the Riemann Zeta function consist solely of negative even integers and complex numbers with real part equal to 1/2. Several numerical studies have been conducted to find new zeros of the zeta function with real part equal to 1/2. Evaluating the zeta function for large values involves time-consuming calculations. In this context, it is highly valuable to have accurate predictions about the zeros of the zeta function to avoid a large number of function evaluations needed to locate them. In this work, we apply the Quantum Support Vector Regression as a tool to assist in empirical studies of zero locations. We also compare it with classical versions of the regressor.**

***Keywords*— Zeta Riemann Function, Quantum Machine Learning, Quantum Support Vector Regressor.**

I. Introduction

The Riemann hypothesis has intrigued mathematicians for centuries, standing as one of the most profound unsolved problems in the field [1]. Its significance lies in its intricate connection to the distribution of prime numbers. The Riemann Zeta Function (ζ-function) serves as the cornerstone of this hypothesis. Defined for complex numbers, it plays a pivotal role in understanding the distribution of prime numbers along the real number line. The non-trivial zeros of the ζ-function hold the key to unraveling the mysteries of prime numbers [2].

Researchers have explored various approaches to understand the properties of the ζ-function. Ref. [3] investigates the connection between scattering amplitudes and ζ-function, emphasizing locality and meromorphicity. Another study introduces a novel generalization of the ζ-function that converges locally and approximates both trivial and non-trivial zeros [4], while a different work presents efficient methods for computing the ζ-function on the critical line, each with varying complexities [5]. On the other hand, regression techniques, coupled with non-parametric machine learning models, offer a fresh perspective. Among these models, Support Vector Regression (SVR) stand out. These algorithms analyze extensive sequences of system parameters or relevant variables, providing insights into the behavior of the ζ-function. Surprisingly, despite the growing interest in SVRs, only a few known applications exist for modeling the ζ-function [6], [7].

Several recent studies have made significant contributions in exploring the connection between the Riemann ζ-function and quantum computing. Ref. [8] utilizes a Floquet method to identify the first nontrivial zero of the Riemann ζ-function and the first two zeros of Pólya's function through periodically driving a single qubit. This experiment successfully characterizes the zeros of these functions by observing crossings of quasi-energies, providing experimental insight into the connection between those two areas. Additionally, quantum computation has been proposed for prime number functions, leveraging Grover's algorithm and the quantum Fourier transform to estimate the prime counting function and verify the Riemann hypothesis efficiently [9]. Another study applied the quantum Fourier transform to analyze functions related to the Riemann hypothesis using quantum computations, highlighting the potential of quantum computing in this domain [10]. Furthermore, a proposed connection between quantum computing and Zeta functions of finite field equations suggests that quantum circuits could approximate the number of solutions of these equations with unparalleled accuracy [11]. These advancements underscore the promising avenues for research at the intersection of quantum computing and the Riemann hypothesis, emphasizing the need for further exploration in this intriguing field.

In this article, we delve into the Riemann hypothesis and explore its implications within the field of quantum computing, shedding light on an area where the intersection of quantum computing and ζ-functions remains largely unexplored. Quantum Support Vector Regression (QSVR) opens up an exciting possibility for this examination. Here, we introduce a novel approach by employing QSVR for estimating ζ-functions, contributing to the ongoing quest for deeper insights into the connection between quantum mechanics and number theory.

The article is organized as follows: in Section II, we present the Riemann zeta function. Section III introduces the classical version of the SVR, while Section IV discusses the quantum version of the regressor. In Section V, we detail the data selection process for applying the regressors in predicting the zeros of the zeta function. Section VI presents the results obtained by the algorithms in predicting the zeros of the zeta function. Finally, Section VII provides the concluding remarks on the work conducted.

II. Riemann Zeta Function

In the 1730s, Leonhard Euler delved into the study of the series

$$\sum_{n=1}^{\infty} \frac{1}{n^s}, \tag{1}$$

Tharso D. Fernandes is professor at Department of Pure and Applied Mathematics, UFES, Alegre, ES, E-mail: tharso.fernandes@ufes.br. Demerson N. Gonçalves is professor at Collegiate of Mathematics, CEFET/RJ, Petrópolis, RJ, E-mail: demerson.goncalves@cefet-rj.br. João T. Dias is professor at the Department of Telecommunications, CEFET/RJ, Maracanã, RJ, E-mail: joao.dias@cefet-rj.br.

discovering that $1+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\cdots=\frac{\pi^2}{6}$, and subsequently deriving expressions for other positive even integers. In 1748, Euler made another significant discovery, famously known as Euler's product:

$$\sum_{n=1}^{\infty}\frac{1}{n^s}=\prod_{p}\frac{1}{1-p^{-s}}, \tag{2}$$

where the product is taken over all prime numbers p. In 1859, Bernhard Riemann defined the function known as the zeta function (ζ-function) as

$$\zeta(s)=\sum_{n=1}^{\infty}\frac{1}{n^s}, \tag{3}$$

where s is a complex variable. This series converges absolutely for complex numbers s with real part of s ($\mathcal{R}(s)$) greater than 1. Riemann extended the domain of the zeta function, via the process of analytic continuation, to all complex s, except at the pole $s=1$. The newly extended ζ-function [2].

$$\zeta(s)=2^s\pi^{s-1}\sin\left(\frac{\pi s}{2}\right)\Gamma(1-s)\zeta(1-s), \tag{4}$$

where Γ is the Gamma function, a factorial extension defined as $\Gamma(s)=\int_0^{\infty}x^{s-1}e^{-x}dx$. From equation (4), we observe that the ζ-function vanishes at negative even integers, which Riemann referred to as trivial zeros. Later, Riemann hypothesized that all other (non-trivial) zeros lie on a critical line ($\mathcal{R}(s)=\frac{1}{2}$). This conjecture is widely known as the Riemann Hypothesis and is now among Hilbert's unsolved problems.

To date, all computations support the Riemann hypothesis. Furthermore, it has been rigorously demonstrated that no non-trivial zeros of the ζ-function lie outside of a critical strip, defined by $s\in\mathbb{C}:0<\mathcal{R}(s)<1$ [12]. On the critical line, we define the Riemann-Siegel Z-function:

$$Z(t):=\zeta\left(\frac{1}{2}+it\right)\pi^{-it/2}e^{i\arg\left(\Gamma\left(\frac{1}{4}+\frac{it}{2}\right)\right)}. \tag{5}$$

By using the fact that for any complex number z, $\arg(z)=\frac{1}{i}\ln\sqrt{\frac{z}{\bar{z}}}$, the equation (5) can be simplified to:

$$Z(t)=\zeta\left(\frac{1}{2}+it\right)e^{i\theta(t)}, \tag{6}$$

where $\theta(t)$ is the Riemann-Siegel θ-function:

$$\theta(t)=\arg\left(\Gamma\left(\frac{1}{4}+\frac{it}{2}\right)\right)-\frac{\ln\pi}{2}t. \tag{7}$$

Then, the representation for ζ-function on the critical line is:

$$\zeta\left(\frac{1}{2}+it\right)=Z(t)e^{-i\theta(t)}. \tag{8}$$

One advantage of working with the Z-function over the ζ-function is that its values are easier to compute. Carl Siegel developed the asymptotic expansion of the Z-function in 1932, which is defined as

$$Z(t)=2\sum_{n=1}^{N}\frac{\cos(\theta(t)-t\ln(n))}{\sqrt{n}}+R, \tag{9}$$

where $N=\lfloor\sqrt{t/2\pi}\rfloor$ and R is a remainder term. If we separate the ζ-function, on critical line into, its real and imaginary parts, we obtain:

$$\zeta(1/2+it)=Z(t)\cos(\theta(t))-iZ(t)\sin(\theta(t)), \tag{10}$$

$$A(t):=Z(t)\cos(\theta(t)), \tag{11}$$

$$B(t):=-Z(t)\sin(\theta(t)). \tag{12}$$

Locating zeros of the ζ-function on the critical line is equivalent to identifying values of t where both $A(t)$ and $B(t)$ are equal to zero. With this insight, the mathematician Jørgen Gram defined the points bearing his name, g_n, such that $\theta(g_n)=(n-1)\pi$. The existence and uniqueness of such solutions are guaranteed by the monotonicity of the θ-function. Regarding the Gram points, it's important to note that:

$$\zeta(1/2+ig_n)=(-1)^{n-1}Z(g_n). \tag{13}$$

In accordance with the aforementioned equation, if g_n and g_{n+1} are two Gram points where $A(g_n)$ and $A(g_{n+1})$ share the same sign, then $Z(g_n)Z(g_{n+1})<0$. Consequently, $Z(t)$ vanishes at least once within the interval (g_n,g_{n+1}). Thus, Gram points serve as indicators of intervals containing roots of the Riemann-Siegel Z-function and, by extension, nontrivial zeros of the Riemann function within the critical strip. By employing this technique, Jørgen Gram demonstrated that the first 15 imaginary parts (γ_n) of the nontrivial zeros are located between Gram points, specifically $g_{n-1}<\gamma_n<g_n$. While this result cannot be generalized [13], it inspired the demonstration that Gram points and the imaginary part of nontrivial zeros are asymptotically equivalent, namely

$$g_n\sim\gamma_n\sim\frac{2n\pi}{\ln(n)}, \tag{14}$$

for large n [14].

In this study, we will harness this equivalence by computing the Gram points and training both classical (SVR) and quantum (QSVR) algorithms using information about the Gram points. Our objective is to forecast the differences between the Gram points and the imaginary part of the nontrivial zeros of the ζ-function (γ_n-g_{n-1}).

III. Support Vector Regression

SVR is a potent tool in machine learning, offering a flexible approach to modeling complex relationships in data. In our investigation to predict the zeros of the Riemann ζ-function, SVR emerges as a promising technique. Specifically, we will use SVR to predict time-series obtained by taking differences γ_n-g_{n-1}, where γ_n is the imaginary part of the n-th zero located on the critical line and g_{n-1} is the $(n-1)$-th Gram point.

SVR belongs to the class of supervised learning algorithms and excels at handling nonlinear relationships between input and output variables. By employing kernel functions, SVR can efficiently transform data into a higher dimensional feature space where linear separation becomes feasible. This transformation enables the technique to capture intricate patterns and nuances in the data that may not be discernible in the original feature space [15], [16].

Mathematically, SVR aims to learn a function $f(x)$ that predicts the output y for a given input x:

$$y = f(x) = \langle w, \phi(x) \rangle + b, \tag{15}$$

where w represents the weight vector, b is the bias term, and $\phi(x)$ denotes the feature mapping function that transforms the input data x into a higher-dimensional space. The angle brackets denote the inner product between w and $\phi(x)$, which is defined as:

$$\langle w, \phi(x) \rangle = \sum_{i=1}^{N} w_i \phi_i(x). \tag{16}$$

In SVR, the goal is to find the optimal values of w and b that minimize the error between the predicted output $f(x)$ and the true output y, subject to a specified tolerance margin ϵ. This leads to the following optimization problem:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \tag{17}$$

subject to the constraints:

$$y_i - \langle w, \phi(x_i) \rangle - b \leq \epsilon + \xi_i, \tag{18}$$

$$\langle w, \phi(x_i) \rangle + b - y_i \leq \epsilon + \xi_i^* \tag{19}$$

$$\xi_i, \xi_i^* \geq 0. \tag{20}$$

C is the regularization parameter that controls the trade-off between minimizing the error and maximizing the margin, ξ_i and ξ_i^* are slack variables that measure the deviation of the predicted output from the true output, and ϵ is the tolerance margin that determines the acceptable deviation.

To further enhance our analysis, we will also explore the use of Quantum QSVR in addition to classical SVR. This will allow us to compare the performance of both approaches and evaluate their effectiveness in predicting the zeros of the Riemann ζ-function.

IV. Quantum Support Vector Regression

A. Quantum kernels

To exploit the power of quantum computing and make use of its advantages, we must encode classical data $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ into a quantum state $|\psi\rangle$ [17]. In quantum computing, the quantum state $|\psi\rangle$, which fully describes the qubit, resides in the Hilbert space $\mathcal{H}$, providing a natural framework for defining a quantum kernel. The process of mapping classical data $\mathbf{x}$ to the quantum state $|\psi\rangle$ is achieved through the map function $\phi : \mathcal{X} \to \mathcal{H}$. Then, the quantum kernel is formulated as:

$$k(\mathbf{x}, \mathbf{x}') = |\langle \phi(\mathbf{x}) | \phi(\mathbf{x}') \rangle|^2, \tag{21}$$

where $|\phi(\mathbf{x})\rangle = \mathcal{U}_{\phi(\mathbf{x})}|0\rangle$, and the circuit $\mathcal{U}_{\phi(\mathbf{x})}$ encodes the classical data $\mathbf{x}$ into the quantum state $|\phi(\mathbf{x})\rangle$ using a unitary operator $\mathcal{U}$. The state $\langle \phi(\mathbf{x})|$ represents the dual vector of $|\phi(\mathbf{x})\rangle$, obtained by taking the adjoint of $|\phi(\mathbf{x})\rangle$, denoted as $\langle \phi(\mathbf{x})| = |\phi(\mathbf{x})\rangle^\dagger$. The kernel defined in Eq. (21) can be efficiently estimated by a quantum computer using the well known SWAP Test [18].

In our research, a classical data vector $\mathbf{x}$ consists of features containing the Riemann-Siegel Z-function and terms of the Riemann-Siegel series applied to two successive Gram points. To obtain a quantum representation of $\mathbf{x}$, we employ the feature map defined in [19]:

$$\mathcal{U}_{\Phi(\mathbf{x})} = \prod_d U_{\Phi(\mathbf{x})} H^{\otimes n}, \tag{22}$$

where

$$U_{\Phi(\mathbf{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\mathbf{x}) \prod_{k \in S} Z_k \right). \tag{23}$$

The number n of qubits is equal to the dimensionality of the classical data $\mathbf{x}$. The symbols are encoded through the coefficients $\phi_S(\mathbf{x})$, where $S \subseteq [n] = \{1, \ldots, n\}$ describes all possible connections of qubits in the quantum circuit. The encoding function is given by

$$\phi_S : x \mapsto \begin{cases} x_i & \text{if } S = \{i\} \\ (\pi - x_i)(\pi - x_j) & \text{if } S = \{i, j\} \end{cases} \tag{24}$$

and Z_k is the Z Pauli matrix acting on the k-th qubit.

For instance, a quantum circuit that implements $\mathcal{U}_{\Phi(\mathbf{x})}$ using a single-qubit Z rotation, two-qubit ZZ rotation and interactions between all qubit pairs will produce blocks of the form

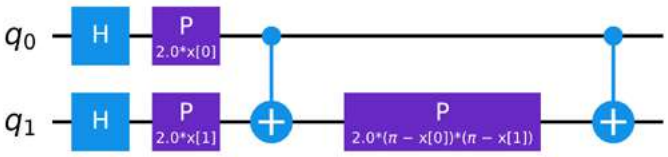


Fig. 1. Quantum circuit of $\mathcal{U}_{\Phi(\mathbf{x})}$ with $n = 2$ qubits, depth $d = 1$ and Pauli rotation $P = R_Z$.

Finally, it's worth mentioning that the main distinction between SVR and QSVR lies in the origin of the kernel. When the kernel is computed using quantum algorithms, known as a quantum kernel, it pertains to QSVR. Conversely, if the kernel is derived using classical methods, it relates to SVR.

V. Data set

A. Features data

While compiling our initial feature set, we drew inspiration from a relevant study by Shanker [6]. Our selected features for training the algorithms encompass the values of the Riemann-Segal Z-function, along with the first ten terms of the Riemann-Segal series (Eq. (9)), and nine terms where the cosine function is replaced by its corresponding sine function for consecutive pairs of Gram points. The exclusion of the initial sine term is justified by its consistent evaluation to zero at Gram points. Thus, the resulting size of our input feature set totals 40. Out of the 40 characteristics available for our simulations, we opted to utilize 10, as the selection of the quantum kernel for the algorithm limited us to no more than 10 features. The chosen features include the values of the Riemann-Segal Z-function, the first two terms of the Riemann-Segal series, and 2 terms where the cosine function is replaced

by its corresponding sine function for consecutive pairs of Gram points. The features were computed and arranged in the order described above.

B. Training and test data

For training and testing the algorithm, we utilized 500 data points corresponding to the Gram points and the zeros of the Riemann zeta function. The primary objective of machine learning regression is prediction, making predictive performance the key metric in evaluating machine learning models. To assess a model's generalizability and identify potential overfitting, it's crucial to evaluate the model using independent data, distinct from the training set. This process, known as data splitting, involves dividing the dataset into separate test and training sets. In our experiments, we employed an 80%/20% data split: 80% of the shuffled data was allocated for training, while the remaining 20% was reserved for testing purposes. The evaluation of the test data was based on the mean squared error (MSE), defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{25}$$

where y_i represents the true values of the dependent variable and $\hat{y}_i$ represents the corresponding model predictions. To obtain accurate data on the zeros of the ζ-function, we accessed the dataset provided by Odlyzko [21]. This dataset includes over 2 million zeros of the Riemann zeta function, each measured with an accuracy of 4×10^{-9}.

VI. Predicted Zeros

After training and testing the SVR with 10 features in its classical and quantum versions, we selected 50 random points from the test data to plot a graph showing the algorithm's predictions alongside the known distances of the ζ-function to Gram points. The results can be seen in the Fig. 2 and Fig. 3.

The algorithms, in their quantum and classical versions, were trained and tested. The features used were incremented according to what is described in subsection V-A, and the performances of both versions were compared, as can be seen in Fig. 4. We can see that the algorithm using the quantum kernel performed worse than its classical version. However, it's worth noting that the quantum algorithm shows considerable potential for improvement, given the vast possibilities for variations in entanglement gates. Other possibilities should be considered as well. Additionally, an asymptotic analysis of the algorithm's behavior as the dataset size grows was not conducted. An improvement in the performance of the quantum algorithm could be expected in this regard.

VII. Final remarks

As can be seen throughout the development this work, the quantum version of SVR can be viewed as an alternative for predicting the zeros of the ζ-function on the critical line. Despite exhibiting a relatively inferior performance compared to the classical version, the quantum version offers a greater

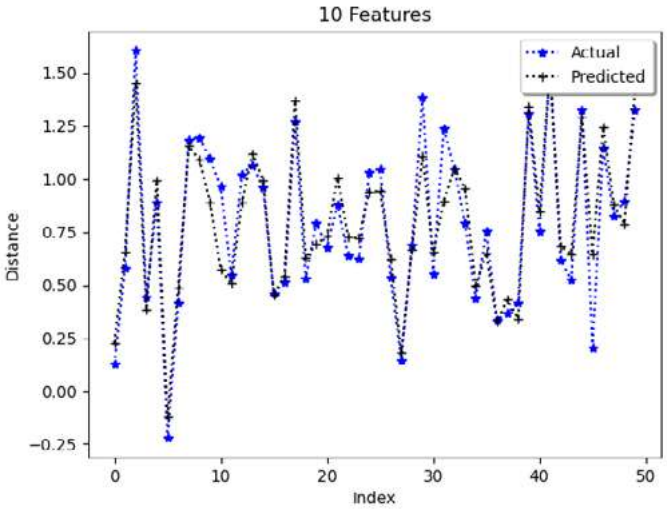


Fig. 2. Using 10 features, SVR predictions versus the actual distance $\gamma_n - g_{n-1}$ for 50 randomly selected observations from the test data.

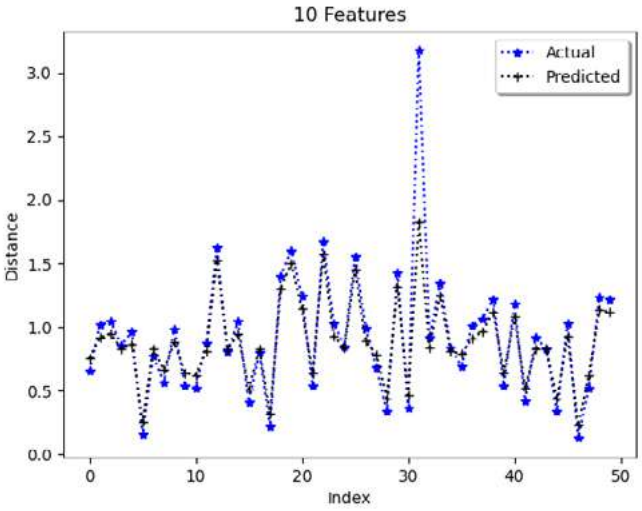


Fig. 3. Using 10 features, QSVR predictions versus the actual distance $\gamma_n - g_{n-1}$ for 50 randomly selected observations from the test data.

number of possibilities and variations. The challenges arising from the high complexities of the quantum version hinder the investigation of the problem but also provide a range of options for improving the version. A statistical analysis can be conducted on the features used by the quantum version, aiming to select the variables that offer better accuracy for the algorithm. Additionally, studies should be undertaken to enhance the performance of quantum algorithm simulations, as the simulations of quantum algorithms adopted for solving the problem make it impractical to conduct asymptotic analyses on the determination of the zeros of the zeta function.

References

[1] E. Bombieri. *The Riemann Hypothesis–official problem description.* Clay Mathematics Institute, 2000.

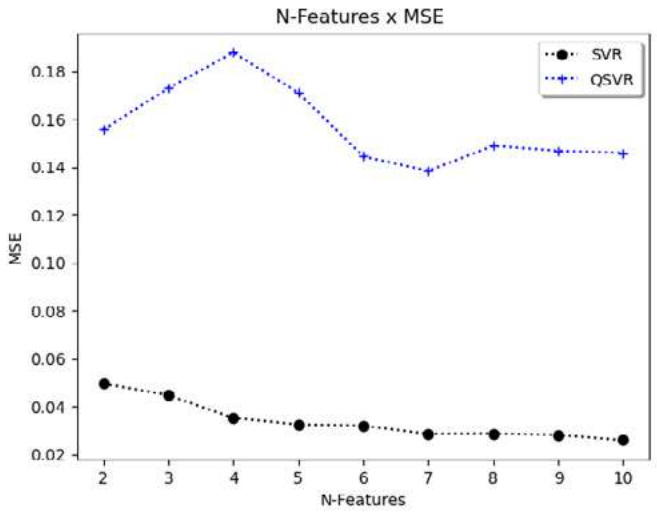


Fig. 4. The MSE decreases with the addition of features to the system.

[2] A. Awan. *On the Theory of Zeta-functions and L-functions.* Electronic Theses and Dissertations. 53. 2015. Available in https://stars.library.ucf.edu/etd/5

[3] G. N. Remmen. *Amplitudes and the Riemann Zeta Function.* Phys. Rev. Lett. 127, 241602 – Published 8 December 2021.

[4] M. A. Chaudhry, A. Tassaddiq. *A new generalization of the Riemann zeta function and its difference equation.* Adv Differ Equ 2011. Available in https://doi.org/10.1186/1687-1847-2011-20

[5] G. A. Hiary. *Fast methods to compute the Riemann zeta function*. Pages 891-946 from Volume 174, Issue 2, 2011.

[6] O. Shanker. *Neural Network prediction of Riemann zeta zeros.* Advanced Modeling and Optimization, 14(3), 717-728, 2012.

[7] J. Kampe, A. Vysogorets. *Predicting zeros of the riemann zeta function using machine learning: A comparative analysis.* 2018, available online: https://www. sci. sdsu. edu/math-reu/2018-2. pdf.

[8] Ran He, Ming-Zhong Ai, Jin-Ming Cui, Yun-Feng Huang, Yong-Jian Han, Chuan-Feng Li, Tao Tu, C. E. Creffield, G. Sierra, and Guang-Can Guo. *Identifying the Riemann zeros by periodically driving a single qubit.* Phys. Rev. A 101, 043402, 2020.

[9] J. Latorre, G. Sierra. *Quantum computation of prime number functions*. in Quantum information and Computation, 2014.

[10] M. McGuigan. *Quantum Computing and the Riemann Hypothesis*. arXiv2303.04602, 2023.

[11] W. van Dam. *Quantum Computing and Zeroes of Zeta Functions.*quant-ph/0405081, 2004.

[12] Stein, E. M., & Shakarchi, R. (2010). Complex analysis (Vol. 2). Princeton University Press.

[13] Hutchinson, J. I. (1925). On the roots of the Riemann zeta function. Transactions of the American Mathematical Society, 27(1), 49-60.

[14] Korolev, M. A. (2014). On small values of the Riemann zeta-function at Gram points. Sbornik: Mathematics, 205(1), 63.

[15] Smola, A.J., Schölkopf, B. *A tutorial on support vector regression*. Statistics and Computing 14, 199–222 (2004).

[16] Nello C., John N. S. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press; 2000:i-iv.

[17] Schuld, M.; Petruccione, F. *Supervised learning with quantum computers*. Cham: Springer, 2018.

[18] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf. *Quantum fingerprinting*. Phys. Rev. Lett. v. 87, 2001.

[19] Havlíček, V., Córcoles, A.D., Temme, K. et al. *Supervised learning with quantum-enhanced feature spaces*. Nature 567, 209–212 (2019).

[20] M. Nielsen, and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[21] A. Odlyzko. Tables of zeros of the Riemann zeta function. Available in https://www-users.cse.umn.edu/~odlyzko/zeta_tables/index.html

7º WECIQ

Using simulations to validate improvements over Shor's Algorithm

Fábio Santos and Luis Kowada

***Abstract*—There are already a bunch of works presenting a simulation of Shor's algorithm. However, the great majority lacks when it is run against large integers. It happens because the exponential characteristic of the problem requires a huge amount of resources. We propose an approach where we execute all steps of Shor's algorithm. Some steps are adapted to run faster in classical environment. This enabled us to execute a simulation against large integers and being able to get a relevant information. Through simulations, we could validate that using Jacobi symbol improves the Shor's algorithm to require only one order finding execution for some cases.**

***Keywords*—Number Theory, Quantum computing, Shor's algorithm.**

I. Introduction

Shor's algorithm [1] for factoring integers was a breakthrough in computer science as it showed how a quantum computer could be used to factor integers in polynomial time. This algorithm to factor an integer N is based on a quantum routine to find the order r of an element that belongs to a group Z_N^*. Even after almost 30 years, some aspects of the algorithm are still not perfectly understood. Shor proposes some approaches to estimate r from the results measured in the quantum routine, but testing the algorithm is not an easy task because the access to quantum computers is still scant nowadays. To solve this issue, simulations can be carried out on classical computers.

A simulation is more realistic when it presents the closest circumstances of what is intended to be simulated. If it was not possible to distinguish the simulated situation in relation to the real situation in all its details we would have a perfect simulation. In the context of algorithms, you may want to simulate its execution to check if it works or to extract information regarding its use. Simulating an algorithm is especially useful when it is not possible (or difficult) to execute it in a real situation. This happens, for example, for quantum algorithms. The difficulty of quantum simulations is that, in general, they require exponential time and space, which brings challenges to the simulation process. Some very relevant information can be obtained, for example, the success rate of probabilistic algorithms in the average case. Theoretical results show that Shor's algorithm [1] for factoring integers (QIFA - Quantum Integer Factorization Algorithm) has at least a 50% chance of success if the quantum routine returns the order r of an integer x modulo N, where r is a smaller natural number such that:

$$x^r \equiv 1 \bmod N \tag{1}$$

Fábio Santos, IC, Universidade Federal Fluminense. Este trabalho foi parcialmente financiado pelo projeto FAPERJ APQ1 (260003/015313/2021).

However, through some simulations executed in [2], we see different results for probability of success when testing prime numbers with close size and randomly chosen. Simulations can also be very useful not only to corroborate theoretical estimates or extract some information about the execution, but also to refine the algorithm in some cases where details of it have not been well defined or if there is more than one possible approach. In the case of the quantum factorization algorithm, Shor suggests some techniques to estimate the order from the data returned by the quantum routine [1]. The 1^{st} technique is to verify the observed state. If the observed state is $|c\rangle$, he suggests testing also the values $c \pm 1$, $c \pm 2$, and so on. For each possibility, r is estimated by approximating the fraction $c/q \approx d/r$, where q is the total number of possible states. If the quantum routine returns a peak or a position near to a peak, we should be able to estimate the value for r. We can also test as candidates for r, small multiples $2r'$, $3r'$, ... This is the 2^{nd} technique. As the 3^{rd} and last technique, given two candidates r_1 and r_2, we take lcm(r_1, r_2) as a candidate for r. Having the r value in hands, we can obtain the factors for a number N. Simulations could provide us with interesting information like that there is no need to try c values greater than 1 in the 1^{st} technique or that the 3^{rd} technique is something so rare to happen that it could be disregarded. Having this kind of information, customized versions of the algorithm can be proposed.

As we can see, simulations can be very helpful. Some works [3], [6] have already proposed simulations for Shor's algorithm, however, they have used few bits. Other works, as [2], have already run simulations against large numbers, but did not execute the simulation as close to the real scenario as we would like. In this work, we propose a simulation able to factoring large numbers executing the steps as closer as possible to the steps proposed in [1]. We also purpose a theorem about using Jacobi symbol to achieve a similar result of [8] with a wider set of numbers.

The rest of this work is divided in three more sections. In the second section, we will show different kinds of simulations to test Shor's algorithm, one more fine-grained and other more coarse-grained, the third shows the details of the proposed simulation by this work and its results. Finally, in the last section, we conclude this work with insights and results obtained.

II. Simulations Approaches

As we saw in the introduction section, simulations can be very helpful to understand and study some classes of algorithms, including quantum algorithms. A simulation of a quantum program would be more realistic when it reproduces all steps executed by an algorithm in a quantum computer and stores all the necessary information. This type of simulation is possible only for programs involving a few bits, as this type of simulation has an exponential cost. If the purpose of the simulation is to verify whether the program quantum is correct,

it would be enough to be able to reproduce its outputs along with the expected probabilities, regardless of the simulation steps. In this context, a more perfect simulation would pass through each intermediate state until reaching the final state and, if possible, spend the same processing time at each step and store all information generated by these steps (fine-grained simulation). But instead of storing all information generated by the algorithm steps, like using a matrix representing these data, we could go from the initial to the final state without going through all the states and even so we would have a satisfactory simulation (coarse-grained). We can say that there are several granularity levels of simulation for a procedure. If the purpose of the simulation was to analyze the propagation of noise in the system, the states intermediaries are important in this case, a simulation that goes from first to the last state (widest level of granularity) might not be useful.

In the case of Shor's algorithm [1] for factorization, we can consider the routine of order estimation (QOFA - Quantum Order Finding Algorithm) as a black box, as was done in [2] or simulate smaller steps of the algorithm. Considering the quantum routine as a black box that returns the order of a

number, we can obtain simulation results of arbitrarily large numbers as those used in RSA, for instance, 1024 bits or more. In this way, we can get an estimate of how many successful QOFA executions are necessary, but not how many executions would actually be necessary, since some QOFA executions may not bring relevant information to the problem.

Shor's algorithm simulations that include the QOFA simulation have very few bits. See, for example, [3], [4] with N being a number with few bits. There is also [6] where they could execute QOFA with numbers about 70/80 bits but they had to provide a large hardware to accomplish their results. The simulation of the algorithm with small numbers may not be realistic in the sense of being a degenerate situation. For example, if the number is very small, there is a high chance of finding a factor randomly. In addition to the level of granularity, another important property is the set of information that simulations use. In general, the strictest simulations, with finer granularity, do not use information other than the inputs of the problem. But there are situations where other information can help with the simulation. For example, it is only possible to simulate factorization for arbitrary wide numbers, knowing their factors in advance, as happens in simulations in [2]. Knowing and using this information does not invalidate the simulation, if the results obtained are similar to those in the real situation.

The purpose of this work is the simulation of Shor's factorization algorithm including the QOFA simulation, so that the values obtained are similar to the values obtained if this routine were executed on an ideal quantum computer. The advantage of this simulation in relation to existing simulations of the finer-grained Shor algorithm is that it allows the simulation of factoring larger numbers and in relation to the simulations in [2], it is more realistic. The objective of these simulations is to show the behavior of the algorithm of factoring and estimate the number of necessary executions of the QOFA, using different strategies.

III. Proposed simulation

As we mentioned in earlier sections, when we are simulating some algorithm, we can go from a fine-grained approach to another coarse-grained one. There will be pros and cons of choosing one of them. This is certainly also true when applying some simulation for Shor's algorithm. A simulation should try to reproduce the steps as closely as possible from the real ones. The list below shows how we simulate Shor's algorithm step-by-step:

1) Check if number N is a prime or prime power;
2) If it isn't, go to step 4;
3) Return the prime number and its power;
4) Pick a random number x, where x must be $0 < x < N$;
5) Check if gcd(x,N) result is 1;
6) If it is 1, go to step 8;
7) Return gcd(x,N) and N/gcd(x,N);
8) Execute quantum routine to find the order r (quantum order finding);
9) Execute post-processing and validate r as the order value for x;
10) If order finding has failed, go back to step 4;
11) Return gcd(($x^{r/2} - 1$),N) and gcd(($x^{r/2}+1$),N);

In our work, we are more interested in analyzing what happens in post-processing depending in what x value is randomly picked in step 4. At this point, it should be clear that we want to execute simulations for values N where N is a large integer (at least greater than 128 bits). In this way, if we can suppress some steps or even make them less detailed to accomplish our goal, we should do that. However, suppressing some steps or simplifying them should not compromise the results. Therefore, we would like to show that using the coarsegrained approach will not lose quality when compared to a more fine-grained one. To accomplish this, we created a configurable simulation where we can switch on/off the finegrained option and, after that, compare results.

Both simulations, coarse-grained and fine-grained, execute the steps as shown in the step-by-step list. The differences between them are isolated in step 8. When we switch on the fine-grained simulation, the first thing we do in this step is build a structure data, such as an array, where we set period markers. The period is defined by the x element's order calculated during this step. Each position of this structure represents a quantum state and these position values should be between N^2 and $2N^2$ [1]. After this, we execute a Fast Fourier Transform on these data. The Fast Fourier Transform will set amplitudes for each of these quantum states and we can then calculate the probabilities. We use the calculated probabilities to pick one of these simulated quantum state positions and execute the post-processing step.

Executing step 8 as described before in a fine-grained approach requires a great amount of memory and processing time. It requires a great amount of memory because as the number N becomes greater, the number of states grows exponentially. This also happens with processing time to find the x element's order. There is no known polynomial algorithm to do that. Because of this, if we also want to execute

simulations for large integers, we have to switch off the fine-grained option.

In our coarse-grained approach, we do not use a data structure to store information about period markers and we also do not execute Fast Fourier Transform aiming to get amplitudes. We, instead, calculate directly some simulated quantum state position related to a peak. As Shor describes in his work [1], when a Quantum Fast Fourier Transform is executed over a register in superposition, the output generated by this procedure is a set of amplitude data with some points with higher amplitudes (peaks). The number of peaks is equal to r, where r is the x element's order. We can establish a relation between the number of states and the number of peaks. We are going to call this offset. If we know the number states and the r value, we can calculate the quantum states that correspond to a peak. This way, we replace the quantum order finding routine and calculate some peak values randomly. It solves the problem related to memory space (memory) but not the processing time one (order algorithm). The solution for the order problem is based on restrictions we set in our simulation. We use only safe primes during our simulations. This restriction should not be a problem as we are going to see in simulation results. Through simulation results, we could see that the chance of success when running the test of Shor, for x chosen at random and N being a product of two safe primes, is approximately 50%. This is equal to the lower bound described by Shor when finding the nontrivial factors of N [1]. In other words, this would be the worst case for the Shor's algorithm. Another work we can mention is [8]. In this work, the author improves the Shor's algorithm to require approximately only one execution of QOFA when using safe primes as factors of N.

The number N used in our simulation is composed of two safe primes. A safe prime is a number in the format $p = 2p' + 1$, where p' is also a prime number. By Lagrange's theorem, when we get an element x from a multiplicative group Z_N^*, the element's order must divide the group's order. A group's order is the number of elements of a group. It can be calculated by Euler's Totient Function. The value of Euler's Totient Function for a prime number p is $p-1$ in Z_p. If we combine these two definitions, safe primes and group's order, we can see that 2 divides the group's order when the group is generated modulo a safe prime ($p' = (p - 1)/2$). It makes the possible values for these group's order the small set of values 1, 2, p', $2p'$. To finish the approach, we need one last additional information. When we have a number $N = p \cdot q$, the order r of some group Z_N is lcm(rp,rq), where rp is the order for a element x in a group Z_p^* and rq is the order for a element x in a group Z_q. It makes our possible group's order values go from 1, 2, p', $2p'$ to 1, 2, p', q', $2p'$, $2q'$, $2p'q'$. So, instead of executing an algorithm to find the element's order, we only check the values in this group.

A. Comparing fine-grained against coarse-grained

As we mentioned before in the section 3, we want to execute the simulation for large values and to do that, we should execute the coarse-grained option. It implies suppressing some steps or simplifying them without compromising the results. To check if the coarse-grained option does not compromise the results, we must execute both simulations and compare them. To help us making these comparisons, we defined some classifications for simulation results:

TABLE I

Classification kinds.

Kind	Description
LUCKY-RANDOM-VALUE	The x random value obtained is factor of N
GOOD-ROUNDING	The value returned by quantum order find simulation (peaks) was used to get factors without adjustments
SHOR-1	The value returned by quantum order find simulation (peaks) had to be adjusted for close values to get factors
SHOR-2	The value returned by quantum order find simulation (peaks) divides the x element's order
SHOR-3	The x element's order is obtained from lcm between two quantum order find executions
NO-SOLUTION-FOUND	The simulation was not able to get factors of N

We arrange the set of values 23, 47, 59, 83, 179 and create composite numbers through combinations of 2 elements of this set. After that, we execute both simulations. We chose this set for two reasons. The first one is because they are safe primes and even though this is not mandatory for the fine-grained approach, we must execute the coarse-grained option only with numbers composed of safe primes. The second one is because as the fine-grained option requires a lot of computing resources, we must avoid large integers in this situation.

After defining a way to compare both simulations, we executed them and get the following results: GOOD-ROUNDING - 48% of results were classified as GOOD-ROUNDING in coarse-grained against 46% in fine-grained simulation LUCKY-RANDOM-VALUE - 11% of results were classified as LUCKY-RANDOM-VALUE in coarse-grained against 6% in fine-grained simulation - SHOR-2 - 40% of results were classified as SHOR-2 in coarse-grained against 46% in finegrained simulation - SHOR-1 and SHOR-3 - 1% of results were classified as combination of SHOR-1 and SHOR-3 in coarse-grained and 2% were classified as SHOR-3 in finegrained simulation - NO-SOLUTION-FOUND - There was no result classified as NO-SOLUTION-FOUND.

Analyzing the results of both simulations, we can see that there is no great qualitative differences between them and both were able to find the factors for all inputs. Therefore, we can consider that coarse-grained option does not compromise the results and we can use it to execute simulation for large numbers.

TABLE II

Classification results when running simulation. The input set is a combination of 2 elements of numbers 23, 47, 59, 83, 179.

classification kind	fine-grained	coarse-grained
GOOD-ROUNDING	46%	48%
LUCKY-RANDOM-VALUE	6%	11%

SHOR-2	46%	40%
SHOR-3	2%	0%
SHOR-1 and SHOR-3	0%	1%

B. Executing against large numbers

Besides the qualitative results, we also generated a set of large numbers with 128, 512 and 1024 bits, where factors are safe primes. Using the openssl application [5], we could generate these safe primes through the command *openssl prime -bits number-of-desired-bits -safe -generate*. The parameter *number-of-desired-bits* was set with values 64, 256 and 512 respectively. We generated 820 composed numbers with these safe prime factors. For all composed number in the list, we executed the simulation until it returns the factors for the current composed number.

TABLE III

Large numbers factoring for x randomly chosen. SAA STANDSFOR SUCCESSFULLYATTEMPTSAVERAGEAND MSA STANDSFORMAX SUCCESSFULLYATTEMPTS.

bits	SAA	MSA
128 bits	2.01	10
512 bits	2.00	9
1024 bits	2.07	10

Through simulation results, we can see that they are totally compliant with the theoretical results [1] and other simulation results [2]. Using the original Shor's proposal [1], it is necessary run, in average, twice to get the factors of some composed number N. Another interesting result we can see is that the maximum number of tries is closer to 10 as in [2].

It is important to emphasize that our simulation simplify some steps from Shor's algorithm. However, the entire postprocessing procedure is executed as expected in a real situation. This way, we can focus in the order estimation and how the choice of the random value can influence in results for large numbers with until 1024 bits.

C. Using Jacobi Symbol approach

Jacobi Symbol is a generalization of Legendre symbol. Both belongs to the field of number theory. Jacobi Symbol is also known as quadratic residue. The concept behind quadratic residues is that given two co-prime integers x and p, x is a quadratic residue modulo p if there is some y value that satisfies $y^2 \equiv x \bmod p$ (Jacobi Symbol = 1). Au contrair, if there is no y value that satisfies this congruence, we say x is a quadratic non-residue modulo p (Jacobi Symbol = -1).

Theorem 1: Let $N = p \cdot q$, where p and q are prime numbers. Further, let $p = 2^c \cdot u+1$ and $q = 2^c \cdot v +1$, where u and v are integers and c is a constant. For all $x \in Z_N^*$, where $gcd(x,N) = 1$ with $x > 1$, we have:

$$\text{If Jacobi Symbol}(x, N) = -1 \text{ then}$$
$$1 < gcd(x^{r/2} - 1, N) < N$$

Proof: The Jacobi Symbol (JS) of x with $N = p \cdot q$ can be represented as a product of two Legendre Symbol (LS) of x with p and y with q. In this way, given any integer x and any positive odd integer N, we can substitute $JS(x,N)$ for $LS(x,p) \cdot LS(x,q)$, where $N = p \cdot q$ with p and q being odd primes. By definition of Legendre Symbol, we know that the result of $LS(a,b) \in \{-1,0,1\}$. So, if we have $JS(x,N) = -1$, then $\{LS(x,p) = 1$ and $LS(x,q) = -1\}$ or $\{LS(x,p) = -1$ and $LS(x,q) = 1\}$.

Without loss of generality, we suppose $LS(x,p) = 1$ and $LS(x,q) = -1$. In this case, with $LS(x,p) = 1$ we have $r_p = ord(x,p)$ divides $\frac{p-1}{2} = 2^{c-1}u$, where $ord(x,p)$ is the order of x modulo p. Then $2^c / | r_p$ and $\gcd(2^c, r_p) < 2^c$.

On the other hand, $LS(x,q) = -1$ implies that $r_q = ord(r,q)$ does not divide $\frac{q-1}{2} = 2^{c-1}v$. It means $2^c \mid r_q$.
In other words, $\gcd(2^c, r_q) = 2^c$.

Taking into account that $r = ord(x,N) = lcm(r_p, r_q)$ (see Lemma 2.3 in [2]), so we have $\gcd(r, 2^c) = 2^c$. Therefore, $r/2$ is multiple of r_p, but $r/2$ is not multiple of r_q. In this way, $x^{r/2} \equiv 1 \bmod p$ but $x^{r/2} \not\equiv 1 \bmod q$. It means that $x^{r/2} - 1$ is multiple of p but is not multiple of q. As $N = p \cdot q$ then $\gcd(x^{r/2} - 1, N) = p$.

■

TABLE IV

Large numbers factoring for x randomly chosen but using Jacobi Symbol to select them. SAA STANDSFORSUCCESSFULLYATTEMPTSAVERAGEAND MSA STANDSFORMAXSUCCESSFULLYATTEMPTS.

bits	SAA	MSA
128 bits	1.00	1
512 bits	1.00	1
1024 bits	1.00	1

Taking into account this theorem, if we test the random value used in QOFA routine and call the QOFA only when this random value has Jacobi Symbol = -1, we should always be able to get factor for some N value. In [7], the author proposes a improvement in Shor's algorithm using Jacobi Symbol to evaluate the random value that is used by QOFA. Through our simulation approach, we can verify this behavior for large numbers. In fact, we can see there is an improvement in the success rate. The success rate is even better than that mentioned by this work (from 3/4 to 1). However, as we have already mentioned, we used only safe primes in our simulations.

IV. CONCLUSIONS

Our target was to build a simulation where we could test Shor's algorithm executions for large numbers and also trying to verify some already known assumptions and get new information. In fact, we were able to verify some already known results such as the average number of executions necessary to obtain the factors of a composite number and also verify new stuff. For instance, we could verify that using Jacobi symbol to select the x random value in Shor's algorithm makes a improvement in the QOFA's success rate. When we do that, the procedure requires the QOFA routine be executed only once. It is a really good result. This result is very similar the result achieved by [8]. However, based on theorem 1, our proposal is wider because it is not restricted to
$N = p_1 \cdot p_2 = (2q_1+1)(2q_2+1)$, with $q_1 \not= q_2$ and $q_1, q_2 > 2$. Instead, our result proposes $N = p \cdot q = (2^c \cdot u +1)(2^c \cdot v +1)$, where u and v are integers, with c being a constant.

REFERENCES

[1] Shor, Peter W, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM review, 41, 2, 303–332, 1999, SIAM

[2] Chicayban Bastos, Daniel and Kowada, Luis Antonio, How to detect whether Shor's algorithm succeeds against large integers without a quantum computer, Procedia Computer Science, 195, 145–151, 2021, Elsevier

[3] David S Wang, Charles D Hill, and Lloyd CL Hollenberg. Simulations of Shor's algorithm using matrix product states.Quantum Information Processing, 16:1–13, 2017.5

[4] Tankasala A, Ilatikhameneh H. Quantum-kit: simulating shor's factorization of 24-bit number on desktop. arXiv preprint arXiv:1908.07187. 2019 Aug 20.

[5] OpenSSL Homepage, https://www.openssl.org. Last accessed 22 Oct 2023

[6] Willsch D, Willsch M, Jin F, De Raedt H, Michielsen K. Large-Scale Simulation of Shor's Quantum Factoring Algorithm. Mathematics. 2023 Oct 9;11(19):4222.

[7] Leander G. Improving the Success Probability for Shor's Factoring Algorithm. arXiv preprint quant-ph/0208183. 2002 Aug 29.

[8] Grosshans F, Lawson T, Morain F, Smith B. Factoring safe semiprimes with a single quantum query (2015).

7º WECIQ

HHL: Estado da Arte, Limitações e Melhorias

Lucas Amaral e Luis Kowada

Resumo—Este trabalho apresenta um algoritmo para solucionar a versão quântica de sistemas de equações lineares chamado *HHL*, salientando suas limitações quanto a sua praticidade. Será introduzido o tipo de problema e descrita as partes do algoritmo. Em seguida, será discutida algumas das restrições que tem que ser levada em consideração para que o ganho computacional se concretize. Por último, expomos melhorias na complexidade que houveram desde então. Dessa forma, este trabalho se apresenta como uma breve revisão da literatura do *HHL*.

Palavras-Chave—HHL, Problema de Sistema Linear Quântico, Simulação hamiltoniano.

Abstract—This work presents an algorithm to solve the quantum version of systems of linear equations called *HHL*, highlighting its limitations in terms of its practicality. The type of problem will be introduced and each part of the algorithm will be described. Next, we will discuss some of the restrictions that have to be taken into consideration for the computational gain to materialize. Finally, we expose improvements in complexity that have occurred since then. With this, this work present itself as a brief literature review on *HHL*.

Keywords—HHL, Quantum Linear System Problem, Hamiltonian simulation.

I. Introdução

Sistemas de equações lineares são ferramentas importantes em diversas áreas da ciência. O melhor algoritmo clássico para solucionar este tipo de problema é o Método de Gauss com $O(n^3)$, o qual nos é retornada a solução exata, e o método do Gradiente Descendente com número de operações crescendo linearmente em n, o qual devolve uma solução aproximada. Em 2009, foi apresentado o algoritmo *HHL* para a solução de um problema *QLSP* (*Quantum Linear System Problem*) que retorna um estado quântico cuja solução é normalizada e codificada nas amplitudes de um registrador, com *speedup* exponencial em relação aos algoritmos clássicos [1].

O *HHL* tem complexidade $O(\log n s^2 \kappa^2/\epsilon)$, isto é, logarítmico em função de n. A melhora da eficiência na solução do problema não é obtida sem ressalvas. Algumas restrições são requeridas para a obtenção do *speedup*. Por exemplo, a matriz de coeficientes deve ser esparsa – o s da complexidade mencionada acima é a esparcidade da matriz – e o número condicional κ dever ser mínimo – κ é a razão entre o maior e o menor autovalores da matriz de coeficientes [1].

Nas próximas seções, será explorado mais aprofundadamente o algoritmo *HHL*. Na Seção II, será apresentado cada parte do algoritmo. Na Seção III, iremos analisar, brevemente, algumas restrições do algoritmo e como estas se relacionam com a reivindicação de ganho exponencial em eficiência. Na Seção IV, atualizaremos algumas melhoras do algoritmo *HHL* original que foram formuladas ao longo dos anos desde a sua invenção.

Lucas Amaral, IC-UFF, Niterói-RJ, e-mail: amarallucas@id.uff.br; Luis Kowada, IC-UFF, Niterói-RJ, e-mail: luis@ic.uff.br.

II. O algoritmo *HHL*

O problema, essencialmente, consiste em obter a solução de um sistema de equações lineares na forma:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases} \quad (1)$$

O mesmo sistema de equações também pode ser representado pela forma matricial. Neste caso, temos, como entrada, uma matriz, por exemplo, A, e um vetor b. Dessa maneira, temos as seguintes representações:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \quad |b\rangle = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad (2)$$

A ideia geral do algoritmo é a seguinte:

Entrada: A entrada do algoritmo é um sistema de equações lineares representada por um operador hermitiano A, isto é, ($A = A^\dagger$) e um vetor representado pelas amplitudes de um estado quântico $|b\rangle$. Visto que não é provável que sistemas de equações lineares tenha tal configuração, é apresentado um truque que flexibiliza essa restrição. Caso A não seja uma matriz hermitiana, é possível transformá-la expandindo a matriz e os vetores de entrada com o seguinte truque [1]:

$$C = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \quad (3)$$

o qual C seria a nova matriz hermitiana que poderia ser resolvida na nova forma $Cy = \begin{pmatrix} \vec{b} \\ 0 \end{pmatrix}$ para obter $y = \begin{pmatrix} 0 \\ \vec{x} \end{pmatrix}$. Vamos assumir que A, neste trabalho, já seja hermitiana ou transformada pelo método acima. Assim, queremos resolver o sistema de equações lineares na forma:

$$A|x\rangle = |b\rangle \quad (4)$$

Almejamos aplicar operadores quânticos até chegar a configuração:

maneira que os autovalores sejam codificados pelos *qubits*. A porta $RY(\theta)$ é mostrada na equação 13.

$$RY(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad (13)$$

O que almejamos é seguinte transformação:

$$|0\rangle \rightarrow \sqrt{1 - \frac{1}{\lambda_j^2}}\,|0\rangle + \frac{1}{\lambda_j}|1\rangle \quad (14)$$

A partir desta equação podemos ver que, se o *qubit* auxiliar é $|1\rangle$, obtemos uma inversão do autovalor. Para preparar este estado definimos $\theta = 2\arcsin\frac{1}{\lambda}$.

Após essas etapas, aplicamos a computação inversa (*uncomputation*). Utilizamos a QPE^{-1} para obter o resultado da multiplicação entre $\lambda_i^{-1}|u_i\rangle\langle u_i|$ no primeiro registrador, e $|b\rangle$ codificado no segundo, resultando em $|x\rangle$ no segundo registrador.

III. Limitações do Algoritmo

Apesar de, atualmente, o *HHL* ser reconhecido como uns dos algoritmos recentes mais promissores da computação quântica, tal reputação não chega sem desconfiança. Em [2], é apontado alguns problemas do *HHL* que podem influenciar no ganho vultuoso de rapidez do algoritmo.

As primeiras limitações do *HHL* são as preparações da porta unitária U e do estado $|b\rangle$ de maneira que a sua preparação não afete o ganho obtido. Isto pode ser feito para casos específicos o qual exploraremos nesta Seção. A preparação de U é feita a partir da simulação hamiltoniana de [3], enquanto a preparação de $|b\rangle$ é feita a partir de um algoritmo apresentado por [4] para criar estados a partir de uma lista que forma uma distribuição de probabilidade. Por outro lado, os autores aventam a possibilidade do *HHL* ser usado como uma subrotina de outros problemas que já entregam os estados prontos.

A. Simulação Hamiltoniana

1) O Hamiltoniano: Na mecânica quântica, representamos um procedimento a partir de matrizes unitárias. As matrizes unitárias são transformações lineares de um estado quântico de tempo discreto. A evolução do sistema pode ser representado como:

$$|\psi\rangle \rightarrow U|\psi\rangle \quad (15)$$

Na física, no entanto, o tempo é tratado como um aspecto contínuo, isto é, ao invés de pularmos de um estado $|\psi\rangle$ para um estado $U|\psi\rangle$, esse "pulo"é tratado como um processo contínuo feito em um intervalo de tempo. Assim, definimos o hamiltoniano como gerador de operadores unitários a partir de um tempo instantâneo. Os hamiltonianos são sempre representados como matrizes hermitianas, isto é, dado um hamiltoniano H, temos que $H = H^{\dagger}$. O hamiltoniano não é, necessariamente, uma matriz positiva semi-definida, ou seja, pode ter autovalores negativos [5].

A equação de *Schrödinger* é apresentada em 16:

$$i\frac{d}{dx}|\psi\rangle = H|\psi\rangle \quad (16)$$

em que H é um hamiltoniano. Ao assumirmos que H independe do tempo, podemos derivar, a partir da equação 16, que o estado do sistema, após o tempo t, é:

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle \quad (17)$$

Dessa forma, transformamos uma equação diferencial, a qual solucionaríamos para cada coordenada do vetor $|\psi\rangle$, em uma equação a qual exponenciamos uma matriz como se fosse um escalar [5].

Exponenciar uma matriz pode ser um conceito não usual, mas podemos verificar que, para qualquer matriz diagonal, a solução é a exponenciação dos termos da diagonal. Isto é:

$$exp\left(\begin{bmatrix} \lambda_0 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}\right) = \begin{bmatrix} e^{\lambda_0} & & \\ & \ddots & \\ & & e^{\lambda_n} \end{bmatrix} \quad (18)$$

Então, o problema se torna representar o hamiltoniano como uma matriz diagonal. Sabemos que qualquer matriz hermitiana pode ser decomposta em $VDV^{\dagger}$ pelo Teorema Espectral. Dessa forma, podemos escrever e^{H} como $V\,e^{D}V^{\dagger}$. Com isso, para exponenciar a matriz, é preciso, primeiramente, decompôla [5].

No entanto, no contexto do *HHL*, decompor a matriz para codificar o hamiltoniano se torna inviável visto que almejamos uma complexidade de O(log$n$). Como podemos melhorar isso? Existe métodos de simular esse hamiltoniano eficientemente?

2) O Problema da Simulação: Então, como visto na Seção anterior, representamos a dinâmica de um sistema quântico a partir da equação de *Schrödinger* e a solução do sistema, dada uma matriz hermitiana independente do tempo, é $e^{iHt}|\psi_{inicial}\rangle$.

$$i\hbar\frac{d}{dx}|\psi\rangle = H|\psi\rangle \Rightarrow |\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle \quad (19)$$

O problema é que exponenciar uma matriz como e^{-iHt} pode ser computacionalmente difícil, mesmo com a matriz sendo esparsa, o cálculo pode ser exponencial. No entanto, há instâncias às quais o hamiltoniano pode ser escrito como uma combinação linear de termos locais o qual cada termo atua sobre parte do sistema [6][7].

$$H = \sum_{l=1}^{L} H_l \quad (20)$$

O hamiltoniano é chamado k-local quando H atua sobre m *qubits* e cada H_l atua, não trivialmente, em, no máximo, k *qubits*. É assumido que, cada H_l também é hermitiano e pode ser

exponenciado diretamente, isto é, é uma evolução que pode ser construída como um circuito quântico e executar em um computador quântico [7]. Fisicamente, este conceito também é muito importante, pois a maioria dos hamiltonianos que ocorrem na natureza são *k*-local.

Daí, podemos concluir que é possível decompor um hamiltoniano em um produtório das evoluções dos termos locais. Além disso, como cada expoente pode ser representado como um circuito quântico, o problema é resolvido para todo sistema.

$$e^{-iHt} = e^{-iH_1 t} e^{-iH_2 t} \dots e^{-iH_L t} \quad (21)$$

$$\text{se } [H_i, H_j] = 0$$

Sendo a restrição $[H_i, H_j] = H_i H_j - H_j H_i = 0$, ou seja, se todos os termos locais comutam entre si. Isso ocorre pois esta equação é derivada da fórmula binomial que gera produtos como $H_1 H_2 H_1$ que é diferente de $H_1^2 H_2$ [6].

Assim, o problema de simular um hamiltoniano pode ser descrito da seguinte maneira:

Problema 1.1. Dado um hamiltoniano H, uma matriz quadrada hermitiana $2^n \times 2^n$, agindo sobre sobre *n qubits* no tempo t, com erro ϵ. O objetivo é encontrar a sequência de portas quânticas que implementam uma evolução em função do tempo U cuja norma da diferença entre a simulação e a evolução ideal seja, no máximo, ϵ [6]. Ou seja:

$$|U - e^{iHt}| \leq \epsilon \quad (22)$$

Se $[H_i, H_j] \neq 0$, o produtório de termos locais não se mantém. Neste caso, é usada outra ferramenta mostrada a seguir.

3) Troterização: Para o caso geral, quando os termos locais não são comutativos entre si, pode ser utilizada a *Trotterização*. A fórmula de *Trotter* dá um limite de quantas exponenciações são possíveis para termos locais não comutativos. A fórmula é dada em 23.

$$e^{A+B} = \lim_{n \to 0} (e^{A/m} e^{B/m})^m \quad (23)$$

neste caso, m é o numero de iterações. Esse valor deve ser atribuído visando o número de iterações necessárias para que o erro da simulação atinja o valor desejado de ϵ, dado em 22. A e B são matrizes hermitianas que atuam sobre k *qubits* com $k < n$. Dessa maneira, a simulação hamiltoniana que, a partir da decomposição em hamiltonianos locais A e B, pode ser feita em m iterações. É importante salientar que existe variações de 23, aqui é mostrada a fórmula de *Trotter* de 1ª ordem.

4) Decomposição de Hamiltoniano: Sabe-se que as matrizes de *Pauli* formam uma base ortonormal de matrizes hermitianas cujas dimensões são potências de 2. Consequentemente, dado um hamiltoniano, podemos decompô-lo em matrizes de *Pauli*. Dessa maneira, o problema de simular o hamiltoniano se resume a capacidade de conseguirmos decompor eficientemente um hamiltoniano em termos locais, o que incluem matrizes de *Pauli*.

Uma maneira imediata de obter a decomposição de hamiltonianos é testando todas as possibilidades de produtos entre as matrizes de *Pauli*. Assim, podemos identificar a composição do hamiltoniano, mas há um custo exponencial de comparações que torna o ganho do *HHL* dispensável.

O que queremos é um método de decomposição que seja, pelo menos, tão bom quanto o custo do *HHL* em função da dimensão do hamiltoniano, isto é, uma função logarítmica com o tamanho número de colunas da matriz de entrada.

Muitos trabalhos apostam na representação do hamiltoniano como um grafo para fazer a decomposição em hamiltonianos locais. A ideia é decompor este grafo em múltiplos grafos mais simples a partir da coloração de arestas [3][8].

Em [3] é apresentado um método para decompor eficientemente hamiltonianos representados por matrizes hermitianas esparsas. O procedimento decompõe a matriz de entrada em $6s$ matrizes 1-esparsa. De [9] sabe-se que podemos aplicar $e^{-iH_j t}$ diretamente para matrizes 1-esparsa. Daí, podemos utilizar essa informação como *black-box* para implementar a decomposição do hamiltoniano, pelo método mostrado em [3], com custo $O(\log^* n)$.

Mais especificamente, o algoritmo consiste em representar o hamiltoniano como um grafo não direcionado e construir coloração de arestas de maneira que arestas incidentes ao mesmo vértice tenham cores distintas. A coloração é feita diretamente a partir de um ordenação das arestas incidentes aos vértices, isto é, se o vértice a é o i-ésimo vizinho de b e b, o j-ésimo vizinho de a, a cor da aresta é tida, provisoriamente, como o par (i,j). Haverá casos em que a associação dos pares desmantelará a coloração de arestas, por isso, também é usado um índice v, construída a partir de um procedimento que define v a partir da comparação dos índices dos vértices baseado em [11].

Esse tipo de decomposição, no entanto, adiciona uma restrição ao algoritmo. Da sua execução, é adicionada a dependência de s na complexidade do *HHL*. A decomposição em termos locais apenas será feita eficientemente se o hamiltoniano for representado como uma matriz esparsa.

Outros procedimentos foram apresentados, posteriormente ao *HHL*, que melhoram a dependência de s na complexidade. Há uma redução, apresentada por [10], que permite generalizar o problema de qualquer hamiltoniano para grafos bipartidos e aplicar a coloração de arestas para este caso. Outra maneira de fazer isso é dada por [8]. Este consiste em decompor o hamiltoniano em $6s$ galáxias, a partir da decomposição de s florestas, para isso, se utiliza do algoritmo dado em [12], que constrói uma floresta a partir da coloração, não necessariamente própria, das arestas.

B. Preparação de $|b\rangle$

Outra limitação do algoritmo diz respeito a preparação eficiente de $|b\rangle$. Como transformar de maneira eficiente $\vec{b}$ em

um estado quântico cujas amplitudes codificam cada b_i normalizado. Assim, como a simulação da forma e^{-iHt} descrita anteriormente, [2] aponta a preparação de $|b\rangle$ como uma ressalva da alegação "*HHL* soluciona $Ax = b$ em tempo logarítmico".

Em [2], é apontado que b precisa ser rapidamente carregado na memória de um computador quântico e que, em teoria, isso poderia ser feito através de um *QRAM*, ou *Quantum RAM*, i.e., uma memória que armazena valores clássicos e os possibilita serem lidos uma vez, em superposição. Outra alternativa seria se $\vec{b}$ fosse descrito por uma fórmula explicita, então, o computador quântico poderia calculá-lo para o próprio utilizá-lo.

No artigo original [1], a preparação de $|b\rangle$ é terceirizada para o trabalho de [4]. Neste trabalho, é apresentado um procedimento para gerar uma superposição de estados quânticos a partir de uma distribuição de probabilidades com a restrição de que esta distribuição seja eficientemente integrável.

A ideia é gerar eficientemente uma superposição quântica a partir da distribuição de probabilidade p_i. Tal superposição é mostrada em 24.

$$|\psi(\{p_i\})\rangle = \sum_i \sqrt{p_i}|i\rangle \qquad (24)$$

Perceba que, na mecânica quântica, o quadrado da amplitude é tido como a probabilidade daquele estado ser medido, $\sqrt{}$ sendo assim, o procedimento prepara p_i. Dessa maneira, o estado $|i\rangle$ será medido com probabilidade p_i.

No entanto, [4] não resolve o problema genérico de eficientemente preparar um estado na forma de 24.

Em linhas gerais, o algoritmo ocorre da seguinte maneira. Seja $n = \log N$, o qual N é o número total de uma lista que representa uma distribuição total de probabilidade. Inicialmente, dividimos a distribuição em 2^m regiões com $m = 0$. O algoritmo é executado iterativamente construindo 25.

$$|\psi_{(m)}\rangle = \sum_{i=0}^{2^m-1} \sqrt{p_i^{(m)}}|i\rangle \qquad (25)$$

Note que $p_i^{(m)}$ é a probabilidade de uma variável aleatória qualquer x – ou melhor, a probabilidade de medição – estar na região $|i\rangle$. Seguindo a iteração, adicionamos um novo *qubit* ao estado 25 e alcançamos a evolução de 26.

$$\sqrt{p_i^{(m)}}|i\rangle \rightarrow \sqrt{\alpha_i}|i\rangle|0\rangle + \sqrt{\beta_i}|i\rangle|1\rangle \qquad (26)$$

o qual α é a probabilidade de medição na região $|i,0\rangle$ e β, a probabilidade de medição na região $|i,1\rangle$. Deste jeito, ajustamos as probabilidades de medição das $m + 1$ regiões atuais como mostrado em 27.

$$|\psi_{(m+1)}\rangle = \sum_{i=0}^{2^{m+1}-1} \sqrt{p_i^{(m+1)}}|i\rangle \qquad (27)$$

Esse processo é repetido até que $m = n$, isto é, até que seja obtida a superposição desejada.

IV. Estado da arte do HHL

Desde que o *HHL* foi apresentado, em 2009, melhoras na complexidade foram trabalhadas. Houveram melhoras que reduziram a complexidade para um crescimento linear de κ [13], logarítmico em $\frac{1}{\epsilon}$ e independente da esparsidade s.

Em [13] é definido uma variação do conhecido *Amplitude Amplification* chamado *Variable Time Amplitude Amplification*, de maneira resumida, consiste em repetir um algoritmo t_m vezes. À cada iteração, é atualizado um registrador auxiliar que tem como saída: 0, se o algoritmo parou sem a saída desejada; 1, se deveria ser amplificado; e 2, se a computação ainda não parou. A ideia é aplicar a *Variable Time Amplitude Amplification* no *QPE*, isto é, permitir a estimação dos autovalores até a precisão alcance $O(\epsilon\tilde{\lambda}_i)$. O algoritmo diminui a complexidade do algoritmo de $O(\kappa^2 \log n)$ para $O(\kappa \log^3 \kappa \log n)$.

Em [14] foi apresentado uma melhora na precisão do *HHL* pela evitação da utilização do *QPE*. Em linhas gerais, a proposta baseia-se em uma técnica genérica de implementar qualquer operador a partir de uma representação de uma série de *Fourier*. A proposta diminui a dependência em $poly(\frac{1}{\epsilon})$ para $poly(\log \frac{1}{\epsilon})$.

Em [15], por sua vez, há uma melhora no requisito de esparsidade s da matriz de entrada. O algoritmo original tinha crescimento quadrático com em função de s, com a melhora, o algoritmo se torna independente de s, o que abre caminho para aplicação do *HHL* em matrizes densas. O algoritmo é baseado na execução de uma sub-rotina apresentada de estimação do valor singular que, por sua vez, utiliza o próprio *QPE* como sub-rotina. Ao final, a complexidade total fica $O(\kappa^2 \sqrt{n} \log^* n)$.

A Tabela I mostra a diferença na complexidade dos diversos algoritmos citados comparados com *HHL* original e com o algoritmo clássico do gradiente descendente.

TABELA I

Comparação de complexidades do Gradiente Descendente com versões do *HHL*.

Problema	Algoritmo	Complexidade
LSP	GD	$\mathcal{O}(ns\kappa \log 1/\epsilon)$
QLSP	HHL	$\mathcal{O}(\log n s^2 \kappa^2/\epsilon)$
QLSP	VTAA HHL	$\mathcal{O}(\log n s^2 \kappa/\epsilon)$
QLSP	Childs et al. 2017	$\mathcal{O}(sk \log^*(sk/\epsilon))$
QLSP	QLSA	$\mathcal{O}(k^2 \log^*(n) \|H\|/\epsilon)$

V. Conclusões

Neste trabalho, fazemos uma descrição do algoritmo *HHL* tentando focar em suas limitações. Fornecemos uma intuição do algoritmo e descrevemos as partes. Em seguida, analisamos dificuldades da aplicação do algoritmo de maneira que realmente se extraia a eficiência. Por último, procuramos mostrar o estado da arte com melhorias que ocorreram desde a criação do algoritmo.

Agradecimentos

Agradecimento a coordenação do WECIQ por permitir um ambiente de troca de conhecimento.

Referências

[1] A. W. Harrow, A. Hassidim e S. Lloyd, *Quantum algorithm for linear systems of equations*. Physical review letters, 2009.

[2] S. Aaronson *Quantum Machine Learning Algorithms: Read the Fine Print*, Nature Physics, 2014.

[3] D. W. Berry *et al*. *Efficient quantum algorithms for simulating sparse Hamiltonians*, Communications in Mathematical Physics, 2007.

[4] L. Grover e T. Rudolph. *Creating superpositions that correspond to efficiently integrable probability distributions.* 2002.

[5] S. Aaronson. *Introduction to Quantum Information Science: Lecture Notes* (2018). Communications in Mathematical Physics, 2007.

[6] M. A. Nielsen e I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge university press, 2010.

[7] S. Lloyd. *Universal Quantum Simulators.* Science, 1996.

[8] R. Kothari. *Efficient simulation of Hamiltonians*. MS thesis. University of Waterloo, 2010.

[9] A. M. Childs *et al*. *Exponential algorithmic speedup by a quantum walk.* Proc. 35th ACM Symposium on Foundations of Computer Science, 2003.

[10] D. W. Berry et al. *Exponential improvement in precision for simulating sparse Hamiltonians.* Proceedings of the forty-sixth annual ACM symposium on Theory of computing, 2014.

[11] R. Cole e U. Vishkin. *Deterministic coin tossing with applications to optimal parallel list ranking.* Inform. and Control, 1986.

[12] A. Panconesi e R. Rizzi. *Some simple distributed algorithms for sparse networks.* Distributed computing, 2001.

[13] A. Ambainis. *Variable time amplitude amplification and quantum algorithms for linear algebra problems.* 29th Symposium on Theoretical Aspects of Computer Science, 2012.

[14] A. M. Childs *et al. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision.* SIAM Journal on Computing, 2017.

[15] L. Wossing, Z. Zhao e A. Prakash. *Quantum Linear System for Dense Matrices.* Phis. Rev. Let., 2018.

7º WECIQ

A new Euclidean framework for Quantum-Enhanced Neural Networks.

Francisco Javier Ropero Peláez, Ricardo Tiosso Panassiol, Clovis Caface, Karla Vittori

***Abstract*— This paper proposes a novel mathematical framework based on Euclidean algebra that seamlessly integrates the principles of neural and quantum computing. We introduce a method for calculating synaptic weights through vectorial summations, an approach that is naturally aligned with quantum operations. By drawing parallels between the operational mechanisms of biological synapses and quantum systems, we illustrate how principal component analysis can be effectively implemented in the synaptic weights of a standard neuron so that biology is able to mimic processes akin to quantum computation. This integration paves the way for brain-inspired neural architectures that could potentially outperform current ANNs in both speed and cognitive capabilities.**

Keywords*— *Quantum Computing, Neural Networks, Euclidean Algebra, Synaptic Weights Principal Component Analysis.

I. INTRODUCTION

Artificial neural networks, initially conceptualized in the mid-20th century, have undergone significant transformations through the advent of robust algorithms like AlexNet [1] and Generative Pre-trained Transformers, GPTs [2], and advancements in hardware technologies such as GPUs. These innovations have catalyzed remarkable improvements in model performance, managing complex computations across billions of parameters at unprecedented speeds. However, despite these technological strides, the scalability of neural networks continues to be hindered by substantial latency issues, particularly as server demands escalate due to increasing client interactions.

This bottleneck, primarily caused by the extensive computational requirements of contemporary neural network algorithms, presents a critical challenge: achieving real-time processing speeds. Quantum computing emerges as a promising solution with its potential to perform calculations nearly instantaneously, especially for algebraic operations integral to neural networks like summation and matrix multiplication.

Furthermore, traditional neural network training methodologies, such as error back-propagation, are not only time-intensive, often extending over weeks or months, but also prone to converging to local minima rather than global optima. This method also distributes learned information across numerous synaptic weights, making it difficult to pinpoint specific synapses' contributions to learned behaviors.

In contrast, the human brain exhibits a remarkable capacity for continuous, real-time learning, avoiding the pitfalls of suboptimal local minima and providing a clearer traceability of information across synapses. Despite the relatively slow signal transmission speeds in biological neurons, the brain's efficiency in processing and interpreting complex stimuli remains superior to artificial systems.

Looking ahead, there is a compelling vision where brain-inspired neural networks could gradually replace purely artificial constructs. This transition is supported by the foundational use of vector algebra in both domains, which facilitates operations such as vector projection, normalization, and orthogonalization. The intrinsic alignment between the operational principles of real neuron synapses, whose weight adjustment is governed by Hebbian learning rules equivalent to conditional probabilities [3], and the mathematical underpinnings of quantum computing provides a robust framework for this integration.

Our proposed mathematical foundation aims to unify the principles of Euclidean algebra underpinning both neural and quantum computations, offering a streamlined approach to calculating synaptic weights through simple vectorial summations—ideally suited for quantum processing. Moreover, this framework aligns with principal components analysis a process used by quantum physics to pinpoint the relevant variables associated with a physic phenomenon that we demonstrate is calculated easily under our framework by biological neurons, further illustrating our central nervous system's capability to identify and utilize crucial orthogonal features, a process that quantum systems can parallel.

I. From vectors to probabilistic concepts.

N-tuples of real numbers are an ordered collection of numbers. The n-tuple $[3,1,2]$ may, for example, indicate the number of presynaptic spikes that reach each one of the three synapses of a neuron during a certain period of time. N-tuples are vectors because they have all the properties of linear spaces.

Although many types of bases are possible, in the n-tuple linear space we will use a special type of basis called "universal basis", which is calculated in the following way: having a finite set of m n-tuples $\vec{A}^1, \vec{A}^2, \ldots, \vec{A}^m$ each of the n orthogonal axes $\vec{Y}_i$ of this universal basis is selected so that it has only one component whose value, K_i, is different from zero.

This component is chosen to be the sum of the absolute values of the corresponding components among the set of n-tuples:

$\forall i = 1,2,\ldots,n;\ j = 1,2,\ldots,m,\ \vec{Y}_i = [0,0,\ldots,K_i,\ldots,0]$ such that

$$K_i = \sum_{j=1}^{m} |A_i^j$$

For example, in the case of the set of n-tuples:

$$\vec{W}^1 = [3,1,2];\ \vec{W}^2 = [2,1,1];\ \vec{W}^3 = [3,5,1]$$

the first axis $\vec{Y}_1$ is the one whose first component is $3+2+3=8$ with the other components equal to zero. Then, $\vec{Y}_1 = [8,0,0]$.

Following this criterion, the universal basis is:

Francisco Javier Ropero Peláez, CMCC, UFABC, Santo André-SP, francisco.pelaez@ufabc.edu.br; Ricardo Tiosso Panassiol, NBPIP, USP, ricardo.tiosso@usp.com.br; Clovis Caface, CCENT, UEMASUL, Imperatriz-MA e CMCC, UFABC, Santo André-SP, clovis.caface@uemasul.edu.br; Karla Vittori, CMCC, UFABC, Santo André-SP, karla.vittori@ufabc.edu.br.

$$\vec{Y_1} = [8,0,0];\ \vec{Y_2} = [0,7,0];\ \vec{Y_3} = [0,0,4]$$

Notice that the universal basis is not normalized (we use the l_1-norm as we will explain).

Here we define an specific vector, U, that we call universe or universal vector defined as:

$$\vec{U} = [Y_1, Y_2, Y_3] = [8,7,4] \quad (1)$$

Redefinition of universe, $\vec{U}$, occurs each time a new vector is placed in it, either as a result of an arbitrary decision, like "let us create a certain vector in $\vec{U}$", or as a result of an algorithm involving $\vec{U}$ vectors.

In order to define an Euclidean space, besides selecting an orthogonal basis, we need to define an inner product. According to linear algebra, an inner product is an operation between vectors accomplishing four specific axioms. In the Appendix we show that the following operation between two vectors $\vec{A} = [A_1, A_2, \ldots, A_n]$ and $\vec{B} = [B_1, B_2, \ldots, B_n]$ accomplishes all the axioms that defines any inner product:

$$\vec{A}.\vec{B} = \frac{A_1B_1}{Y_1} + \frac{A_2B_2}{Y_2} + \cdots + \frac{A_nB_n}{Y_n} = \sum_{i=1}^{n} \frac{A_iB_i}{Y_i} \quad (2)$$

Where each of the Y_is corresponds to each universal basis vectors. In terms of a norm we have chosen the, so called, sum norm (or l_1 norm) that finds the magnitude of vectors as follows

$$||\vec{A}|| = \sum_{i=1}^{n} ||A_iY_i|| = \sum |A_i||Y_i|$$

The axioms defining this type of norm are shown in the second part of the Appendix. A generic n-tuple $\vec{W}$ is, according to the classical treatise from Apostol [4], expressed in terms of its basis as:

$$\vec{W} = C_1\vec{Y_1} + C_2\vec{Y_2} + \cdots + C_n\vec{Y_n} = \frac{\vec{W}.\vec{Y_1}}{\vec{Y_1}.\vec{Y_1}}\vec{Y_1} + \frac{\vec{W}.\vec{Y_2}}{\vec{Y_2}.\vec{Y_2}}\vec{Y_2} + \cdots + \frac{\vec{W}.\vec{Y_n}}{\vec{Y_n}.\vec{Y_n}}\vec{Y_n}, \quad (4)$$

The dots represent inner products, being each coefficient C_i the i^{eth} component relative to the basis element $\vec{Y_i}$

$$C_i = \frac{\vec{W}.\vec{Y_i}}{\vec{Y_i}.\vec{Y_i}} \quad (5)$$

For example, a specific n-tuple $\vec{W^1} = [3,1,2]$ can be expressed as:

$$\vec{W^1} = \frac{[3,1,2].[8,0,0]}{[8,0,0].[8,0,0]}\vec{Y_1} + \frac{[3,1,2].[0,7,0]}{[0,7,0].[0,7,0]}\vec{Y_2} + \frac{[3,1,2].[0,0,4]}{[0,0,4].[0,0,4]}\vec{Y_3} = \frac{\frac{3\times8}{8}}{\frac{8\times8}{8}}\vec{Y_1} + \frac{\frac{1\times7}{7}}{\frac{7\times7}{7}}\vec{Y_2} + \frac{\frac{2\times4}{4}}{\frac{4\times4}{4}}\vec{Y_3} = \left[\frac{3}{8}, \frac{1}{7}, \frac{2}{4}\right]$$

According to this, equation 4 can also be written as:

$$\vec{w} = \frac{W_1}{Y_1}\vec{Y_1} + \frac{W_2}{Y_2}\vec{Y_2} + \cdots + \frac{W_n}{Y_n}\vec{Y_n} = \left[\frac{W_1}{Y_1}, \frac{W_2}{Y_2}, \ldots, \frac{W_n}{Y_n}\right];$$

where each component W_i/Y_i of the vector is referred to an axis $\vec{Y_i}$ from the universal basis.

It is possible to give a probabilistic interpretation when a certain event/vector $\vec{W_1}$ is composed of finite separate categories, $\vec{Y_i}$s. For example, $\vec{W_1}$ can be interpreted as the "statistical event" whose probability of belonging to class $\vec{Y_1}$ is 3/8, to class $\vec{Y_2}$ is 1/7, and to class $\vec{Y_3}$ is 2/4. Therefore, vector $\vec{W_1}$ is treated as a probabilistic event, and so are the classes, $\vec{Y_i}$.

To obtain the different types of conditional probabilities the components relative to the basis elements are calculated:

$$C_i = \frac{\vec{W^1}.\vec{Y_i}}{\vec{Y_i}.\vec{Y_i}} = \frac{(W_i^1Y_i)/Y_i}{(Y_iY_i)/Y_i} = \frac{W_i^1}{Y_i} = P(\vec{W^1}/\vec{Y_i}); \quad (6)$$

In the case of the example:

$$P(\vec{W^1}/\vec{Y_1}) = C_1 = \frac{W_1}{Y_1} = \frac{3}{8};\ P(\vec{W^1}/\vec{Y_2}) = C_2 = \frac{W_1}{Y_2} = \frac{1}{7};$$

$$P(\vec{W^1}/\vec{Y_3}) = C_3 = \frac{W_1}{Y_3} = \frac{2}{4}$$

According to this probabilistic interpretation each generic $\vec{W}$ is a combination of its orthogonal axes where its corresponding components C_i can be interpreted as conditional probabilities

$$\vec{W} = \sum_{i=1}^{N} C_i\vec{Y_i} = \sum_{i=1}^{n} P(\vec{W}/\vec{Y_i})\vec{Y_i}$$

For calculating the standard probability of W, the component relative to the universe, U, is calculated as if the universe were the only axis $\vec{W}$, where

$$C_U = \frac{\vec{W}.\vec{U}}{\vec{U}.\vec{U}} = \frac{(WU)/U}{(UU)/U} = \frac{W}{U} = P(\vec{W}/\vec{U}) = P(\vec{W}); \quad (7)$$

So that vector $\vec{W}$ is defined as a fraction of the universe $\vec{U}$. In the case of the example:

$$P(\vec{W^1}) = P(\vec{W^1}/\vec{U}) = C_U = \frac{W^1}{U} = \frac{6}{19}$$

Performing the same operation with a generic $\vec{W}$ but, in this case, having the $\vec{U}$ in terms of the bases vectors $\vec{Y_i}$ we obtain:

$$\begin{gathered} P(\vec{W}) = C_U = \frac{\vec{W}.\vec{U}}{\vec{U}.\vec{U}} = \frac{[W_1,W_2,\ldots,W_n].[Y_1,Y_2,\ldots,Y_n]}{[Y_1,Y_2,\ldots,Y_n].[Y_1,Y_2,\ldots,Y_n]} = \\ \frac{\frac{W_1Y_1}{Y_1} + \frac{W_2Y_2}{Y_2} + \cdots + \frac{W_nY_n}{Y_n}}{\frac{Y_1Y_1}{Y_1} + \frac{Y_2Y_2}{Y_2} + \cdots + \frac{Y_nY_n}{Y_n}} = \frac{W_1+W_2+\cdots+W_n}{Y_1+Y_2+\cdots+Y_n} = \\ \frac{W_1}{Y_1+Y_2+\cdots+Y_n} + \frac{W_2}{Y_1+Y_2+\cdots+Y_n} + \ldots \\ + \frac{W_n}{Y_1+Y_2+\cdots+Y_n} = \\ \frac{W_1}{Y_1}\frac{Y_1}{Y_1+Y_2+\cdots+Y_n} + \frac{W_2}{Y_2}\frac{Y_2}{Y_1+Y_2+\cdots+Y_n} + \ldots \\ + \frac{W_n}{Y_n}\frac{Y_n}{Y_1+Y_2+\cdots+Y_n} = \\ P(\vec{W}/\vec{Y_1})P(\vec{Y_1}/\vec{U}) + \cdots + P(\vec{W}/\vec{Y_i})P(\vec{Y_i}/\vec{U}) + \ldots \\ + P(\vec{W}/\vec{Y_n})P(\vec{Y_n}/\vec{U}) = \\ P(\vec{W}/\vec{Y_1})P(\vec{Y_1}) + \cdots + P(\vec{W}/\vec{Y_i})P(\vec{Y_i}) + \ldots \\ + P(\vec{W}/\vec{Y_n})P(\vec{Y_n}) \end{gathered} \quad (8)$$

That corresponds to the, so called, Law of Total Probability. Another way of reaching to the same result is by calculating the projection of $\vec{W}$ over $\vec{U}$.

As an example, let us calculate $P(\vec{W^1})$:

$$P(\vec{W_1}) = C_U = \frac{\vec{W^1}\vec{U_i}}{\vec{U}\vec{U_i}} = \frac{[3,1,2][8,7,4]}{[8,7,4][8,7,4]} = \frac{3}{8}\frac{8}{19} + \frac{1}{7}\frac{7}{19} + \frac{2}{4}\frac{4}{19}$$

Another specific case of equation 7 is obtaining the probability of the Universe, U.

$$P(\vec{U}) = P(U/U) = \frac{U}{U} = 1 \quad (9)$$

So that we have:

$$C_U = P(\vec{U}) = 1 \text{ and}$$

$$C_U = \frac{\vec{U}.\vec{U}}{\vec{U}.\vec{U}} = \frac{Y_1}{Y_1}\frac{Y_1}{U} + \cdots + \frac{Y_i}{Y_i}\frac{Y_i}{U} + \cdots + \frac{Y_n}{Y_n}\frac{Y_n}{U} =$$

$$P(Y_1/U) + \cdots + P(Y_i/U) + \cdots + P(Y_n/U)$$

and, therefore the sum of the probabilities of the axes in the universal basis is equal to one.

$$P(\vec{Y}_1/U) + \cdots + P(\vec{Y}_i/U) + \cdots + P(\vec{Y}_n/U) = 1;$$

$$P(\vec{Y}_1) + \cdots + P(\vec{Y}_i) + \cdots + P(\vec{Y}_n) = 1$$

Let us have a generic $\vec{W}; \vec{W} = k_1\vec{Y}_1 + k_2\vec{Y}_2 + \cdots + k_n\vec{Y}_n$. Let us see where the probability operator can be used in both terms of the equation as if were a linear transformation. Calculating $P(\vec{W})$, as C_U:

$$P(\vec{W}) = C_U = \frac{(k_1\vec{Y}_1 + k_2\vec{Y}_+ \cdots + k_n\vec{Y}_n).\vec{U}}{\vec{U}.\vec{U}} =$$

$$\frac{(k_1\vec{Y}_1 + k_2\vec{Y}_+ \cdots + k_n\vec{Y}_n).(\vec{Y}_1 + \vec{Y}_2 + \cdots + \vec{Y}_n)}{(\vec{Y}_1 + \vec{Y}_2 + \cdots + \vec{Y}_n).(\vec{Y}_1 + \vec{Y}_2 + \cdots + \vec{Y}_n)} =$$

$$\frac{\frac{k_1Y_1Y_1}{Y_1} + \frac{k_2Y_2Y_2}{Y_2} + \cdots + \frac{k_nY_nY_n}{Y_n}}{\frac{Y_1Y_1}{Y_1} + \frac{Y_2Y_2}{Y_2} + \cdots + \frac{Y_nY_n}{Y_n}} =$$

$$\frac{k_1Y_1 + k_2Y_2 + \cdots + k_nY_n}{Y_1 + Y_2 + \cdots + Y_n} = k_1\frac{Y_1}{U} + k_2\frac{Y_2}{U} + \cdots + k_n\frac{Y_n}{U} =$$

$$k_1P(Y_1) + k_2P(Y_2) + \cdots + k_nP(Y_n)$$

So that

$$P(\vec{W}) = k_1P(\vec{Y}_1) + k_2P(\vec{Y}_2) + \cdots + k_nP(\vec{Y}_n) \quad (10)$$

this last equation might not be valid for higher-than-one values of k_i as for example a case in which $P(\vec{Y}_i) = 1$ and $k_i > 1$ yielding a higher than one $P(\vec{W})$. However, in the case each factor is a probability (a less than one value) this operation is valid.

A. Calculating the net input of an artificial neuron

An example of a previous concepts is calculating the net input, net, to an artificial neuron with individual inputs I_i.

In this case, the C_i, the components of a generic $\vec{W}$ relative to $\vec{I}$ is

$$C_i = \frac{\vec{W}.\vec{I}}{\vec{I}.\vec{I}} = \frac{[W_1,W_2,\ldots,W_n][I_1,I_2,\ldots,I_n]}{[I_1,I_2,\ldots,I_n][I_1,I_2,\ldots,I_n]} =$$

$$\frac{\frac{W_1I_1}{I_1} + \frac{W_2I_2}{I_2} + \cdots + \frac{W_nI_n}{I_n}}{\frac{I_1I_1}{I_1} + \frac{I_2I_2}{I_2} + \cdots + \frac{I_nI_n}{I_n}} =$$

$$\frac{W_1}{I_1}\frac{I_1}{I_1+I_2+\cdots+I_n} + \frac{W_2}{I_2}\frac{I_2}{I_1+I_2+\cdots+I_n} + \cdots$$

$$+ \frac{W_n}{I_n}\frac{I_n}{I_1+I_2+\cdots+I_n} =$$

$$P(W/I_1)\frac{I_1}{||I||} + P(W/I_2)\frac{I_2}{||I||} + \cdots + P(W/I_n)\frac{I_n}{||I||}$$

This result can also be interpreted as the net input, net, of a neuron whose weights are $P(W/I_i)$.

$$\text{net} = \sum_{i=1}^{n} P(W/I_i)\frac{I_i}{||I||} = \sum_{i=1}^{n} P(W/I_i)P(I_i) \quad (11)$$

This equation is biologically plausible [3] and can be interpreted, as the projection of $\vec{W}$ over input pattern $\vec{I}$. Let us recall that in competitive neural networks, each neuron's weight vector is projected over the current input pattern. The neuron with the higher projection is the one that "wins" in the so called, "winner take all" process.

III. Neuron´s weight adjustment through redefinition.

In our neurons, synaptic weights are realistically modeled as conditional probabilities. Consequently, $\vec{W}^1$, $\vec{W}^2$ and $\vec{W}^3$, when expressed in terms of their universal axis, they can represent the weights of three neurons in the second layer of a competitive neural network, with the first layer being where input patterns are positioned.

When a new input pattern is presented to the network, and one of the three competitive neurons fires, it means that the input pattern should be added to the category of the activated neuron. For instance, consider an input pattern $\vec{I} = [5,1,3]$ being input to the network, and neuron 1 with the "raw" weight $\vec{W}^1 = [3,1,2]$ fires. The process of modifying the weights of neuron 1 unfolds as follows. Initially, we update $\vec{W}^1 = [3,1,2]$ by adding the n-tuple $\vec{I} = [5,1,3]$. This way, the new $\vec{W}^1$ is obtained as follows:

$$\vec{W}^1_{new} = \vec{W}^1 + \vec{I} = [3,1,2] + [5,1,3] = [8,2,5]$$

Subsequently, we express this $\vec{W}^1_{new}$ in terms of the universal basis. In this scenario, the universal basis is updated in accordance with its definition by incorporating the contribution of each component of $\vec{I} = [5,1,3]$ to each axis. Therefore, the new universal basis becomes:

$$\vec{Y}_1 = [Y_1,0,0] = [8+5,0,0] = [13,0,0];$$
$$\vec{Y}_2 = [0,Y_2,0] = [0,7+1,0] = [0,8,0];$$
$$\vec{Y}_3 = [0,0,Y_3] = [0,0,4+3] = [0,0,7]$$

Being the updated Universe

$$U_{new} = U + I = [8,7,4] + [5,1,3] = [13,8,7]$$

Therefore , when represented in terms of the Universal basis becomes:

$$\vec{W}^1_{new} = \frac{8}{13}\vec{Y}_1 + \frac{2}{8}\vec{Y}_2 + \frac{8}{7}\vec{Y}_3 = \left[\frac{8}{13}, \frac{2}{8}, \frac{5}{7}\right]$$

While the n-tuple $\vec{W}^2 = [2,1,1]$ remains unchanged, the redefinition of $\vec{W}^1 = [3,1,2]$ leads to a modification in the representation of $\vec{W}^2 = [2,1,1]$ when expressed in terms of the universal basis:

$$\vec{W}^2_{new} = \frac{2}{13}\vec{Y}_1 + \frac{1}{8}\vec{Y}_2 + \frac{1}{7}\vec{Y}_3 = \left[\frac{2}{13}, \frac{1}{8}, \frac{1}{7}\right]$$

Following the redefinition of $\vec{W}^1$, $\vec{W}^2$ transforms into $\vec{W}^2_{new}$, solely due to the modification of the universal basis during the redefinition of $\vec{W}^1$. Consequently, the synapses of O_2, activated by the input pattern $\vec{I}$, decrease their weights, diminishing future firing under similar conditions.

We can see that once the sum of inputs is represented in the universal basis, its components become conditional probabilities like in the pre-synaptic rule that is the most biologically plausible rule among Hebbian rules [3].

Considering this fact, it is straightforward to comprehend the following criterion: "The training process of a neuron gradually aligns synaptic weights with the set of input patterns that activated that neuron." This occurs because the set of weights for a neuron is essentially the summation of all input patterns that triggered that neuron, referenced to the universal basis. Vector $\vec{T}^1$ can also be viewed as the prototype, akin to an

average vector, of the input vectors associated with the activation of neuron O_1.

We can broaden the "redefinition" concept for including not only step-functions (yielding binary outputs) but the output, O, of any other activation function $f(x)$:

$$\vec{W}_{new} = \vec{W} + O\vec{I} = \vec{W} + f(I_{net})\vec{I}$$

where O is usually between 0 and 1. In this scenario, these equations can be utilized by non-active neurons as well. For instance, let's reconsider the previous example, but this time with an output of $O = 0.8$. Starting with the same initial values as in the previous examples, $\vec{W}^1$ is redefined as follows:

$$\vec{W}^1_{new} = \vec{W}^1 + O\vec{I} = [3,1,2] + 0.8[5,1,3] = [7, 1.8, 4.4]$$

When represented in terms of the updated universal basis (which is not influenced by the factor O), it produces:

$$\vec{W}^1_{new} = \frac{7}{13}\vec{Y}_1 + \frac{1.8}{8}\vec{Y}_2 + \frac{4.4}{7}\vec{Y}_3 = \left[\frac{7}{13}, \frac{1.8}{8}, \frac{4.4}{7}\right]$$

Considering the effect of the activation function, we can rewrite the "golden rule" for weight modification stated at previous section as follows: "*The training process of a neuron makes synaptic weights gradually follow the set of input patterns in a proportion indicated by the output of the activation function*".

This simply rule for weight updating can boost neural networks training under both the conventional and quantum paradigms.

IV. Neuron´s weight convergence to principal components

Consider a scenario of a single-layer neural network where all neurons share nearly identical activation functions characterized by the following shape: starting from zero at the origin, they exhibit a moderate increase for average net inputs and experience a substantial, almost exponential increment for higher net inputs. This specific activation pattern aligns with the characteristics observed in thalamic reticular neurons, as depicted in Figure 8 of Mulie et al.'s seminal paper [5].

Let us also suppose that, regarding the inputs, we have a collection of input patterns $\vec{I}^1, \vec{I}^2, \ldots, \vec{I}^p$. These patterns have been adjusted by subtracting their mean, resulting in inputs presented to the network as $\vec{I}^1 - \vec{\theta}, \vec{I}^2 - \vec{\theta}, \ldots, \vec{I}^p - \vec{\theta}$. This mean subtraction process aligns with the behavior observed in thalamocortical neurons, as explained in the comments to Figure 4 in Aguiar \& Peláez paper [6]. Thus, the output of thalamocortical neurons (neurons in the first layer of the thalamus) that feed into reticular neurons (neurons in the second layer of the thalamus) is stripped of its mean.

What would happen to the weights of reticular-like neurons when patterns $\vec{I}$ similar to $\vec{\theta}$ are presented? These patterns would result in an almost zero output for each of the reticular neurons. According to our "golden rule," their weights would tend towards zero. This reduction occurs because these patterns, similar to $\vec{\theta}$ are, due to redefinition, summed in the denominator of our weight expression rather than in the numerators. Therefore, as mentioned earlier, the weights will diminish the contribution of input patterns that resemble $\vec{\theta}$, leading to reduced net inputs for these types of patterns.

Now that we know that only demeaned patterns are fed into our reticular-like neurons, what will be the effect of these patterns in reticular weights?

Only patterns that significantly deviate from their mean can activate reticular neurons, considering the shape of the activation function. These patterns will contribute to the weight increment. This cumulative weight sum is expected to indicate the direction of the first Principal Component since it represents the direction in which patterns exhibit the highest variability concerning their mean. In the thalamus, where only the highest activated neuron fires, weights of subsequently firing reticular neurons evolve to represent subsequent orthogonal principal components.

V. Quantum alternative "bra-ket" notation

Our previous formulation could be used in quantum computing as an alternative notation to the usual "bra-ket" notation [7]. For programmers that do not want to delve into the complexities of the quantum physics notation, our vector formulation in which components are probabilities could be easier to understand. Let us explicit this idea with the classical example of two atoms whose magnetic field, M, is in a state of "quantum superposition". We could, for example, obtain a probability of 0.25 of both being in North orientation, $P(NN) = 0.25$. The other orientations could be as follows: $P(NS) = 0.125; P(SN) = 0.125; P(SS) = 0.5$. Our notation would show this situation in a very simple way:

$$\vec{M} = 0.25\overrightarrow{NN} + 0.125\overrightarrow{NS} + 0.125\overrightarrow{SN} + 0.5\overrightarrow{SS} \quad (12)$$

However, if we were to use the bra-ket notation the same phenomenon would be represented like this.

$$\vec{M} = \sqrt{0.25}NN + \sqrt{0.125}NS + \sqrt{0.125}SN + \sqrt{0.5}SS$$

Our notation would also facilitate the application of quantum logic gates since it is not necessary to square the result to obtain the probabilities at the end, like in this application of the CNOT gate. (In specific cases, little alterations of the matrix would be necessary.)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.25 \\ 0.125 \\ 0.5 \\ 0.125 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.125 \\ 0.125 \\ 0.5 \end{bmatrix}$$

VI. Conclusions

This paper introduces a rigorous mathematical approach integrating quantum computing with neural network methodologies through a Euclidean algebra framework. By adopting vectorial summations for synaptic weight calculations, this method not only aligns closely with the operational principles of quantum mechanics but also enhances the computational efficiency and transparency of neural networks. This foundational work leverages principal component analysis, an analytical tool prevalent in quantum mechanics, to streamline complex computations traditionally executed by artificial neural networks (ANNs).

Our approach extends beyond theoretical development by demonstrating practical implications for neural architecture design, potentially leading to advancements in cognitive capabilities and processing speeds. By elucidating the relationship between biological neural processes and quantum mechanics, we pave the way for innovative brain-inspired computing architectures. These architectures promise to leverage the intrinsic efficiencies of quantum processes,

possibly surpassing existing ANNs in both performance and cognitive depth.

REFERÊNCIAS

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[3] Javier Ropero Peláez and Diego Andina. Do biological synapses perform probabilistic computations? Neurocomputing, 114:24–31, 2013.

[4] Tom M Apostol et al. Calculus ii: multi variable calculus and linear algebra, with applications to differential equations and probability. 1969.

[5] Ch Mulle, Anamaria Madariaga, and M Deschênes. Morphology and electrophysiological properties of reticularis thalami neurons in cat: in vivo study of a thalamic pacemaker. Journal of Neuroscience, 6(8):2134–2145, 1986.

[6] Mariana Antonia Aguiar-Furucho, Francisco Javier Ropero Peláez, et al. Alzheimer's disease as a result of stimulus reduction in a gaba-a-deficient brain: A neurocomputational model. Neural Plasticity, 2020, 2020.

[7] Michael A. Nielsen and Isaac Chuang. Quantum computation and quantum information. American Association of Physics Teachers, 2002.

APÊNDICE

Here we demonstrate that our inner product accomplishes the axioms that defines any inner product for each pair of elements A and B of the linear space of n-tuples, T, and for all choices of A, B, C in T and all real scalars k: 1. Commutativity: $AB = BA$; 2. Distributivity: $A(B + C) = AB + AC$; 3. Associativity: $k(AB) = (kAB)$; and, 4. Positivity: If $A \neq 0 \Rightarrow A.A > 0$.

Proof.Commutativity:

$$AB = \frac{A_1B_1}{Y_1} + \frac{A_2B_2}{Y_2} + \cdots + \frac{A_nB_n}{Y_n} = \frac{B_1A_1}{Y_1} + \frac{B_2A_2}{Y_2} + \cdots + \frac{B_nA_n}{Y_n} = BA$$

Distributivity:

$$A.(B+C) = \frac{A_1(B_1+C_1)}{Y_1} + \frac{A_2(B_2+C_2)}{Y_2} + \cdots + \frac{A_n(B_n+C_n)}{Y_n} =$$
$$\left(\frac{A_1B_1}{Y_1} + \frac{A_2B_2}{Y_2} + \cdots + \frac{A_nB_n}{Y_n}\right) +$$
$$\left(\frac{A_1C_1}{Y_1} + \frac{A_2C_2}{Y_2} + \cdots + \frac{A_nC_n}{Y_n}\right) = A.B + A.C$$

.

Associativity:

$$k(A.B) =$$
$$k\frac{A_1B_1}{Y_1} + k\frac{A_2B_2}{Y_2} + \cdots + k\frac{A_nB_n}{Y_n} = \frac{kA_1B_1}{Y_1} + \frac{KA_2B_2}{Y_2} + \cdots + \frac{kA_nB_n}{Y_n} = (kA.B)$$

Now we demonstrate that the norm $||\vec{A}|| = \sum_{i=1}^{n} ||A_i\vec{Y}_i|| = \sum |A_i||Y_i|$ accomplishes the following axioms that defines any norm for each pair of elements A and B of the linear space of n-tuples, T, and for all choices of A, B, C in T and all real scalars k: 1. $||\vec{A}|| = 0$ if $\vec{A} = 0$; 2. Positivity: $||\vec{A}|| > 0$ if $\vec{A} \neq 0$; 3. Homogeneity: $||k\vec{A}|| = |k|||\vec{A}||$; 4. Triangle inequality: $||\vec{A} + \vec{B} \leq ||\vec{A}|| + ||\vec{B}||$

Proof. Properties 1, 2 and 3 directly follow from the norm definition. To demonstrate 4, notice that $||\vec{A} + \vec{B}|| = \sum_{i=1}^{n} |A_i + B_i||Y_i|$ due to the possibility of A_i and B_i being of different sign, is always less or equal than $\sum_{i=1}^{n}(|A_i| + |B_i|)|Y_i|$ which is equal to $||\vec{A}|| + ||\vec{B}||$ as required.

7º WECIQ

Uma Proposta de QPU Fotônica para Qubits de Estados Coerentes

Antonio Aguiar[1], Orleans C. V. Gomes[2], Gabriel F. Leite[3] e João Batista R. Silva[4]

***Resumo*— Este trabalho apresenta uma proposta de *QPU* fotônica versátil para qubits de estados coerentes usando dispositivos baseados em óptica linear capaz de implementar, probabilisticamente, as funções lógicas (*AND, OR, C-NOT, C²-NOT* e *C-SWAP*) com uma eficiência de até 1/4.**

Palavras-Chave*— *QPU fotônica, porta Toffoli, porta Fredkin, óptica linear, estados coerentes.

***Abstract*— This work presents a versatile photonic *QPU* proposal for coherent state qubits using linear optics-based devices able to implement logical functions (*AND*, *OR*, *C-NOT*, *C²-NOT*, and *C-SWAP*) probabilistically with an efficiency of up to 1/4.**

Keywords*— *Photonic QPU, Toffoli gate, Fredkin gate, linear optics, coherent states.

I. Introdução

Em computação quântica, a busca por *QPU*'s (Unidades de Processamento Quântico) eficientes e escaláveis se intensifica. Nesse contexto, a fotônica emerge como uma plataforma promissora, com diversas propostas na literatura explorando características importantes dos fótons como sua capacidade de transportar informação quântica de baixo ruído por serem menos suscetíveis a decoerência [1]–[6]. Assim, eliminando a necessidade de temperaturas baixíssimas ou ambientes com alto vácuo. No entanto, a construção de uma *QPU* fotônica completa ainda representa um desafio significativo. Diversas propostas foram exploradas na literatura, cada uma com suas vantagens e desvantagens [7]–[11].

Os avanços em dispositivos de óptica integrada gerou uma ampla variedade de plataformas voltadas para aplicações em processamento quântico de informação (*QIP*), incluindo sílica sobre isolante [12]–[17], dentre muitos outros. A evolução do Q*IP* tem sido favorável para construção de *QPU*'s que contornem obstáculos relacionados ao princípio da coerência quântica. Portanto, trabalhos teóricos mostram que o ruído e a decoerência não são obstáculos fundamentais aos estudos com a utilização de estados coerentes no processamento de informação quântica [18]–[20]. Usando óptica linear, com o objetivo de codificar informações, os fótons únicos são substituídos por estados coerentes, abrindo a possibilidade de se obter portas lógicas com maiores probabilidades de sucesso.

Considerando que uma porta lógica reversível parte do princípio de que a informação pode ser reconstruída exclusivamente a partir de sua saída, dada sua entrada, e vice-versa, sem qualquer perda de informação. A reversibilidade é uma característica fundamental no desenvolvimento de algoritmos e circuitos quânticos que realizem operações complexas. Diante disso, o objetivo deste trabalho é apresentar um estudo teórico de um dispositivo *QPU* versátil para implementação de funções lógicas (*AND*, *OR*, *C*-NOT, $\bar{C}$-NOT, C^2-NOT e *C*-SWAP) para qubits de estados coerentes, usando um sistema óptico proposto que irá implementá-las probabilisticamente.

DETI[1,2,4] e DEE[3], Universidade Federal do Ceará, Fortaleza - CE; e-mail: antonioaguiar.etiufc@alu.ufc.br[1], orleanscardoso@outlook.com[2], fonsecag.leite1810@gmail.com[3] e joaobrs@ufc.br[4].

II. Qubits Fotônicos e Dispositivos Baseados em Óptica Linear

Uma forma de representar qubits fotônicos é por meio de estados coerentes da luz. Uma das características importantes dos estados coerentes é o fato de serem auto-estados do operador aniquilação $\hat{a}$ (criação, $\hat{a}^\dagger$), com autovalor complexo α ($\hat{a}|\alpha\rangle = \alpha|\alpha\rangle$). Tais estados foram descritos por R. J. Glauber em 1963 [21] e podem ser escritos na base dos estados de Fock, $|n\rangle$, também conhecidos como estados de número de fótons. Logo, o estado coerente $|\alpha\rangle$ é dado por:

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle. \quad (1)$$

Na computação quântica os estados que representam os qubits lógicos precisam obedecer a uma relação fundamental de ortogonalidade entre si. Tal princípio garante a distinguibilidade e precisão das informações armazenadas usando qubits, proporcionando maior confiabilidade e precisão dos sistemas quânticos. Assim, a distinguibilidade entre dois dos estados coerentes, $|\alpha\rangle$ e $|\beta\rangle$, é dado por:

$$|\langle\alpha|\beta\rangle|^2 = e^{-|\alpha-\beta|^2}. \quad (2)$$

Ou seja, quanto mais $|\langle\alpha|\beta\rangle|^2$ tente a zero, mais distinguíveis são os estados. Portanto, em *QIP* com estados coerentes, usa-se $|0\rangle = |-\alpha\rangle$ e $|1\rangle = |\alpha\rangle$, tal que α é um número real e $|\alpha|^2 \geq 4$ [22].

Então, quando dois estados coerentes $|\alpha\rangle$ e $|\beta\rangle$ atravessam um divisor de feixe (*BS*) balanceado, como pode ser visto na Figura 1a, seu estado na saída é dado por [21]:

$$|\alpha,\beta\rangle_{12} \xrightarrow{BS} \left|\frac{\alpha-\beta}{\sqrt{2}}, \frac{\alpha+\beta}{\sqrt{2}}\right\rangle_{1'2'}. \quad (3)$$

Quanto a interação do sinal óptico com um deslocador de fase *PS*(ϕ), é adicionada uma fase ϕ quando ele o atravessa, conforme mostrado na Figura 1b. Assim:

$$|\alpha\rangle_1 \xrightarrow{PS} |e^{j\phi}\alpha\rangle_{1'}. \quad (4)$$

Com isso, um *PS* com $\phi = \pi$ funcionará como uma porta NOT (*X*) para qubits de estados coerentes.

(a) (b)

Fig. 1: Dispositivo ópticos: (a) divisor de feixes balanceado (*BS*) e (b) deslocador de fase (*PS*).

III. UMA PROPOSTA DE QPU FOTÔNICA

Uma proposta *QPU* fotônica (*QPU X*) é apresentado na Figura 2. Esse sistema óptico é composto por onze *BS*'s onde ocorrem as interferências entre os feixes de luz, um *PS* e dois fotodetectores (D_1 e D_2) que permitirão, conforme as medições realizadas, deduzir se as operações lógicas entre os estados de entradas (0, 1 e 4) foram obtidas nas saídas (0', 1', 2' e 5').

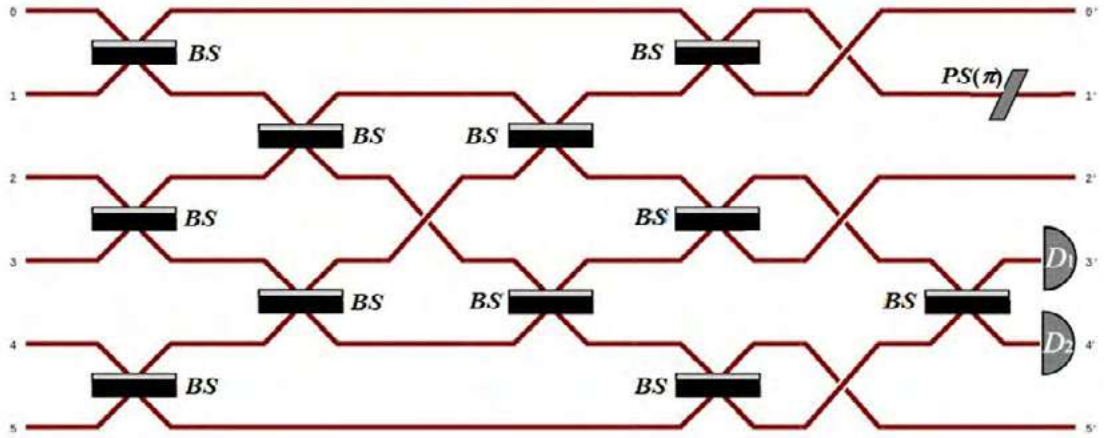


Fig. 2: QPU X fotônica: sistema óptico proposto usando apenas dispositivos ópticos lineares.

Na Figura 2, as entradas enumeradas por 0, 1 e 4 são, respectivamente, as entradas de informação representadas pelos qubits de estados coerentes $|A\rangle_0 = N_A(a_0|-\alpha\rangle + a_1|\alpha\rangle)$, $|B\rangle_1 = N_B(b_0|-\alpha\rangle + b_1|\alpha\rangle)$, $|C\rangle_4 = N_C(c_0|-\alpha\rangle + c_1|\alpha\rangle)$, e as demais entradas (2, 3 e 5) são qubits auxiliares do tipo $|\varphi\rangle = N(|-\alpha\rangle + |\alpha\rangle)$, onde $N's$ são as constantes de normalização. Assim, o estado de entrada, $|\psi_{in}\rangle = |A\rangle_0|B\rangle_1|\varphi\rangle_2|\varphi\rangle_3|C\rangle_4|\varphi\rangle_5$, é dado por:

$$\begin{aligned}|\psi_{in}\rangle = N_A N_B N_C \quad & (a_0b_0c_0|-\alpha,-\alpha,-\alpha\rangle + \\ & a_0b_0c_1|-\alpha,-\alpha,\alpha\rangle + \\ & a_0b_1c_0|-\alpha,\alpha,-\alpha\rangle + \\ & a_0b_1c_1|-\alpha,\alpha,\alpha\rangle + \\ & a_1b_0c_0|\alpha,-\alpha,-\alpha\rangle + \\ & a_1b_0c_1|\alpha,-\alpha,\alpha\rangle + \\ & a_1b_1c_0|\alpha,\alpha,-\alpha\rangle + \\ & a_1b_1c_1|\alpha,\alpha,\alpha\rangle)_{014}|\varphi,\varphi,\varphi\rangle_{235}.\end{aligned} \tag{5}$$

Após esse estado (5) evoluir pelo sistema óptico, obtém-se o seguinte estado (não normalizado) na saída (antes das medições), estado $|\psi_{out}\rangle$:

$$\begin{aligned}|\psi_{out}\rangle \approx \quad & \frac{1}{2\sqrt{2}}|\psi\rangle_{0',1',2',5'}|0,\pm\sqrt{2}\alpha\rangle_{3',4'} \\ & +\frac{1}{2}\sqrt{\frac{7}{2}}|\psi_u\rangle_{0',1',2',3',4',5'},\end{aligned} \tag{6}$$

onde

$$\begin{aligned}|\psi\rangle \approx\ & a_0b_0c_0|-\alpha,-\alpha,-\alpha,\alpha\rangle + a_0b_0c_1|-\alpha,-\alpha,-\alpha,-\alpha\rangle \\ & +a_0b_1c_0|-\alpha,\alpha,\alpha,-\alpha\rangle + a_0b_1c_1|-\alpha,\alpha,-\alpha,\alpha\rangle \\ & +a_1b_0c_0|\alpha,-\alpha,\alpha,-\alpha\rangle + a_1b_0c_1|\alpha,-\alpha,-\alpha,\alpha\rangle \\ & +a_1b_1c_0|\alpha,\alpha,\alpha,\alpha\rangle + a_1b_1c_1|\alpha,\alpha,\alpha,-\alpha\rangle\end{aligned} \tag{7}$$

é o estado desejado na saída, obtido após a medição quando houver detecção apenas no detector D_2. Caso contrário, o sistema falha e o estado na saída será $|\psi_u\rangle_{0',1',2',5'}$. Logo, nota-se (6) que a probabilidade de obter o estado (7) é de, aproximadamente, até 1/8.

IV. APLICAÇÕES

Uma representação da *QPU X* funcional é mostrado na Figura 3.

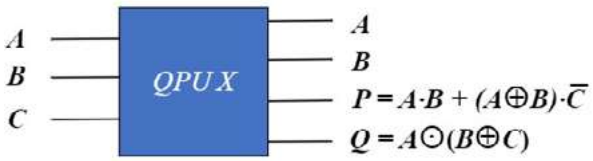


Fig. 3: Uma representação da *QPU X*.

A partir da Figura 3 obtém-se duas portas 3x3, conforme mostrado na Figura 4: a porta *X'* e a porta *X''*. A porta *X'* (Figura 4a), onde a saída *Q* é medida e é diferente de zero, não é reversível, mas quando a entrada *C* for 1 (0), tem-se a função *AND* (*OR*) de *A* e *B* na saída *P*. Por outro lado, a porta *X''* (Figura 4b), quando a saída *P* é medida e não é zero, é reversível e pode obter as funções $\overline{C}$-NOT (*NXOR*) e *C*-NOT (*XOR*) de *A* e *B* quando *C* = 0 e *C* = 1, respectivamente, na saída *Q*.

As funções lógicas obtidas a partir da *QPU X* com predefinição de uma das entradas são mostradas na Tabela I. A

probabilidade de sucesso para obtenção das funções listadas na Tabela I é de até 1/4, uma vez que a entrada 3, na Figura 2, pode ser atribuída o mesmo valor da entrada *C*.

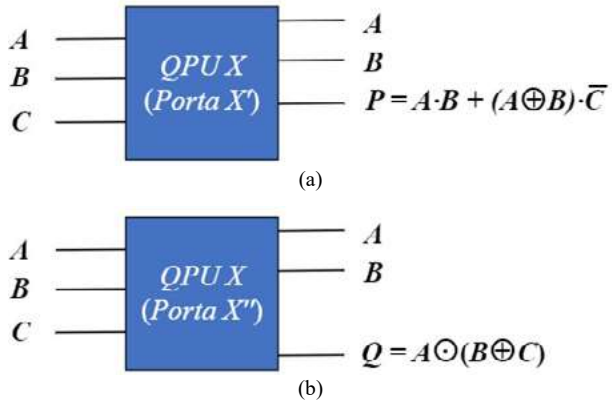


Fig. 4: Representação da (a) porta *X'* e da (b) porta *X''* obtidas a partir da *QPU X*.

TABELA I. *FUNÇÕES LÓGICAS A PARTIR DA QPU X.*

QPU X	***P***	***Q***
$C = 0$	$A + B$	$A \odot B$
$C = 1$	$A \cdot B$	$A \oplus B$

Pode-se obter a porta Toffoli (C^2-NOT) a partir das portas *X'* e *X''* conforme mostrado na Figura 5. Já na Figura 6, é apresentado uma proposta da porta Fredkin (*C*-SWAP) a partir de uma porta C^2-NOT (Figura 5) e de duas *C*-NOT (Figura 4b com $C = 1$). Uma vez que a eficiência das portas quânticas fotônicas restringe a taxa de sucesso da porta Fredkin a 10^{-5} [23]-[24], as portas Toffoli e Fredkin propostas tem uma eficiência significativa de até 1/16 e 1/256, respectivamente.

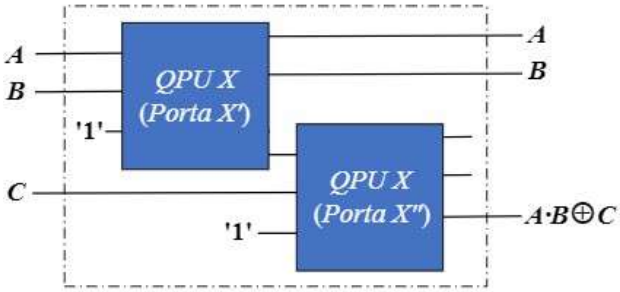


Fig. 5: Representação da porta Toffoli a partir da *QPU X*.

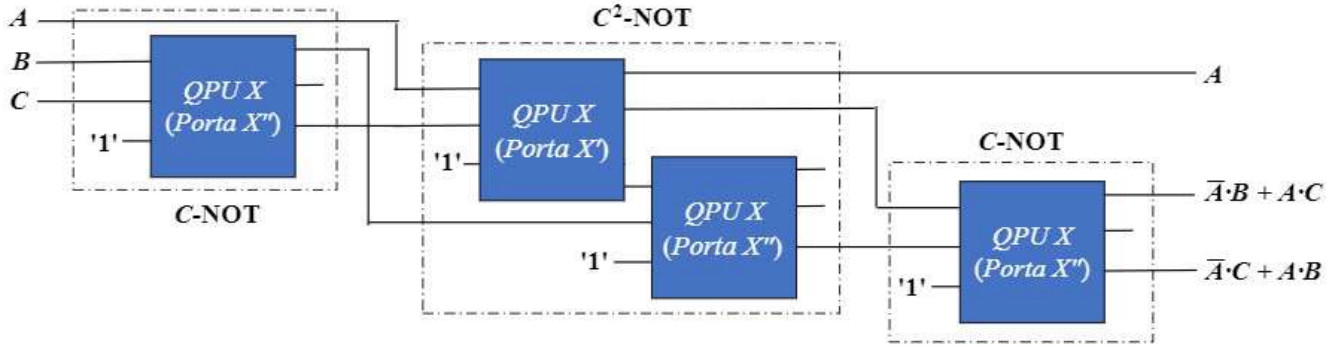


Fig. 6: Representação da porta Fredkin a partir da *QPU X*.

V. CONCLUSÕES

A proposta de uma *QPU* fotônica (*QPU X*) para qubits de estados coerentes apresentada neste artigo demonstra a viabilidade e eficácia da utilização de estados coerentes na implementação de funções lógicas em computação quântica. Ao explorar a óptica linear e a substituição de fótons únicos por estados coerentes, foi possível gerar funções lógicas importantes, como *AND*, *OR*, *C*-NOT, C^2-NOT e *C*-SWAP, de forma probabilística. A reversibilidade dessas funções é essencial para o desenvolvimento de algoritmos e circuitos quânticos complexos, garantindo a reconstrução da informação sem perdas.

A eficiência das funções lógicas geradas a partir da *QPU* proposta é um ponto relevante, destacando a capacidade de implementar operações complexas com estados coerentes. A porta Toffoli (C^2-NOT) e a porta Fredkin (*C*-SWAP) foram obtidas a partir das portas *X'* e *X''* propostas, demonstrando a versatilidade e potencial do sistema proposto. Apesar da eficiência do sistema diminuir à medida que se pretende obter circuitos mais complexos, o que já é esperado para esse tipo de tecnologia, o desempenho da *QPU X* proposta é promissor.

Vale ressaltar a originalidade da *QPU X* por implementar, simultaneamente, duas funções lógicas por processamento com a preservação dos estados de entradas. Portanto, a combinação de óptica linear e estados coerentes mostra-se promissora para superar desafios e alcançar resultados confiáveis no processamento de informação quântica.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, da Funcap e do INCT-IQ.

REFERÊNCIAS

[1] E. Knill, R. Laflamme, e G. J. Milburn, "A scheme for efficient quantum computation with linear optics", *nature*, p. 46–52, 2001.

[2] A. Laing *et al.*, "High-fidelity operation of quantum photonic circuits", *Appl. Phys. Lett.*, vol. 97, nº 21, 2010.

[3] R. Raussendorf e H. J. Briegel, "A one-way quantum computer", *Phys. Rev. Lett.*, vol. 86, nº 22, p. 5188, 2001.

[4] M. A. Nielsen, "Optical quantum computation using cluster states", *Phys. Rev. Lett.*, vol. 93, nº 4, p. 040503, 2004.

[5] J. Wang, F. Sciarrino, A. Laing, e M. G. Thompson, "Integrated photonic quantum technologies", *Nat. Photonics*, vol. 14, nº 5, p. 273–284, 2020.

[6] S. Slussarenko e G. J. Pryde, “Photonic quantum information processing: A concise review”, *Appl. Phys. Rev.*, vol. 6, nº 4, 2019.
[7] P. J. Shadbolt *et al.*, “Generating, manipulating and measuring entanglement and mixture with a reconfigurable photonic circuit”, *Nat. Photonics*, vol. 6, nº 1, p. 45–49, 2012.
[8] J. Huang, Y. Chi, Z. Zhang, Y. Yang, Q. Gong, e J. Wang, “A programmable photonic chip for high-dimensional quantum computing”, em *TENCON 2022-2022 IEEE Region 10 Conference (TENCON)*, IEEE, 2022, p. 1–4.
[9] F. Zilk, K. Staudacher, T. Guggemos, K. Fürlinger, D. Kranzlmüller, e P. Walther, “A compiler for universal photonic quantum computers”, em *2022 IEEE/ACM Third International Workshop on Quantum Computing Software (QCS)*, IEEE, 2022, p. 57–67.
[10] A. Peruzzo *et al.*, “A variational eigenvalue solver on a photonic quantum processor”, *Nat. Commun.*, vol. 5, nº 1, p. 4213, 2014.
[11] Y. Chi *et al.*, “A programmable qudit-based quantum processor”, *Nat. Commun.*, vol. 13, nº 1, p. 1166, 2022.
[12] B. Arrazola *et al.*, “Quantum circuits with many photons on a programmable nanophotonic chip”, *Nat. Publ. Group UK Lond.*, vol. 591, p. 54–60, 2021, doi: https://doi.org/10.1038/s41586-021-03202-1.
[13] N. Maring *et al.*, “A versatile single-photon-based quantum computing platform”, *Nat. Photonics*, vol. 18, nº 6, p. 603–609, 2024.
[14] A. Politi, M. J. Cryan, J. G. Rarity, S. Yu, e J. L. O’brien, “Silica-on-silicon waveguide quantum circuits”, *Science*, vol. 320, nº 5876, p. 646–649, 2008.
[15] B. J. Smith, D. Kundys, N. Thomas-Peter, P. Smith, e I. Walmsley, “Phase-controlled integrated photonic quantum circuits”, *Opt. Express*, vol. 17, nº 16, p. 13516–13525, 2009.
[16] H. Takesue *et al.*, “Entanglement generation using silicon wire waveguide”, *Appl. Phys. Lett.*, vol. 91, nº 20, 2007.
[17] D. Bonneau *et al.*, “Quantum interference and manipulation of entanglement in silicon wire waveguide quantum circuits”, *New J. Phys.*, vol. 14, nº 4, p. 045003, 2012.
[18] T. C. Ralph, A. Gilchrist, G. J. Milburn, W. J. Munro, e S. Glancy, “Quantum computation with optical coherent states”, *Phys. Rev. A*, vol. 68, nº 4, p. 042319, 2003.
[19] H. Jeong e M. S. Kim, “Efficient quantum computation using coherent states”, *Phys. Rev. A*, vol. 65, nº 4, p. 042305, 2002.
[20] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, e G. J. Milburn, “Linear optical quantum computing with photonic qubits”, *Rev. Mod. Phys.*, vol. 79, nº 1, p. 135, 2007.
[21] R. J. Glauber, “The quantum theory of optical coherence”, *Phys. Rev.*, vol. 130, nº 6, p. 2529, 1963.
[22] J. B. R. Silva e R. V. Ramos, “Smart generation of a tripartite GHZ-type state for coherent state qubit”, *Opt. Commun.*, vol. 281, nº 9, p. 2705–2709, 2008.
[23] H. F. Hofmann e S. Takeuchi, “Quantum phase gate for photonic qubits using only beam splitters and postselection”, *Phys Rev A*, vol. 66, nº 2, p. 024308, ago. 2002.
[24] T. Ono, R. Okamoto, M. Tanida, H. F. Hofmann, e S. Takeuchi, “Implementation of a quantum controlled-SWAP gate with photonic circuits”, *Sci. Rep.*, vol. 7, nº 1, p. 45353, 2017.

7º WECIQ

Propostas de Portas Reversíveis para Obtenção de Funções Lógicas e *C*-NOT para Qubits de Estados Coerentes.

Orleans C. V. Gomes[1], Gabriel F. Leite[2], Antônio F. Aguiar[3], Kleber Z. Nóbrega[4] e João Batista R. Silva[5]

***Resumo* — Este trabalho propõe duas portas reversíveis inéditas de três qubits e um sistema óptico para implementá-las probabilisticamente. Usando dispositivos baseados em óptica linear, o sistema proposto para qubits de estados coerentes é capaz de implementar funções lógicas (AND, OR, NAND e/ou NOR) para dois qubits, simultaneamente, e as portas *C*-NOT e $\bar{C}$-NOT com uma eficiência de até 1/8.**

Palavras-Chave— Portas reversíveis, óptica linear, estados coerentes, C-NOT, funções lógicas.

***Abstract*— This work proposes two gates reversible three-qubit gates and an optical system to probabilistically implement them. Using linear optics-based devices, the proposed system for coherent state qubits is able to implement logical functions (AND, OR, NAND, and/or NOR) for two qubits simultaneously, as well as the *C*-NOT and $\bar{C}$-NOT gates, with an efficiency of up to 1/8.**

Keywords — Reversible gates, linear optics, coherent states, C-NOT, logical functions.

I. Introdução

A computação quântica é uma abordagem inovadora para processamento de informações, baseada na mecânica quântica [1]-[2]. Enquanto a computação clássica armazena informações em bits, que podem estar em um estado de 0 ou 1, a computação quântica utiliza qubits, que podem estar em uma superposição de ambos os estados simultaneamente, permitindo realizar cálculos muito mais rapidamente do que seria possível com computadores clássicos.

Apesar dos desafios para implementar portas quânticas que são essenciais para processamento quântico de informação, atualmente, existem várias tecnologias sendo utilizadas para realização de computação quântica, incluindo pontos quânticos [3]-[5]. supercondutores [6]-[8], ressonância magnética nuclear (RMN) [9]-[11], íons aprisionados [12]-[14], spins eletrônicos em diamantes, entre outros. No entanto, uma das tecnologias mais promissoras é a computação quântica baseada em óptica linear e dispositivos fotônicos [15]-[25].

Portanto, este trabalho propõe um estudo teórico de duas portas reversíveis inéditas de três qubits e um sistema óptico para implementá-las probabilisticamente. O sistema, baseado em dispositivos ópticos lineares para qubits de estados coerentes, oferece flexibilidade para obtenção: de funções lógicas (AND, OR, NAND e/ou NOR) para dois qubits, simultaneamente; e das portas *C*-NOT e $\bar{C}$-NOT.

DETI[1,3,4,5] e DEE[2], Universidade Federal do Ceará[1–5], Fortaleza - CE; e-mail: orleanscardoso@outlook.com[1], fonsecag.leite1810@gmail.com[2], antonioaguiar.etiufc@alu.ufc.br[3], kznobrega@ufc.br[4] e joaobrs@ufc.br[5].

II. As Portas Reversíveis Propostas

No presente trabalho, são propostas duas portas reversíveis para três qubits, conforme ilustrado na Figura 1. As entradas lógicas para o sistema são representadas por *A, B e C*, enquanto as saídas correspondentes, determinadas pelas entradas, são indicadas por S_1, S_2 e S_3.

A porta C_1, Figura 1a, difere da porta C_2, Figura 1b, apenas na saída S_3. E as matrizes das mesmas são apresentadas, respectivamente, (1) e (2).

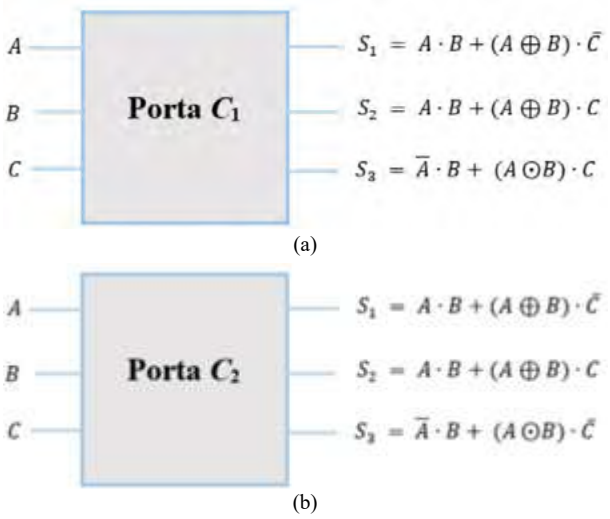
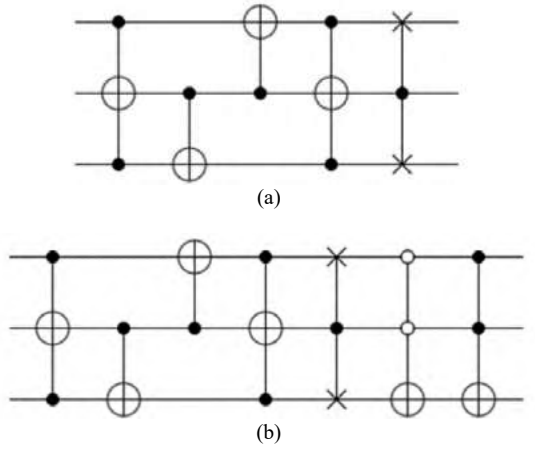


Fig. 1. Portas reversíveis propostas: (a) porta C_1 e (b) porta C_2.

$$C_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$C_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (2)$$

A Figura 2 apresenta os circuitos quânticos não-otimizados para implementar as portas C_1 e C_2. A construção desses circuitos usa-se apenas portas *C*-NOT, Toffoli e Fredkin.



(a)

(b)

Fig. 2. Circuitos equivalentes usando apenas portas *C*-NOT, Toffoli e Fredklin: (a) porta C_1 e (b) porta C_2.

III. Sistema Óptico Proposto

A Figura 3 apresenta um sistema óptico inédito que é capaz de implementar as portas reversíveis C_1 e C_2, probabilisticamente, usando óptica linear. O sistema óptico proposto opera através da manipulação de feixes de luz para implementar as operações lógicas mostradas na Figura 1.

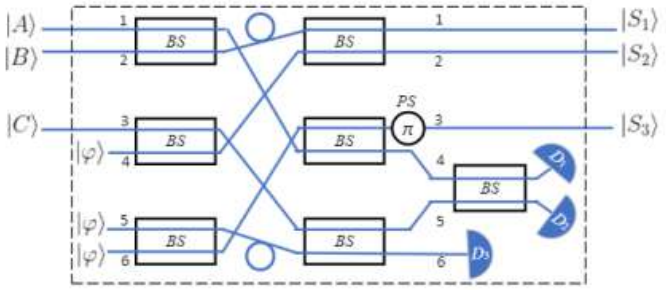


Fig. 3. Circuitos óptico que implementa as portas C_1 e C_2 usando apenas dispositivos ópticos lineares.

Essa manipulação é realizada por meio de componentes ópticos como divisores de feixes balanceados *BS* que direcionam e geram interferência entre feixes de luz e modificam a fase dos mesmos por meio de um deslocador de fase *PS* de acordo com os princípios da óptica. Por fim, medições são realizadas com fotodetectores (*D's*) para determinar se as operações lógicas desejadas foram obtidas.

O sistema óptico na Figura 3 utiliza-se de três estados auxiliares $|\varphi\rangle$ além das três entradas ($|A\rangle$, $|B\rangle$ e $|C\rangle$). Tais estados, tanto os de entrada quanto os auxiliares, são construídos a partir de estados coerentes da luz.

Os estados coerentes são auto-estados do operador de aniquilação $\hat{a}$ (criação, $\hat{a}^\dagger$), com autovalor complexo α ($\hat{a}|\alpha\rangle = \alpha|\alpha\rangle$), e foram introduzidos por R. J. Glauber em 1963 [25]. Estes estados podem ser escritos na base dos estados de Fock (estados de número de fótons) como:

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle. \tag{3}$$

Na área da informação quântica, a ortogonalidade entre os estados que representam os qubits lógicos é fundamental. Essa propriedade garante a distinção precisa das informações armazenadas nos qubits, assegurando a confiabilidade e a precisão dos sistemas quânticos. Portanto, para estados coerentes $|\alpha\rangle$ e $|\beta\rangle$, o produto interno é dado por:

$$|\langle\alpha|\beta\rangle|^2 = e^{-|\alpha-\beta|^2}. \tag{4}$$

Assim, os qubits lógicos podem ser codificados usando $|0\rangle=|-\alpha\rangle$ e $|1\rangle=|\alpha\rangle$, sendo α um número real e a distinguibilidade entre os mesmos é garantida para uma $|\alpha|^2 \geq 4$ [20].

Portanto, dois estados coerentes $|\alpha\rangle$ e $|\beta\rangle$ passam pelo *BS* conforme mostrado na Figura 3, seu estado na saída será [20]:

$$|\alpha, \beta\rangle \xrightarrow{BS} \left|\frac{\alpha-\beta}{\sqrt{2}}, \frac{\alpha+\beta}{\sqrt{2}}\right\rangle. \tag{5}$$

Já o $PS(\theta)$ adiciona uma fase θ ao sinal óptico que o atravessa [20]. Ou seja:

$$|\alpha\rangle \xrightarrow{PS} |e^{j\theta}\alpha\rangle. \tag{6}$$

Logo, o $PS(\theta)$ com $\theta = \pi$ funcionará como a porta NOT (X) para qubits de estados coerentes.

Então, conforme mostrado na Figura 3, os estados de entrada são dados por $|A\rangle = N_A(a_0|-\alpha\rangle + a_1|\alpha\rangle)$, $|B\rangle = N_B(b_0|-\alpha\rangle + b_1|\alpha\rangle)$, $|C\rangle = N_C(c_0|-\alpha\rangle + c_1|\alpha\rangle)$, e $|\varphi\rangle = N(|-\alpha\rangle + |\alpha\rangle)$ com estado auxiliar, onde *N's* são as constantes de normalização. Com isso, o estado de entrada, $|\psi_{in}\rangle = |A\rangle_1 |B\rangle_2 |C\rangle_3 |\varphi\rangle_4 |\varphi\rangle_5 |\varphi\rangle_6$, é dado por:

$$\begin{aligned}|\psi_{in}\rangle = N_A N_B N_C (&a_0 b_0 c_0|-\alpha,-\alpha,-\alpha\rangle + \\ &a_0 b_0 c_1|-\alpha,-\alpha,\alpha\rangle + \\ &a_0 b_1 c_0|-\alpha,\alpha,-\alpha\rangle + \\ &a_0 b_1 c_1|-\alpha,\alpha,\alpha\rangle + \\ &a_1 b_0 c_0|\alpha,-\alpha,-\alpha\rangle + \\ &a_1 b_0 c_1|\alpha,-\alpha,\alpha\rangle + \\ &a_1 b_1 c_0|\alpha,\alpha,-\alpha\rangle + \\ &a_1 b_1 c_1|\alpha,\alpha,\alpha\rangle)_{123}|\varphi,\varphi,\varphi\rangle_{456}.\end{aligned} \tag{7}$$

Após esse estado evoluir pelo sistema óptico, obtém-se o seguinte estado na saída (antes das medições), estado $|\psi_{out}\rangle$, onde os três primeiros qubits (1, 2 e 3) correspondem às saídas S_1, S_3 e S_3 da porta C_1 ou C_2, conforme os resultados obtidos na medição dos três últimos qubits (4, 5 e 6):

$$\begin{aligned}|\psi_{\text{out}}\rangle \approx &\frac{1}{2\sqrt{2}} |\psi_1\rangle_{123} \left|0, \pm\sqrt{2}\alpha, \pm\alpha\right\rangle_{456} \\ &+ \frac{1}{2\sqrt{2}} |\psi_2\rangle \left|\pm\sqrt{2}\alpha, 0, \pm\alpha\right\rangle \\ &+ \frac{\sqrt{3}}{2} |\psi_{\text{u}}\rangle_{1-6},\end{aligned} \tag{8}$$

onde:

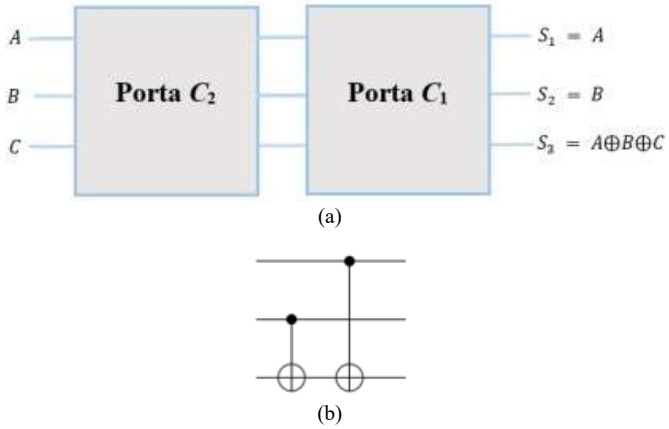
$$\begin{aligned}|\psi_1\rangle \approx &\, a_0 b_0 c_0|-\alpha,-\alpha,-\alpha\rangle + a_0 b_0 c_1|-\alpha,-\alpha,\alpha\rangle + \\ &a_0 b_1 c_0|\alpha,-\alpha,\alpha\rangle + a_0 b_1 c_1|-\alpha,\alpha,\alpha\rangle + \\ &a_1 b_0 c_0|\alpha,-\alpha,-\alpha\rangle + a_1 b_0 c_1|-\alpha,\alpha,-\alpha\rangle + \\ &a_1 b_1 c_0|\alpha,\alpha,-\alpha\rangle + a_1 b_1 c_1|\alpha,\alpha,\alpha\rangle\end{aligned} \tag{9}$$

e

$$|\psi_2\rangle \approx a_0b_0c_0|-\alpha,-\alpha,\alpha\rangle + a_0b_0c_1|-\alpha,-\alpha,-\alpha\rangle + a_0b_1c_0|\alpha,-\alpha,\alpha\rangle + a_0b_1c_1|-\alpha,\alpha,\alpha\rangle + a_1b_0c_0|\alpha,-\alpha,-\alpha\rangle + a_1b_0c_1|-\alpha,\alpha,-\alpha\rangle + a_1b_1c_0|\alpha,\alpha,\alpha\rangle + a_1b_1c_1|\alpha,\alpha,-\alpha\rangle \quad (10)$$

são os estados desejados na saída das portas C_1 e C_2, respectivamente, obtidos após a medição quando o sistema funciona e o estado $|\psi_u\rangle_{123}$ corresponde os casos que o sistema falha. Assim, em (8), quando houver detecção nos detectores D_2 e D_3 e nenhuma detecção em D_1, o estado na saída é (9) e quando houver apenas detecção em D_1 e D_3 e nenhuma detecção em D_2, o estado da saída será (10). Logo, nota-se (8) que a probabilidade de obter tanto o estado $|\psi_1\rangle$ quanto o $|\psi_2\rangle$ é de até 1/8.

Pode-se cascatear a porta C_1 seguida de uma porta, conforme mostrado na Figura 4a, para obter um circuito quântico mais complexo (Circuito *CC*) constituído de duas portas *C*-NOTs, Figura 4b.



(a)

(b)

Fig. 4. Circuitos quântico obtido pelo (a) cascateamento das portas C_1 *e* C_2 e que corresponde a (b) .duas portas *C*-NOT's.

IV. Aplicações

O sistema óptico proposto possui a capacidade de ser configurado, permitindo a obtenção de diversas funções lógicas a partir de uma única configuração inicial. Essa flexibilidade é alcançada através da predefinição de uma das entradas do sistema, enquanto as demais entradas assumem diferentes valores para gerar as funções lógicas desejadas conforme mostrado na Tabela I.

Pode-se notar na Figura 4a que se a entrada *C* for pré-configurada em 0 ($C = 0$), tem-se a função NXOR de *A* e *B* na saída S_3. Por outro lado, se $C = 1$, tem-se a função XOR de *A* e *B* conforme mostrado na Tabela II. Ou seja, o Circuito *CC* é capaz de implementar a porta *C*-NOT.

TABELA I. Funções lógicas obtidas a partir do circuito C_1.

Porta C_1	S_1	S_2	S_3
$C=0$	$A+B$	$A \cdot B$	$\bar{A} \cdot B$
$C=1$	$A \cdot B$	$\overline{A+B}$	$A \cdot \bar{B}$

TABELA II. Funções lógicas obtidas a partir do circuito *CC*.

Circuito *CC*	S_1	S_2	S_3
$C=0$	A	B	$A \odot B$
$C=1$	A	B	$A \oplus B$

V. Conclusões

Este trabalho apresenta um sistema original baseado em dispositivos ópticos lineares para a realização probabilística de duas portas reversíveis, C_1 e C_2, utilizando qubits de estados coerentes. O sistema demonstra a viabilidade de implementar operações lógicas reversíveis em um contexto óptico, possibilitando aplicações em áreas como computação quântica e processamento de informações.

O sistema proposto apresenta uma probabilidade de sucesso de até 1/8 para ambas as portas reversíveis. Além disso, o sistema permite a pré-configuração de um dos estados de entrada para a obtenção de três funções lógicas (AND, OR, NAND e/ou NOR), simultaneamente, das demais entradas por cada processamento. Essa flexibilidade possibilita a implementação da porta *C*-NOT por meio do cascateamento de ambas as portas.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) - Código de Financiamento 001, da Funcap e do INCT-IQ.

Referências

[1] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, "Quantum computers," *Nature*, vol. 464, no. 7285, p. 45, 2010.

[2] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[3] A. Imamoglu, "Are quantum dots useful for quantum computation?" Physica E: *Low-dimensional Systems and Nanostructures*, vol. 16, no. 1, pp. 47–50, 2003.

[4] D. A. Herrera-Martí, A. G. Fowler, D. Jennings, and T. Rudolph, "Photonic imple-mentation for the topological cluster-state quantum computer," *Physical Review A*, vol. 82, no. 3, p. 032332, 2010.

[5] G. S. Uhrig, "Keeping a quantum bit alive by optimized -pulse sequences," *Physical Review Letters*, vol. 98, no. 10, p. 100504, 2007.

[6] You, J. Q., and Franco Nori. "Superconducting circuits and quantum information." *arXiv preprint quant-ph*/0601121, (2006).

[7] Y. Makhlin, G. Schön, and A. Shnirman, "Josephson junction quantum logic gates," *Computer physics communications*, vol. 127, no. 1, pp. 156–164, 2000.

[8] J. Schreier, A. A. Houck, J. Koch, D. I. Schuster, B. Johnson, J. Chow, J. M. Gambetta, J. Majer, L. Frunzio, M. H. Devoret, et al., "Suppressing charge noise decoherence in superconducting charge qubits," *Physical Review B*, vol. 77, no. 18, p. 180502, 2008

[9] I. L. Chuang, N. Gershenfeld, M. G. Kubinec, and D. W. Leung, "Bulk quantum computation with nuclear magnetic resonance: theory and experiment," *in Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 454, pp. 447–467, The Royal Society, 1998.

[10] Vandersypen, L., Steffen, M., Breyta, G. et al. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance". *Nature* 414, 883–887 (2001).

[11] L. M. K. Vandersypen, I. L. Chuang, "NMR techniques for quantum control and computation" *Reviews of Modern Physics*, January 2005.

[12] C. Ospelkaus, C. E. Langer, J. M. Amini, K. R. Brown, D. Leibfried, and D. J. Wineland, "Trapped-ion quantum logic gates based on oscillating magnetic fields," *Physical review letters*, vol. 101, no. 9, p. 090502, 2008.

[13] J. J. García-Ripoll, P. Zoller, and J. I. Cirac, "Speed optimized two qubit gates with laser coherent control techniques for ion trap quantum computing," *Physical Review Letters*, vol. 91, no. 15, p. 157901, 2003.

[14] A. Bermudez, P. O. Schmidt, M. B. Plenio, and A. Retzker, "Robust trapped-ion quantum logic gates by continuous dynamical decoupling", *Phys. Rev. A* 85, 040302(R) 2012.

[15] E. Knill, R. Laflamme, and G. J. Milburn, "A scheme for efficient quantum computation with linear optics", *nature*, vol. 409, no. 6816, p. 46, 2001.

[16] T. C. Ralph, A. Gilchrist and G. J. Milburn, “Quantum computation with optical coherent states”, *Phys. Rev. A*, 68, 042319, 2003.

[17] H. Jeong and M. Kim, “Efficient quantum computation using coherent states”, *Phys. Rev. A*, vol.65, 042305, 2002.

[18] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn, “Linear optical quantum computing with photonic qubits,” *Reviews of Modern Physics*, vol. 79, no. 1, p. 135, 2007.

[19] M.S.R. Oliveira; H.M. Vasconcelos and J.B.R Silva. "A probabilistic CNOT gate for coherent-state qubits". *Phys. Lett. A*, 377, 2821-2825, 2013.

[20] Rosa Silva, J.B., Ramos, R.V., "Smart generation of a tripartite GHZ- type state for coherent state qubit". *Opt. Commun*. 281(9), 2705 (2008).

[21] S. Glancy, J. LoSecco, H. Vasconcelos, and C. Tanner, “Imperfect detectors in linear optical quantum computers,” *Physical Review A*,vol. 65, no. 6, p. 062317, 2002.

[22] J. B. R. Silva and R. V. Ramos, “Implementations of quantum and classical gates with linear optical devices and photon number quantum non-demolition measurement for polarization encoded qubits,” *Physics Letters A*, vol. 359, no. 6, pp. 592–596, 2006.

[23] D. Copsey, M. Oskin, F. Impens, T. Metodiev, A. Cross, F. T. Chong, I. L. Chu- ang, and J. Kubiatowicz, “Toward a scalable, silicon-based quantum computing architecture,” *IEEE Journal of selected topics in quantum electronics*, vol. 9, no. 6, pp. 1552–1569, 2003.

[24] J. L. O’Brien, A. Furusawa, and J. Vučković, “Photonic quantum technologies,” *Nature Photonics*, vol. 3, no. 12, p. 687, 2009.

[25] Glauber, Roy, J. The Quantum Theory of Optical Coherence. *Physical Review*, v. 130, n. 6, p. 2529, 1963.

7º WECIQ

Ensinando o Protocolo BB84 com Simulações Interativas

Gisele Bosso de Freitas e Clovis Caface

***Resumo*— O Protocolo BB84 é essencial na criptografia quântica, assegurando a segurança na transmissão de dados sensíveis. Uma maneira eficaz de compreender esse protocolo é através do uso de simuladores em conjunto com estratégias didáticas que empregam metodologias ativas. Este estudo investiga como o QuVis, um simulador, combinado com abordagens ativas de ensino, pode ser uma ferramenta útil no ensino e na compreensão do Protocolo BB84.**

***Palavras-Chave*— Informação quântica, criptografia, emaranhamento.**

***Abstract*— The BB84 Protocol is essential in quantum cryptography, ensuring security in the transmission of sensitive data. An effective way to understand this protocol is through the use of simulators in conjunction with didactic strategies employing active methodologies. This study investigates how QuVis, a simulator, combined with active teaching approaches, can be a useful tool in teaching and understanding the BB84 Protocol.**

***Keywords*— Quantum information, cryptography, entanglement.**

I. Introdução

Entender os conceitos da mecânica quântica é um dos maiores desafios para os estudantes de física, mesmo quando realizam satisfatoriamente os cálculos, interpretar os resultados ainda é difícil. Neste contexto, para que o processo de aprendizagem seja mais eficiente e menos estressante, faz-se necessária a revisão dos métodos de ensino, mediante a adoção de estratégias e recursos pedagógicos que promovam uma melhor assimilação dos conteúdos por parte dos estudantes [10], [11]. Há evidências de que quando o conhecimento conceitual dos conteúdos é robusto, ele torna-se duradouro e para isso, as simulações interativas podem ser eficazes para ajudar os estudantes a construir representações mentais de conceitos físicos [7], [16]. Desse modo, as simulações interativas são ferramentas de aprendizagem que podem ser utilizadas em todos os níveis de ensino.

Este trabalho tem como finalidade propor uma estratégia de ensino-aprendizagem com a utilização de metodologias ativas [2] aliadas aos simuladores Quantum Cryptography (BB84 photon) [17] do *The Quantum Mechanics Visualisation Project* (QuVis) da Universidade de St Andrews no Reino Unido, para estimular o ensino de conteúdos de criptografia quântica através de uma representação visual dos conceitos abstratos da criptografia quântica, facilitando a compreensão dos estudantes sobre o funcionamento e a eficiência do computador quântico [15].

O QuVis é uma coleção de simulações interativas projetadas para ensinar conceitos de mecânica quântica, desde o ensino educacionais e desenvolvidas para otimizar sua eficácia até o superior. As simulações são baseadas em pesquisa educacional. As simulações ajudam os estudantes a fazer conexões entre diferentes representações, explorar relações entre quantidades e comparar situações diversas. O QuVis é financiado por várias instituições, incluindo a Universidade de St Andrews no Reino Unido. Além disso, o QuVis recebeu reconhecimento por sua excelência educacional, incluindo o Prêmio Physics Classics 2015 dos Recursos Educacionais Multimídia para Aprendizagem e Ensino Online (MERLOT) e o Prêmio de Excelência 2014 da Multimídia em Ensino e Aprendizagem de Física (MPTL). As simulações estão disponíveis gratuitamente para execução e download no site do QuVis [18].

Desenvolvido por Charles Bennett e Gilles Brassard em 1984, o protocolo BB84 [1], utiliza propriedades da mecânica quântica ([4], [5], [8]), como a superposição e o emaranhamento, para garantir a segurança na troca de chaves criptográficas entre dois pontos. Ao simular o Protocolo BB84 no QuVis, os estudantes podem observar como os bits quânticos (qubits) são codificados, transmitidos e decodificados, compreendendo os desafios e as soluções para a segurança da informação [9].

II. Criptografia e o protocolo BB84

A Criptologia é a ciência que estuda e desenvolve os conhecimentos e técnicas da comunicação segura. Ela é constituída pela Criptografia que é a ciência ou a arte de codificar informações e pela Criptoanálise que tem como objetivo estudar métodos para decodificar essa mesma informação [6], [13].

A criptografia analisa problemas de envio de informações que tem um emissor e um receptor de uma informação (ou mensagem), que desejam comunicar-se de forma segura através de um caminho não seguro onde um terceiro personagem pode ter acesso à mensagem trocada entre o emissor e o receptor. Para o processo de criptografar uma mensagem em que somente o emissor e o receptor possam ter conhecimento do significado, precisa-se da própria mensagem, de um algoritmo e de uma chave, que é uma determinada informação adicional ao algoritmo.

O surgimento da chave criptográfica revolucionou a segurança da comunicação ao permitir o uso de algoritmos para garantir o sigilo das mensagens, mesmo em canais não seguros. A segurança passou a ser centrada na chave, não na mensagem em si. As chaves podem ser classificadas em simétricas, onde emissor e receptor compartilham a mesma chave, e assimétricas, onde o emissor pública uma chave para

Gisele Bosso de Freitas, Centro de Ciências Exatas, Naturais e Tecnológicas - CCENT, Universidade Estadual da Região Tocantina do Maranhão - UEMASUL, Imperatiz-MA, e-mail: giselebosso@uemasul.edu.br; Clovis Caface, CCENT, UEMASUL, Imperatiz-MA, e-mail: clovis.caface@uemasul.edu.br.
Este trabalho foi parcialmente financiado pela UEMASUL.

para criptografar mensagens, mantendo uma chave secreta para decodificá-las. Embora os algoritmos assimétricos sejam mais lentos, são frequentemente utilizados para a troca de chaves, sendo a Cifra de Vernam o único algoritmo conhecido como absolutamente seguro, conforme a Teoria da Informação de Shannon [14].

A Cifra de Vernam, criada por Gilbert Vernam em 1917, é uma ferramenta que aplica uma chave qualquer no texto de interesse para produzir uma mensagem cifrada. Nessa cifra é usado os conceitos da teoria de Shannon e da aritmética dos restos [3]. Também a chave é usada como texto com o mesmo tamanho da mensagem que nunca é repetido. Assim as únicas restrições ao algoritmo são: usar uma chave do mesmo tamanho que a mensagem a ser cifrada; utilizar a chave somente uma vez, trocando de chave a cada vez que uma nova mensagem for transmitida.

Por exemplo, considere uma mensagem m, representada por uma sequência de dígitos binários. A chave utilizada, k, do mesmo tamanho que a mensagem, também representada por dígitos binários, mas com dígitos binários aleatórios. A mensagem criptografada, c, dada por uma soma de módulo 2 representada pelo símbolo $\oplus$ é:

$$c = m \oplus k, \qquad (1)$$

em que é a soma módulo 2 [9], que é o mesmo que fazer uma operação lógica (XOR) entre dois bits diferentes (mensagem e chave) resultando um valor lógico verdadeiro, ou fazer essa operação lógica entre dois bits iguais, resultando em um valor lógico falso. Como a chave possui dígitos aleatórios, a mensagem cifrada também possuirá.

Neste caso, a mensagem criptografada terá entropia máxima. Entropia, em termos gerais, é uma medida da incerteza em um sistema. Na teoria da informação de Shannon, a entropia é usada para quantificar o nível de imprevisibilidade em um conjunto de dados. Quanto maior a entropia, maior é a imprevisibilidade ou a falta de informações estruturadas no sistema, que aqui é uma mensagem. Como o alfabeto utilizado para escrever a mensagem é binário, cada letra ocorre com a mesma probabilidade, dessa forma a entropia $$H(c)$$ é:

$$H(c) = -\sum_{i=1} p_i \log_2(p_i) = -2 \log_2 2 + 2 \log_2 2 = 1. \qquad (2)$$

a representação do funcionamento da cifra para várias mensagens fica da seguinte forma:

Para escrever uma mensagem com o atual alfabeto da língua portuguesa, que tem 26 letras, basta fazer uma soma módulo 26. Se uma pessoa que não pode ter conhecimento da mensagem conseguir armazenar várias mensagens e ob- ter informações delas, é porque a chave foi utilizada mais de uma vez. Para entendermos o que isso significa, vamos considerar c_i as mensagens criptografadas, p_i as mensagens não-codificadas e k a chave.

Utilizando a propriedade comutativa da soma módulo 2 e a identidade $x\ x = 0$, é possível ver que se duas mensagens criptografadas foram utilizadas com a mesma chave, então uma

$$c_1 \oplus c_2 = p_1 \oplus p_2 \oplus k \oplus k = p_1 \oplus p_2.$$

(3) Porém, sua implementação prática é limitada devido à necessidade de uma chave única para cada mensagem, o que demanda
canais seguros para troca.

O conceito de criptografia quântica é definido graças a utilização de uma propriedade muito importante da Mecânica Quântica: a impossibilidade de haver cópia de uma informação quântica, ou seja, ela nos diz que não se pode obter informação de um estado quântico sem perturbar o sistema. Assim, o protocolo BB84 [1], utiliza dessa propriedade quântica para realizar trocas seguras de chaves e depois aplicar algum algoritmo clássico para realizar a troca da mensagem.

O protocolo BB84 funciona da seguinte maneira:

1) **Preparação dos qubits:** O emissor (Alice) gera uma sequência aleatória de qubits, como fótons, com estados de polarização desconhecidos. Cada qubit representa um bit da chave a ser compartilhada.
2) **Envio dos qubits:** Alice envia os qubits preparados para o receptor (Bob) através de um canal de comunicação, que pode ser vulnerável a interceptações.
3) **Escolha e medida das bases:** Tanto Alice quanto Bob escolhem aleatoriamente entre as bases de medição possíveis para cada qubit recebido. As bases podem ser, por exemplo, polarizadores a 0º (horizontal), 90° (vertical) ou 45º (diagonal).
   Medição dos qubits: Bob mede os qubits recebidos em uma das bases escolhidas. Após a medição, ele registra os resultados.
5) **Comparação das bases:** Alice e Bob comparam publicamente as bases escolhidas para cada qubit. Eles descartam os qubits para os quais eles escolheram bases diferentes.
6) **Correção dos erros:** Se Alice e Bob escolheram a mesma base, mas obtiveram resultados diferentes, isso indica a presença de um erro, como ruídos, pequenos defeitos no equipamento que faz o envio da mensagem, ou interceptação. Eles descartam esses qubits e repetem o processo.

7) **Amplificação de privacidade:** Após corrigirem os erros, Alice e Bob decidem usar um algoritmo de amplificação de privacidade para proteger sua chave contra terceira pessoa, que não poderia ter conhecimento da mensagem, conseguirá obter informação sobre as mensagens que não dependem da chave, ou seja, não é aleatória. Então possíveis interceptações. Mesmo que o receptor tenha medido o fóton usando a mesma base que Alice usou para polarizá-lo, qualquer alteração na polarização indicaria uma possível interceptação. Nesse caso, os interlocutores descartam a chave comprometida e repetem o protocolo até obterem uma chave segura.

8) **Obtenção da chave compartilhada:** Ao final do proto- colo, Alice e Bob possuem uma chave compartilhada de bits seguros, que podem ser utilizados para co- municações seguras utilizando técnicas de criptografia convencional.

Note que o protocolo tem a vantagem de conseguir perceber a presença de intrusos antes que alguma mensagem valiosa seja trocada.

III. PROTOCOLO BB84 NO SIMULADOR QUVIS

Baseado no texto [16], que destaca a importância das estratégias didáticas no uso eficaz de simulações interativas na sala de aula, elaborou-se atividades para ensinar conceitos e técnicas da computação quântica. As estratégias apresentadas [16] são a "Indagação Grupal", na qual o professor guia a exploração da simulação enquanto os alunos interagem com a ferramenta, promovendo a investigação por meio de questionamentos, as "Aulas Interativas com Demonstrações" (AID), que são centradas na análise e discussão de experimentos demonstrativos (reais ou virtuais) e a "Instrução por Pares" (perguntas Clicker), que envolve os estudantes na tomada de decisões através de perguntas de múltipla escolha. O texto [16] também reconhece que não existe uma única estratégia ideal, sugerindo que diferentes abordagens podem ser combinadas para atender às necessidades específicas de cada contexto educacional.

Todas as estratégias apresentadas em [16] estão fundamentadas em metodologias ativas de ensino, que colocam o aluno como protagonista do processo de aprendizagem [2]. Essas abordagens promovem a participação ativa dos alunos, a construção do conhecimento e o desenvolvimento de habilidades cognitivas e metacognitivas, contribuindo para ambientes de aprendizagem mais dinâmicos, engajadores e eficazes. Essas abordagens enfatizam a flexibilidade das simulações, que podem ser usadas em diferentes momentos do ensino e destaca a importância da interação dos alunos para promover um aprendizado significativo.

O *Quantum Cryptography (BB84 photons)* [18] é uma das simulações interativas disponíveis no QuVis para o Protocolo BB84, permitindo que os estudantes, tanto do nível médio quanto do superior, explorem as nuances da criptografia quântica de maneira prática e envolvente. A simulação está disponível em inglês ou alemão, no entanto é possível traduzi- la satisfatoriamente para o português através da ferramenta de tradução do navegador Google Chrome com o botão direito do mouse, como mostrado na Figura (1).

Ao iniciar a simulação, uma visão introdutória é apresentada, com os objetivos de aprendizado (Figura 1). O foco principal é ajudar o estudante a entender como Alice e Bob podem gerar uma chave segura e detectar a presença de um espião (Eve) em seu experimento. Ao acessar a aba do simulador, é possível configurar os fótons e as polarizações. Alice tem uma fonte de fótons e um polarizador. Ela envia fótons para Bob, que estão polarizados aleatoriamente em uma das quatro direções possíveis - vertical, horizontal, diagonal e anti-diagonal. Estas direções representam diferentes estados quânticos. Essa etapa demonstra como a informação é representada e manipulada em nível quântico.

A seguir, Bob utiliza um analisador de polarização e um detector de fótons para medir a polarização dos fótons que ele recebe. Ele escolhe aleatoriamente uma das duas bases (horizontal/vertical ou diagonal/anti-diagonal) para a medição. Se Bob escolher a mesma base que Alice usou para enviar o fóton, e nenhum espião estiver presente, o resultado de Bob será o mesmo que o estado enviado por Alice, veja "*Most recent key bits*"na Figura (2).

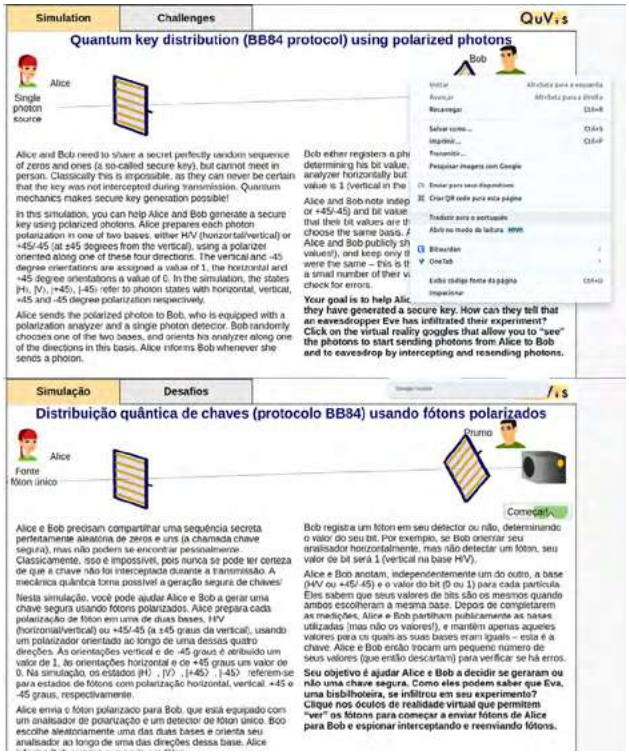


Fig. 1

CAPTURA DE TELA DA TRADUÇÃO PARA O PORTUGUÊS NO GOOGLE CHROME. **FONTE:** PRÓPRIA, 2024.

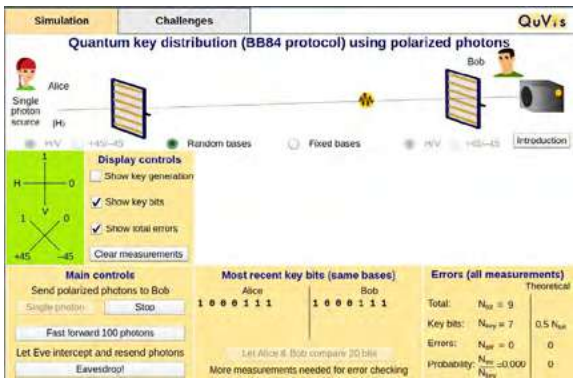


Fig. 2

CAPTURA DE TELA COM O CENÁRIO DA COMUNICAÇÃO ENTRE ALICE E BOB. **FONTE:** PRÓPRIA, 2024.

No QuVis Quantum Cryptography (BB84 photon), os alunos podem observar como os qubits se comportam durante a transmissão e como fatores externos podem afetar a segurança da comunicação. Esta etapa demonstra os desafios enfrentados na transmissão segura de informações quânticas.

Há um cenário onde Eve intercepta e reenvia os fótons, mudando potencialmente seu estado. Se Eve escolher a base errada por acaso, isso introduzirá erros nos resultados de Bob, que são detectáveis quando Alice e Bob comparam uma seleção de seus resultados de medição, possibilitando a

verificação de *Eavesdropping* (Espionagem) (Figura 3).

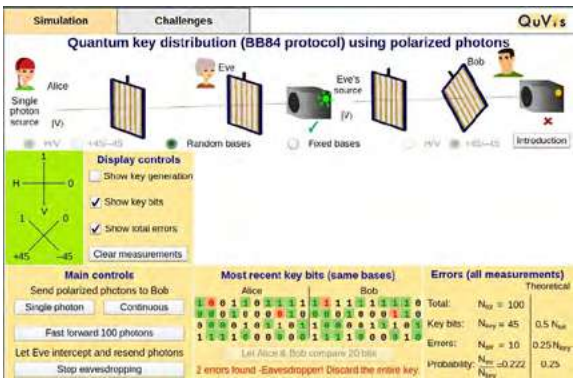


Fig. 3

CAPTURA DE TELA DO CENÁRIO DA COMUNICAÇÃO ENTRE ALICE E BOB COM INTERCEPTAÇÃO DA CHAVE POR EVE. **FONTE:** PRÓPRIA, 2024.

A simulação fornece visualizações gráficas que mostram a base e o resultado de cada medição individual, os bits de chave mais recentes e o número de bits de chave e erros para todas as medições realizadas. Isso permite analisar os casos em que as medições concordam (quando a mesma base é usada) e aqueles em que não (indicando possível interceptação).

Diferentes cenários com diferentes configurações, podem ser explorados pelos estudantes, por exemplo mudar as bases usadas por Alice e Bob, inserir ou não a Eve e observar como isso afeta a segurança da chave gerada. No destino, os qubits recebidos são medidos em uma base de polarização escolhida aleatoriamente. Os alunos podem experimentar com diferentes bases de medição e observar como isso afeta a decodificação da informação. A comparação das bases de codificação e medição é fundamental para determinar se a informação foi corretamente decodificada ou não.

Há também uma série de desafios (*Challenges* na Figura 4), que são alterados a cada vez que a página web é reiniciada, com os quais os estudantes podem testar seus conhecimentos pois proporciona o retorno (resposta) instantaneamente. Além disso, os professorem podem utilizá-los como base para ela- borar atividades e questões a serem desenvolvidas ao longo de uma aula que utilize as estratégias didáticas mencionadas inicialmente [16].

Utilizando a simulação, os estudantes podem entender claramente que enquanto as criptografias clássicas se baseiam em operações matemáticas complexas e dificuldade computacional, o BB84 utiliza qubits, as unidades fundamentais de informação quântica, se beneficiando das propriedades únicas da física quântica, como o princípio da incerteza de Heisenberg e a impossibilidade de clonagem quântica, para garantir a segurança da comunicação.

IV. METODOLOGIA PARA AS AULAS INTERATIVAS COM DEMONSTRAÇÕES (AID)

Aulas Interativas com Demonstrações (AID), ou ILD, na sigla em inglês para *Interactive Lecture Demonstration*, requer o uso de uma folha de predições para o estudante. No entanto,

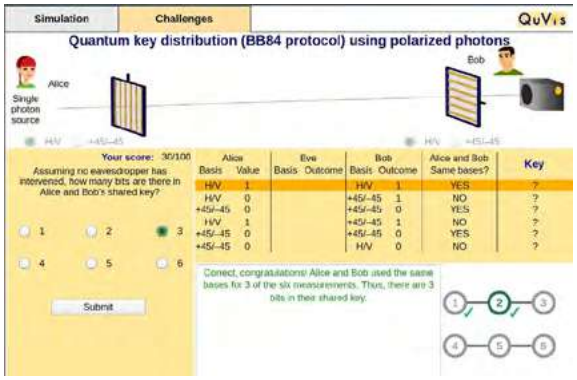


Fig. 4

CAPTURA DE TELA DE UM DOS DESAFIOS DA SIMULAÇÃO. **FONTE:** PRÓPRIA, 2024.

pode ser melhor para o docente preparar primeiro uma folha que antecipe as respostas dos alunos. Um modelo pode ser visto em [12]. A seguir, há um exemplo de plano de atividade usando o QuVis para explorar o Protocolo BB84, utilizando os conceitos das AID, com o objetivo de compreender os princípios do Protocolo BB84 e sua aplicação na criptografia quântica, utilizando o QuVis para simular o processo de codificação, transmissão e decodificação de qubits.

A. Procedimento

Inicialmente divide-se os alunos em grupos (dois a três integrantes) e distribui-se as "folhas de predições"[12] que descrevem a atividade e fornecem espaços para que os alunos registrem suas predições e conclusões.

Em seguida, apresenta-se o conceito do Protocolo BB84, explicando sua importância na segurança da comunicação quântica. Projeta-se a simulação QuVis Quantum Cryptography (BB84 photons) [17] para toda a classe e segue-se os oito passos da estratégia das AID para cada etapa da simulação (sem e com espionagem):

1) Descrição do experimento e/ou situação problemática.
2) Predição individual dos alunos e registro nas "folhas de predição" (Figura 5).
3) Discussão em grupos pequenos sobre as predições.
4) Compartilhamento das predições mais comuns em uma discussão grupal.
5) Registro das predições grupais nas "folhas de predição".
6) Demonstração da simulação pelo professor, destacando os resultados.
7) Descrição dos resultados pelos alunos, com espaço para perguntas adicionais.
8) Discussão sobre outras situações físicas relacionadas ao conceito e preparação para a próxima etapa da atividade.

Após explorar todas as etapas do Protocolo BB84 com o QuVis Quantum Cryptography (BB84 photons), estimula-se uma reflexão sobre os resultados e a importância da criptografia quântica na segurança da informação, encorajando os alunos a registrar suas conclusões nas "folhas de previsões" e a

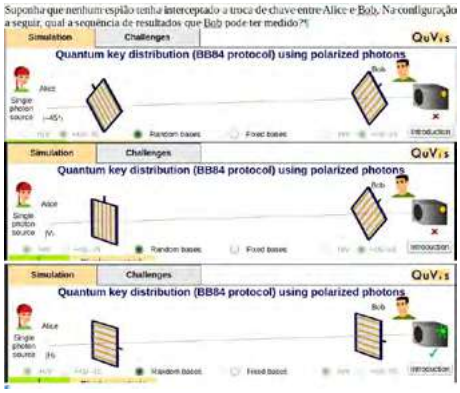


Fig. 5

UMA DAS PERGUNTAS DA FOLHA DE PREVISÕES EM UMA AULA INTERATIVA COM DEMONSTRAÇÕES (AID). **FONTE:** PRÓPRIA, 2024.

compartilhar ideias e descobertas adicionais em uma discussão final para sintetizar o que foi aprendido.

Durante a atividade, o professor deve fornecer orientação e suporte aos alunos, incentivando a participação ativa e a compreensão dos conceitos. É importante que a simulação seja conduzida de forma clara, dinâmica e envolvente, garantindo que os alunos compreendam cada etapa do Protocolo BB84 e suas implicações na criptografia quântica.

Essa atividade proporciona uma oportunidade interativa, visual e prática para os alunos explorarem os princípios do Protocolo BB84 utilizando uma simulação, promove o aprendizado conceitual e a interpretação de representações de fenômenos físicos.

V. CONCLUSÕES

Ao explorar o Protocolo BB84 com o QuVis, os alunos desenvolvem uma melhor compreensão dos princípios quânticos subjacentes, como a superposição e o emaranhamento. Eles aprendem como esses fenômenos são aproveitados para garantir a segurança na comunicação de informações sensíveis. Além disso, se houver a possibilidade de cada estudante manipular o QuVis em sala de aula, eles podem experimentar com diferentes configurações e cenários, fornecendo respostas imediatas sobre suas ações. Eles podem iterar sobre suas decisões e observar como estas afetam a segurança e eficiência do Protocolo BB84. Essa abordagem de aprendizado prático e interativo com problemas reais é essencial para uma apren- dizagem mais ativa.

Espera-se que o conteúdo deste texto possa fomentar a utilização de simulações de forma eficaz em sala de aula em nível superior, além das pesquisas educacionais em torno dessa ferramenta, como o impacto da combinação com experimentos reais e o impacto no desenvolvimento de habilidades na formação de professores no uso dessas ferramentas. Trabalhos futuros incluirão a implementação dessas atividades em disciplinas de óptica, física moderna, além de mecânica quântica, com o auxílio de instrumentos de coleta de dados.

AGRADECIMENTOS

À Universidade Estadual da Região Tocantina do Maranhão - UEMASUL pelo apoio financeiro.

REFERÊNCIAS

[1] BENNETT, Charles H.; BRASSARD, Gilles. *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing* – Bangalore, Índia: 1984

[2] BORGES, Tiago Silva; ALENCAR, Gidélia. Metodologias ativas na promoção da formação crítica do estudante: o uso das metodologias ativas como recurso didático na formação crítica do estudante do ensino superior. *Cairu em revista*, v. 3, n. 4, p. 119-143, 2014.

[3] DOMINGUES, Hygino H.; IEZZI, Gelson. *Álgebra moderna*. reform. São Paulo: Atual, 2003.

[4] HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. *FUNDAMEN- TOS DA FÍSICA v.4: óptica e física moderna*: 5°.ed. Rio de Janeiro: Editora: LTC, 2010.

[5] HEWITT, Paul G. *Física Conceitual:* 11°.ed. São Paulo: Bookman, 2011.

[6] HEYS, Howard M. A tutorial on linear and differential cryptanalysis. *Cryptologia*, v. 26, n. 3, p. 189-221, 2002.

[7] KOHNLE, Antje et al. A new multimedia resource for teaching quantum mechanics concepts. American Journal of Physics, v. 80, n. 2, p. 148- 153, 2012.

[8] MAHON, José R.P.*MECÂNICA QUÂNTICA: desenvolvimento contem- porâneo com aplicações*: Rio de Janeiro: Editora: LTC, 2011.

[9] MARQUEZINO, Franklin de Lima. *Estudo Introdutório do Protocolo Quântico BB84 para Troca Segura de Chaves* – Rio de Janeiro, RJ: CBPF, 2004.

[10] MASON, Andrew; SINGH, Chandralekha. Do advanced physics students learn from their mistakes without explicit intervention?. American Journal of Physics, v. 78, n. 7, p. 760-767, 2010.

[11] MCKAGAN, S. B.; PERKINS, K. K.; WIEMAN, C. E. Deeper look at student learning of quantum mechanics: The case of tunneling. Physical Review Special Topics-Physics Education Research, v. 4, n. 2, p. 020103, 2008.

[12] PhET Interactive Simulations. Aulas Interativas com Demonstrações. Modelo de Folha de Previsão de AID. Disponível em: https://o365coloradoedu.sharepoint.com/:w:/s/PHYS-phet-pilot/EUdIYK5nBzBGhxSs6jHnylQBjkpE_8BtjkZ9qqwso5JqZQ?e=1shcPC. Acesso em: 24. abr. 2024.

[13] RIVEST, Ronald L. Cryptography. *In: Algorithms and complexity*. Elsevier, 1990. p. 717-755.

[14] SHANNON, Claude Elwood. A mathematical theory of communication. *The Bell system technical journal*, v. 27, n. 3, p. 379-423, 1948.

[15] SILVA, Thiago. F. *Transmissão óptica de bits quânticos codificados em freqüência com sincronismo por WDM* – Rio de Janeiro, RJ: PUC-RIO, 2008

[16] TAVARES, Diana Berenice López. Estrategias didácticas para el uso eficaz de simulaciones interactivas en el aula. *Lat. Am. J. Sci. Educ*, v. 7, p. 12019, 2020.

[17] University of St Andrews QuVis Quantum Mechanics Visualization Project. Quantum Cryptography (BB84 photon), 2023. Simulação da distribuição de chave quântica utilizando fótons polarizados. Disponível em: <https://www.st-andrews.ac.uk/physics/quvis/simulations_html5/sims/BB84_photons/BB84_photons.html>. Acesso em: 12. abr. 2024.

[18] University of St Andrews QuVis Quantum Mechanics Visualization Project. 2023. Disponível em: https://www.st-andrews.ac.uk/physics/quvis/index.html. Acesso em: 12. abr. 2024.

7º WECIQ

Analysis of the Quantum Algorithm HHL for the generation of SVMs on NISQ Quantum Devices

Gabriela Pinheiro, Instituto de Computação, Universidade Federal Fluminense (UFF), Niterói-RJ, e-mail: gabrielapc@id.uff.br; Luis Antonio Kowada, Instituto de Computação, Universidade Federal Fluminense (UFF), Niterói-RJ, e-mail: luis@ic.uff.br. This work has been supported by the CNPq (project n.101170/2023-8) and by the FAPERJ (project n.260003/015313/2021).

Gabriela Pinheiro e Luis Antonio Kowada

***Abstract*— Support Vector Machine (SVM) is considered one of the main classification Machine Learning algorithms. Following the original formulation, an SVM generation has quadratic complexity, leaving room for exploring better resolution methods. One way to enhance its efficiency is by utilizing Quantum Computing algorithms, such as the HHL. The challenge with using a quantum algorithm is the type of Quantum Computers currently available, the Noisy Intermediate-scale Quantum(NISQ) Devices, where the noise interference creates measurement errors that generate divergent results from the expected values. This work presents a performance analysis of a Quantum Machine Learning algorithm that uses the HHL for SVM generation on a currently available NISQ quantum device, demonstrating that it is already possible to obtain results close to the expectation with the noise influence and even surpass it in a noiseless scenario.**

Keywords*— *Quantum Machine Learning, HHL, Machine Learning, Quantum Computation.

I. Introduction

The Support Vector Machine(SVM) is considered one of the most powerful classification algorithms which, due to its strong theoretical foundations and generalization capability, is widely used in relevant applications such as bioinformatics and image classification [3]. The original formulation of the algorithm has a quadratic complexity. To reduce the complexity of the algorithm, a least-squares reformulation was applied on the original version of the SVM [15] , transforming it on a system of linear equations. This transformation allows for more efficient linear systems resolution techniques to be applied, such as the quantum algorithm HHL.

Quantum Computing allows the use of quantum mechanics to obtain information processing advantages over classical computers for specific cases. One of them being the resolution of a linear system of equations using the HHL.

HHL [8] is a quantum algorithm where it is possible to extract information about the solution of a linear system of equations with exponential advantage over classical algorithms. The algorithm is used in a variety of quantum machine learning applications, such as linear regression and SVMs [6].

Called the NISQ [7] era of Quantum Computing, the current state of Quantum Computing presents a number of challenges when implementing those algorithms on real devices. Where the errors created with the noise influence yields results that differ from the expectations, going as fair as being completely incoherent in extreme cases.

The objective of this paper is to analyze the performance of two-dimensional SVMs generated using the HHL algorithm on a NISQ quantum device.

II. Fundamental Concepts

A. SVM

Support Vector Machine (SVM) [4] is a Machine Learning algorithm created originally for binary classification of data, where a dataset is divided in two distinct classes and the SVM needs to discover to which class a new element of the dataset belongs to.

The algorithm works by non-linear mapping the data to points in a hyperspace where the dimension is defined by the number of parameters used. The goal is to find a linear decision hyperplane dividing the data from both classes, maximizing the distance between them. As a result, the class of new data can be verified by its position relative to the hyperplane.

The hyperplane to be found is represented by the equation $\vec{w} \cdot \vec{x} - b = 0$, and the classes are represented by the labels 1 and -1. A data point $\vec{x}$ belongs to the 1 class if $\vec{w} \cdot \vec{x} - b \geq 1$ is true, or belongs to the -1 class if $\vec{w} \cdot \vec{x} - b \geq -1$ is true. Figure 1 shows an example of a SVM in a two-dimensional hyperspace.

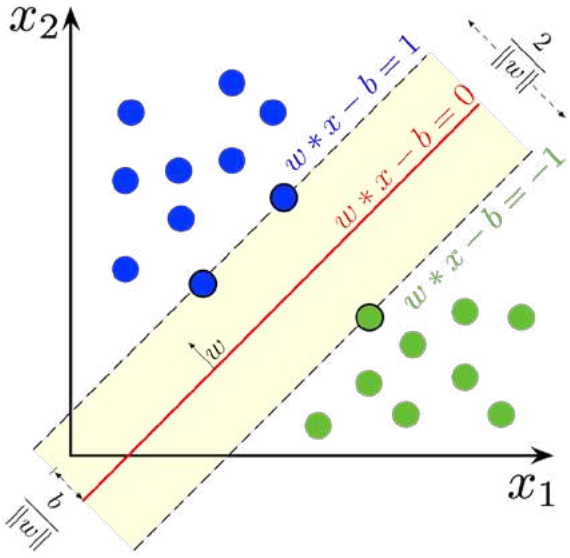


Fig. 1. Maximum-margin hyperplane in a two-dimensional hyperspace. Source: [9].

1) Least-squares reformulation: SVM's original formulation goal is to maximize the distance margin $\frac{2}{||\vec{w}||}$between the two classes, which is equivalent to minimize $\frac{||\vec{w}||}{2}$ with the restriction $y(_j\vec{w} \cdot \vec{x_j} + b) \geq 1$, for every x_j in the dataset.

A least-squares reformulation[13] of the problem was proposed, transforming the SVM into the solution of the linear system represented by Equation (1).

$$F\begin{pmatrix} b \\ \vec{a} \end{pmatrix} \equiv \begin{pmatrix} 0 & \vec{1}^T \\ \vec{1} & K+\gamma^{-1}I \end{pmatrix} \begin{pmatrix} b \\ \vec{a} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{y} \end{pmatrix}, \quad (1)$$

where K the kernel matrix, a $M \times M$ matrix created using the function $K_{ij} = k(\vec{x_i}, \vec{x_j}) = \vec{x_i} \cdot \vec{x_j}$ with the M training vectors, being $\vec{y} = (y_0, y_1, \ldots, y_{M-1})$ and $\vec{1} = (1, \ldots, 1)$. I represents the identity matrix and γ is the user-defined training error weight. Resulting in a (M + 1) × (M + 1) matrix F in which the SVM parameters become represented as a function of b and $\vec{a}$ and the class of a data point $\vec{x_t}$ is now obtained with the condition in (2).

$$y_i = \begin{cases} +1 \ \ if \ \sum_{j=0}^{M-1} a_i k(\vec{x_i}, \vec{x_j}) + b \geq 0 \\ -1 \ \ if \ \sum_{j=0}^{M-1} a_i k(\vec{x_i}, \vec{x_j}) + b < 0 \end{cases}, \quad (2)$$

B. Quantum Computation

Quantum Computation is the field of computation responsible for studying information processing through the use of quantum mechanical systems [11], which makes it possible to obtain algorithms with processing advantages over classical systems.

Named after the type of quantum computers available, Quantum Computation is currently at the NISQ (Noisy Intermediate-Scale Quantum) era, where these computers processing capacity is still limited, between 50 to 100 qubits [12]. Another limiting factor is the high rate of noise, which are measuring errors that occur due to different reasons, such as the difficulty to maintain the system isolated, multi-qubit operations and the circuit transpilation that is necessary to run circuits on real devices and usually results in bigger circuits than the original.

C. HHL algorithm

Named after its creators, the HHL [8] is a quantum algorithm for the resolution of linear systems. Considering the system $A\vec{x} = \vec{b}$, given $\vec{b}$ and the matrix A, it is possible to extract information about the solution $\vec{x}$ more efficiently, being exponentially faster than any classical algorithm for specific systems. A more in depth explanation and circuit tutorial can be found in [1].

III. Implementation

In order to reduce the size and length of the generated circuit for a SVM, The b parameter was fixed in zero [16], implying that the generated hyperplane passes through the origin of the hyperspace. This was done in order to simplify the system, because with b being zero it is possible to remove the first line and column of F due to their values also becoming 0 after the multiplication, resulting in the Equation (3).

$$F(\vec{a}) \equiv (K + \gamma^{-1}I)(\vec{a}) = (\vec{y}), \quad (3)$$

As a result, F becomes a $M \times M$ matrix and the only parameter that must be found is $\vec{a}$. Because $\vec{a}$ is a vector, the ratio between its components is enough to calculate the angle and trace the hyperplane. Resulting in an ideal scenario for the use of HHL, which is able to calculate that ratio efficiently. For that reason, this was the system selected to be used for all of the SVMs created.

A. Implementation

The circuits were implemented using the Python programming language and the Qiskit library, an open-source quantum toolkit developed by IBM [5]. Each HHL circuit used the same structure shown in the tutorial [10], where initially $\vec{b}$ is mapped to the circuit, followed by a Quantum Phase Estimation(QPE), a controlled rotation R and finishes with an inverse QPE. The generated circuit is represented on Figure 2.

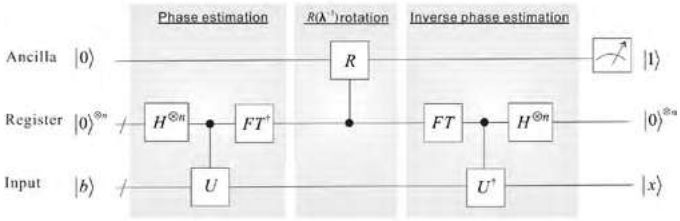


Fig. 2. Generic HHL circuit. Source: [2]..

IV. Experiments

A. Breast Cancer Wisconsin Dataset

In order to validate the circuit automation, the Breast Cancer Wisconsin (Diagnostic) dataset [14] was chosen to perform the tests. The dataset was created from 569 images of breast mass fine needle aspirates (FNA) that were processed and resulted in 30 continuous features.

For each test, the dataset was divided into a training set containing the first 369 elements and a test set with the remaining 200. The Kernel matrix was created with the average value of the features of the benign and malignant training data, and each element was divided by its norm.

After classically testing all possible combinations of feature pairs, the circuits for all SVMs with accuracy higher than 75% where simulated and the three SVMs generated from those circuits with higher accuracies were chosen to compose the test group to be implemented on real quantum devices.

With the test group defined, three types of SVMs generation methods were applied, the first being the classical solution of the linear system, to serve as a performance baseline, the second being the noiseless simulation of the circuit, to represent its behavior on an ideal scenario, a the last one being an execution on the *ibm_sherbrooke*, a 128 qubits quantum computer from IBM. To analyze how the results may vary for each execution, five SVMs of each type were generated for every element of the test group and 10.000 measurements were performed at all circuit simulations and executions.

The performance of the SVMs were judged based on four different metrics: Accuracy, F1 score, Precision and Recall. Represented in Equation (4) with T P being the true positive

values, T N the true negative values, F P the false positive values and F N the false negative values.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

[1]

V. RESULTS

This section presents the results obtained from each subject on the test group divided in three subsections, concluding with a global analysis of the results.

A. Smoothness Mean and Concave Points Worst

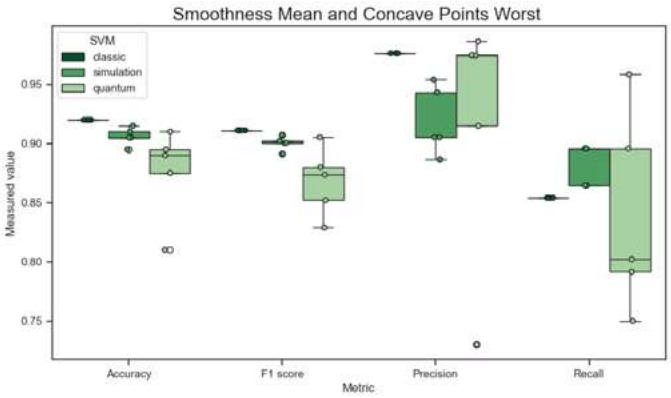


Fig. 3. Accuracy, F1 score, Precision and Recall values obtained from SVMs using Smoothness Mean and Concave Points Worst features, generated by the exact solution, simulation and execution on a real quantum computer, respectively.

Figure 3 shows the results obtained from all SVMs generated using the Smoothness Mean and Concave Points Worst features. With median accuracy values of 90.5% and 89% over the 92% expected, median F1 score values of 90% and 87.4% over the expected 91.1% and the small variation between the results indicates that SVMs generated from simulations and execution on quantum devices have predictive performances close to their classical counterpart. They present more variation when analyzing their precision and recall but both obtained recall values above the expected in at least two cases and one quantum generated SVM obtained a better precision than the expectation.

B. Fractal Dimension Mean and Concave Points Worst

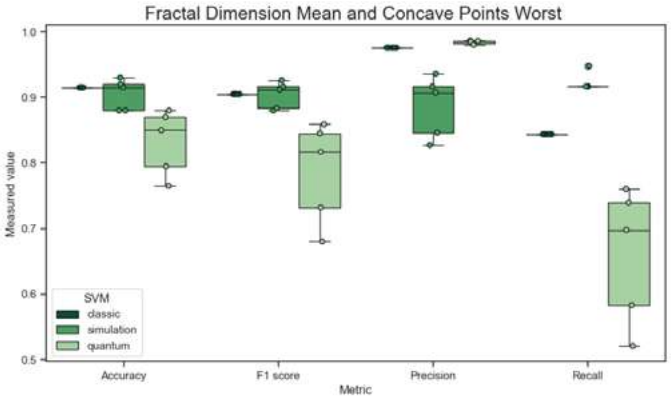


Fig. 4. Accuracy, F1 score, Precision and Recall values obtained from SVMs using Fractal Dimension Mean and Concave Points Worst features, generated by the exact solution, simulation and execution on a real quantum computer, respectively.

Figure 4 shows the results obtained from all SVMs generated using the Fractal Dimension Mean and Concave Points Worst features. With the same median accuracy value of the expected 91.5% and a higher F1 score median value of 91.2% over the 90.5% expected, the SVMs generated from simulations tend to have better predictive performances than their classical counterpart, while the quantum generated SVMs presented a inferior predictive performance but was more precise in all cases. All SVMs generated from simulations also presented recall values above expectation.

C. Smoothness Worst and Concave Points Worst

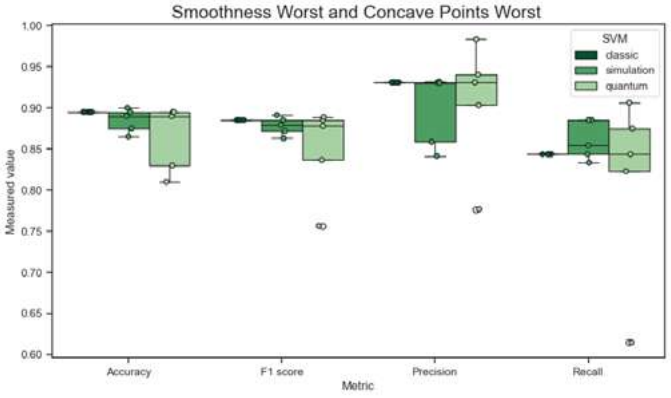


Fig. 5. Accuracy, F1 score, Precision and Recall values obtained from SVMs using Smoothness Worst and Concave Points Worst features, generated by the exact solution, simulation and execution on a real quantum computer, respectively.

Figure 5 shows the results obtained from all SVMs generated using the Smoothness Worst and Concave Points Worst features. With the same median accuracy value of 89% over the expected 89.5% and median F1 score values of 87.9% and 87.8% respectively over the expected 88.5, the SVMs generated from simulations and executions on quantum devices presented the closest predictive performances to their classical counterpart from all tested pairs. They also present precision and recall above the expected in at least one case each.

In a global analysis of all SVMs, the quantum generated SVMs presented median accuracy and F1 score values within a less than 10% range from what was expected, with all of them being

above 80%. Demonstrating that it is already possible to constantly obtain SVMs with high predictive performances on current noisy quantum computers.

When considering only the maximal value obtained for each metric the SVMs generated from simulations obtained the higher accuracy and F1 score values, being 93% and 92.6% respectively, while the quantum generated SVMs obtained the higher precision and recall values, being 98.6% and 95.8% respectively. Indicating that the SVMs generated with Quantum Machine Learning can surpass their classical counterparts on ideal and noisy scenarios.

It is also worth noticing that the Concave Points Worst feature appears on all of the test group pairs, indicating a high separation between values from different classes. This separation is confirmed by the box-plot graph presented in Figure 6, with both boxes having no intersections.

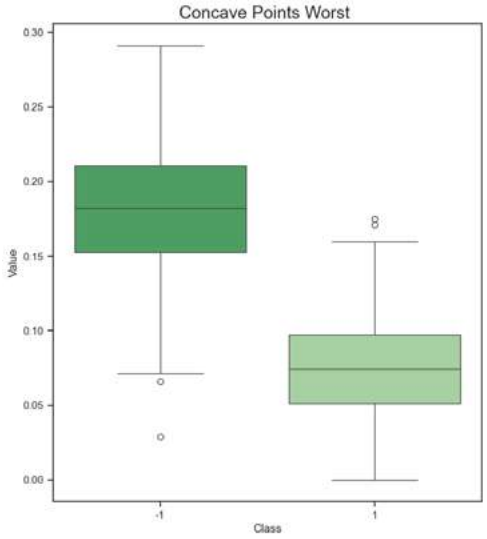


Fig. 6. Comparison between the Concave Points Worst feature values ranges for both classes.

VI. Conclusion

The experiments showed that the SVMs generated using the HHL algorithm can surpass the performance of the classically generated ones, demonstrating the viability of this Quantum Machine Learning algorithm.

The small decrease in predictive performances from SVMs generated on a noisy quantum computer indicates a level of noise resistance from the algorithm alongside the possibility of being successfully used with currently available quantum devices, with an expectation to achieve performances above their classical counterparts on ideal scenarios. In conclusion, this paper presented an analysis of the performance of the Quantum Algorithm HHL for generating SVMs on NISQ Quantum Devices.

VII. Future works

Future works will be focused on extending the tests to different use cases and datasets in order to further analyze the algorithm performance. Alongside the search for the HHL circuit optimizations.

Acknowledgments

This work has been supported by the CNPq (101170/2023-8) and by the FAPERJ (260003/015313/2021).

Referências

[2] Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, Nathan Lemons, et al. Quantum algorithm implementations for beginners. arXiv preprint arXiv:1804.03719, 2018.

[3] X-D Cai, Christian Weedbrook, Z-E Su, M-C Chen, Mile Gu, M-J Zhu, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Experimental quantum computing to solve systems of linear equations. Physical Review Letters, 110(23):230501, 2013.

[4] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408:189–215, 2020.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20:273–297, 1995.

[6] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In APS March meeting abstracts, volume 2018, pages L58–003, 2018.

[7] Bojia Duan, Jiabin Yuan, Chao-Hua Yu, Jianbang Huang, and Chang-Yu Hsieh. A survey on HHL algorithm: From theory to application in quantum machine learning. Physics Letters A, 384(24):126595, 2020.

[8] Laszlo Gyongyosi and Sandor Imre. A survey on quantum computing technology. Computer Science Review, 31:51–71, 2019.

[9] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. Physical Review Letters, 103(15):150502, 2009.

[10] CC BY-SA 4.0 via Wikimedia Commons Larhmam. Maximum-margin hyperplane and margin for an svm trained on two classes. samples on margins are called support vectors., 2018. File: SVM_margin.png.

[11] Hector Jose Morrell Jr, Anika Zaman, and Hiu Yung Wong. Step-by-step hhl algorithm walkthrough to enhance the understanding of critical quantum computing concepts. arXiv preprint arXiv:2108.09004, 2021.

[12] Michael A Nielsen and Isaac L Chuang. Quantum computation and quantum information. Cambridge University Press, 2010.

[13] Edwin Pednault, John A Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, Erik W Draeger, Eric T Holland, and Robert Wisnieff. Pareto-efficient quantum circuit simulation using tensor contraction deferral. arXiv preprint arXiv:1710.05867, 2017.

[14] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. Physical Review Letters, 113(13):130503, 2014.

[15] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In Biomedical image processing and biomedical visualization, volume 1905, pages 861–870. SPIE, 1993.

[16] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. Neural processing letters, 9:293–300, 1999.

[17] Jiaying Yang, Ahsan Javed Awan, and Gemma Vall-Llosera. Support vector machines on noisy intermediate scale quantum computers. arXiv preprint arXiv:1909.11988, 2019.

7º WECIQ

Enhanced Channel Estimation and Data Detection in OFDM Systems without Cyclic Prefix using Quantum Machine Learning Algorithms

Demerson N. Gonçalves and João T. Dias

***Abstract*— Channel estimation in OFDM systems requires minimal complexity with one-tap equalizers. However, this depends on cyclic prefixes, which must be sufficiently large to cover the channel impulse response. Conversely, the use of cyclic prefix (CP) decreases the useful information that can be conveyed in an OFDM frame, thereby degrading the spectral efficiency of the system. In this context, we propose the use of quantum kernel in support vector machine (SVM) algorithm for channel estimation and symbol detection in OFDM systems without CP and compare its performance with the LS and the classic support vector regressor (SVR) for channel estimation and coherent demodulation for symbol detection. The viability of our approach is substantiated by computational simulation results obtained in frequency selective channel models with the presence of Gaussian noise.**

Keywords*— *Channel estimation, symbol detection OFDM, SVR, QSVR.

I. INTRODUÇÃO

Orthogonal Frequency Division Multiplexing (OFDM) has emerged as a prominent scheme for high-bit-rate wireless networking standards [1]-[6]. Its primary advantage lies in its ability to eliminate intersymbol interference (ISI) and intercarrier interference (ICI) without necessitating complex equalization filters at the receiver. While ISI is mitigated through the use of a cyclic prefix, ICI poses challenges in dynamic channels or when there are local oscillator mismatches with high carrier frequency offsets (CFOs). Common strategies to enhance the useful data rate in OFDM systems include reducing the pilot rate, expanding the number of subcarriers, or increasing the modulation order. However, each approach introduces its own set of complications. Decreasing the pilot rate compromises channel estimation accuracy and renders the system more vulnerable to rapid channel variations observed in dynamic channels. Expanding the number of subcarriers, while maintaining the same bandwidth to avoid interference in adjacent channels, escalates both computational complexity and the required clock speed for signal processing. Lastly, increasing the modulation order exacerbates the bit error rate (BER) at the receiver.

In the context of advancing digital communications systems, Support Vector Regression (SVR) has proven effective, particularly in addressing challenges like channel estimation in scenarios with non-linearities [7]. To optimize SVR performance, selecting an appropriate kernel based on Mercer's conditions is essential, alongside parameter adjustments for achieving optimal regression outcomes [8]-[9].

Demerson N. Gonçalves is professor at Collegiate of Mathematics, CEFET/RJ, Petrópolis, RJ, E-mail: demerson.goncalves@cefet-rj.br. João T. Dias is professor at the Department of Telecommunications, CEFET/RJ, Maracanã, RJ, E-mail: joao.dias@cefet-rj.br.This work was partially financed by the program "GPESq-CEFET/RJ".

In OFDM systems, SVR emerges as a powerful tool for efficient and robust channel estimation and data detection. By formulating channel estimation as an SVR problem, we aim to find the best-fitting hyperplane that minimizes error between predicted and actual channel coefficients. SVR achieves this by mapping received signals to true channel coefficients, optimizing a loss function, and considering a regularization parameter to manage model complexity. This approach enables efficient channel response estimation, even amidst noise and interference.

In recent years, there has been a growing interest in quantum computing as a potential solution to the computational challenges faced by modern Machine Learning (ML) systems. Quantum computing has made significant progress, promising faster computations across various scientific and industrial applications. Some studies assert time advantage [10]-[17], while others showcase enhancements in accuracy and convergence [18]-[21]. In this study, we propose using a quantum support vector regressor (QSVR) to address channel estimation and quantum support vector classifier (QSVC) to address data detection in an OFDM system without CP, especially in scenarios characterized by frequency-selective channels and Gaussian noise presence.

This article is divided as follows: in section II, the OFDM system are described. The channel estimations and data detection models are presented in section III. In section IV, the QSVR and QSVC are presented. The results are shown in section V, and conclusions are made in section VI.

II. OFDM SYSTEM MODELING FRAMEWORK

The block diagram of the implemented OFDM system is shown in Fig. 1. In this system, b are the bits to be transmitted, s are the frequency domain data symbols, x are the time domain data samples, y is the received signal in the time domain, $\tilde{s}$ is the received signal in the frequency domain and $\hat{b}$ are the estimated bits.

The OFDM signal can be expressed in the time domain by [1]

$$x[n] = \sum_{k=0}^{K-1} s_k \; e^{j2\pi\frac{k}{K}n}, \tag{1}$$

where s_k is the data symbol on the k-th subcarrier and K is the number of subcarriers in the OFDM symbol.

The signal at the receiver can be written by

$$y[n] = \sum_{k=0}^{K-1} s_k \, H_k \; e^{j2\pi\frac{k}{K}n} + \omega_n, \tag{2}$$

where *y[n]* are time-domain sample before DFT transformation, H_k is the channel's frequency response at the *k*th frequency and ω_n is additive white Gaussian noise (AWGN).

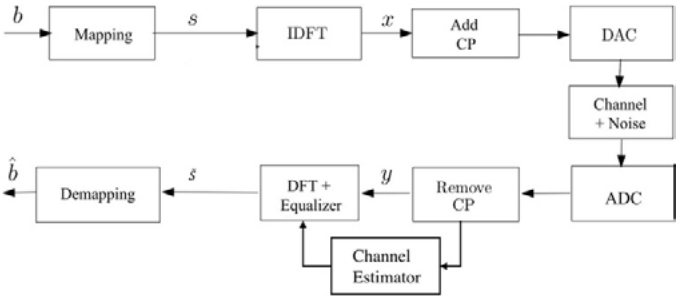


Fig. 1 Block diagram of the implemented OFDM system.

III. Channel Estimation and Data Detection

A. Channel Estimators

Channel estimation can be performed either in the time domain or in the frequency domain. In OFDM systems, pilot symbols s_p are typically inserted between data symbols for the purpose of channel estimation. These pilot symbols allow for the initial estimation of the channel's frequency response across a subset κ_p of subcarriers, known as pilot positions, with a cardinality $|\kappa_p|$. Then, the channel's frequency response is interpolated across the remaining *K*-$|\kappa_p|$ subcarriers. Hence, the OFDM system can be expressed as

$$y[n] = \sum_{k\in\kappa_p} s_p(k)H_p(k)\, e^{j2\pi\frac{k}{K}n} + e_n, \quad (3)$$

where, $y[n]$ represents the received signal in the time domain; $s_p(k)$ denotes the pilot symbol transmitted on the *k*-th subcarrier; $H_p(k)$ represents the frequency response of the channel at the *k*-th subcarrier and

$$e_n = \sum_{k\notin\kappa_p} s(k)H(k)\, e^{j2\pi\frac{k}{K}n} + \omega_n \quad (4)$$

encompasses the residual noise and the interference from data symbols on non-pilot subcarrier. Here, these unknown symbols carrying information will be considered as noise during the training phases. It is well known that for a channel impulse response with a maximum of *L* resolvable paths (and, consequently, degrees of freedom), then $|\kappa_p| \geq L$ [22].

In this study, we will evaluate the performance of various channel estimation techniques. The estimators selected for comparison have been carefully chosen based on their relevance and effectiveness in the context of our investigation:

1) LS:

The least squares (LS) channel estimator yields the estimation of the channel's frequency response at the pilot tone positions as referenced in [23]

$$\hat{H}(k) = \frac{\hat{s}_p(k)}{s_p(k)}, \quad (5)$$

where $\hat{s}_p$(k) represents the received signal on the *k*-th subcarrier in the frequency domain, while s_p(k) denotes the pilot signal transmitted on the *k*-th subcarrier. Following the estimation process, a linear interpolation technique is employed to derive the channel's frequency response across all subcarriers within the OFDM symbol.

2) SVR:

SVR is an extension of the widely-known Support Vector Machine (SVM) technique, initially proposed by Drucker et al. [24]. While SVM aims to find an optimal hyperplane for classification tasks, SVR seeks an optimal hyperplane with an ϵ-tube around it to accommodate a continuous output variable. This ϵ-tube ensures that most data points fall within its boundaries. Like SVM, SVR employs the kernel trick to map the input data into a higher-dimensional feature space, facilitating linear regression analysis.

Considering that SVR was originally designed to operate on real-valued samples [25], we adapt our methodology to handle OFDM symbols, which are typically complex-valued. Our proposed SVR estimator is divided into two parallel estimators, each focusing on either the real part $\Re(y[n])$ or the imaginary part $\Im(y[n])$ of the OFDM symbol. For simplicity and clarity, we detail only the estimation process for the real part below, as the development for the imaginary part is analogous.

The dual optimization problem for ϵ-kernel-SVR is given by (Refs. [26, [27]):

$$\max_{\alpha,\alpha'} -\epsilon\sum_{i=1}^{l_{sv}}(\alpha_i - \alpha_i') + \sum_{i=1}^{l_{sv}}(\alpha_i' - \alpha_i)\Re(y[n]) - \frac{1}{2}\sum_{j=1}^{l_{sv}}\sum_{i=1}^{i_{sv}}(\alpha_i' - \alpha_i)(\alpha_j' - \alpha_j)\,K\left(s_p(i) - s_p(j)\right), \quad (6)$$

subject to the constraints:

$$0 \leq \alpha_i, \alpha_i' \leq C, \sum_{i=1}^{l_{sv}}\alpha_i'\,\alpha_i = 0, \quad (7)$$

where the kernel function is defined as:

$$K\left(s_p(i) - s_p(j)\right) = \langle\phi\left(s_p(\mathrm{i})\right), \phi\left(s_p(\mathrm{j})\right)\rangle \quad (8)$$

Here, *C* serves as a regularization parameter, while α and α' represent Lagrange multipliers. The parameter l_{sv} is the number of support vectors, s_p represents an individual datum, and ϕ denotes the transformation from the feature space to the kernel space.

SVR allows for the use of different kernel functions such as linear, polynomial, or radial basis function (RBF) kernels. The choice of kernel depends on the nature of the problem and the characteristics of the dataset. In this study, we employ a variety of kernels, including traditional ones such as the linear and RBF, alongside the innovative quantum kernel.

B. Data Detection

1) coherent demodulation

The data detection with coherent demodulation, normally, is made assuming symbol-by-symbol minimum distance detection. In this case, the detector can be expressed as

$$\hat{s}_k = arg\min_{\hat{s}_i} J(\tilde{s}_i), \forall_i \in \{0, 1, \ldots, \mathrm{M}-1\}, \quad (9)$$

where,

$$J(\tilde{s}_i) = |\tilde{s}_k - \tilde{s}_i|^2. \quad (10)$$

$\tilde{s}_k$ is the complex symbol at the output of the equalizer on the k-th subcarrier, $\tilde{s}_i$ is the i-th complex symbol of the demodulator constellation of order M, that is, for *16*-QAM modulation, $M=16$, and for QPSK modulation, $M=4$. Due to the need to know the reference signal $\tilde{s}_i$, this demodulation is called coherent demodulation [28].

2) SVC

Support Vector Classifier (SVC) is a useful technique for data classification and is considered to be the state-of-the-art tool for linear and nonlinear classification. It realizes classification tasks for two-class problem by using the separating hyperplane [29]. The hyperplane is found by estimating the maximum distance to the closest data points of the training set. These closest data points are named support vectors (SVs). Data points can be transformed into a high dimensional space (HDS) by using a nonlinear transformation if they are not linearly separable in the input space. HDS is called feature space. These nonlinear transformations are represented by using kernel functions. The data points in the feature space are divided by the optimal separating hyperplane estimated by using the maximum distance to the closest data points of the training set mentioned as above. Although SVM is used to solve two-class learning problems, there are many ways to solve multi-class classification problems with SVM, such as: One Against One (OAO) and One Against All (OAA).

In this work we apply two multiclass SVCs to detect the 16-QAM symbols, one in the real part and the other in the imaginary part, respectively, and an analogous procedure with the binary version to detect the QPSK symbols.

IV. Quantum-Based Approaches for Channel Estimation and Data Detection

A. Quantum Kernels

Quantum kernels represent a fundamental concept in quantum machine learning (QML), leveraging the principles of quantum computing to process and analyze data. It encapsulates the essence of classical kernels within a quantum framework, enabling the exploration of complex data structures and relationships in higher-dimensional quantum feature spaces [30].

Quantum kernels utilize quantum feature maps to implement the kernel trick. This transformation is carried out by a quantum feature map $\emptyset : \mathcal{X} \to \mathcal{H}$ that maps a data point x to a corresponding quantum state in a Hilbert space. The entries of the quantum kernel $K(x_i, x_j)$ represent the fidelities or transition amplitudes between the states $|\emptyset(x_i)\rangle$ and $|\emptyset(x_j)\rangle$, which correspond to the transformed feature vectors x_i and $|x_j$, respectively [31]. For two quantum states $|\emptyset(x_i)\rangle$ and $|\emptyset(x_j)\rangle$, the kernel is defined as $K(x_i, x_j) = |\langle\emptyset(x_i)|\emptyset(x_j)\rangle|^2$, where $|\emptyset(x_i)\rangle = \mathcal{U}_{\emptyset(x_i)}|0\rangle$ and the circuit $\mathcal{U}_{\emptyset(x_i)}$ encodes the classical data x_i into the quantum state $|\emptyset(x_i)\rangle$ using a unitary operator $\mathcal{U}$.

On a quantum computer, the kernel circuit is set up for every conceivable pair of training samples, and the probability of measuring an all-zero string in the Z-basis serves as an estimate of the fidelity of their respective encoded quantum states. For predicting a new data point x, it suffices to estimate the kernel using all the l_{sv} support vectors.

B. Pauli Feature Maps

In traditional ML, feature maps are essential for converting raw input data into a higher-dimensional feature space, aiding in uncovering meaningful patterns and relationships. Similarly, quantum feature maps play a key role in QML, transforming classical data into quantum states suitable for quantum computers.

Quantum feature maps operate by leveraging quantum gate operations to transform input data into a new quantum state vector [30]-[31]. Notably, the Pauli Feature Map, first introduced by V. Havlíček et al. [18], employs Pauli gate operations to efficiently encode classical data into quantum states. This map enables complex transformations by converting input data with n features $x \in \mathbb{R}^n$ into quantum information in n qubits $|\psi(x)\rangle$ using a unitary operator:

$$\mathcal{U}_{\Phi(x)} = \prod_d U_{\Phi(x)} H^{\otimes n}, \tag{11}$$

where

$$U_{\Phi(x)} = \exp\left(i \sum_{S \in \mathfrak{T}} \phi_S(x) \prod_{k \in S} P_k\right), \tag{12}$$

S is a set of qubit indices that describes the connections in the feature map, $\mathfrak{T}$ is a set containing all these index sets and $P_k \in \{\mathbb{I}, X, Y, Z\}$ represents the Pauli matrices. The encoding function is given by

$$\phi_S: \mathbf{x} \mapsto \begin{cases} x_i & if\ S = \{i\} \\ (\pi - x_i)(\pi - x_j) & if\ S = \{i, j\} \end{cases} \tag{13}$$

The Pauli Feature Map enables the representation of higher-order correlations between original data points, allowing for the capture of complex relationships that may not be easily discernible classically. This capability is particularly valuable in tasks such as classification, regression, and clustering, where capturing intricate data dependencies is crucial for achieving high predictive performance.

In our study, we introduce a 5-qubit QSVR algorithm applied to channel estimation in OFDM systems. This QSVR model is designed to handle the intricacies of channel estimation within OFDM systems, where the number of pilot tones, crucial for accurate estimation, corresponds to number of qubits utilized in our proposed model. In the second part of our work we use a two-qubit QSVC to detect the data replacing the coherent demodulator.

To configure our QSVR and QSVC model effectively, we carefully selected values for essential parameters, including the Pauli sequence, the number of repetitions, and the type of entanglement. These parameters serve as arguments within the *PauliFeatureMap* class, which is implemented in the Qiskit Python package [32]. For our specific implementation, we opted for a Pauli sequence comprising the Z and ZZ operators, with no repetition ($d=1$) and linear entanglement. This configuration was chosen based on its compatibility with the characteristics of OFDM systems and its potential to yield accurate channel estimation and data detection results. For 2 features (2 qubits), the Pauli expansion matrix of ZZ feature map can be written as:

$$U_{\emptyset(x)} = \exp\big(i(x_1 Z_1 + x_2 Z_2 + (\pi - x_1)(\pi - x_2) Z_1 Z_2)\big). \quad (14)$$

The first two terms are equivalent with R_Z rotation gates on each individual qubit. Specifically, $\exp(i x_1 Z_1) = R_Z(2x_1)$ and $\exp(i x_2 Z_2) = R_Z(2x_2)$. Furthermore, the tensor product $\exp(\mathrm{i}(\pi - x_1)(\pi - x_2) Z_1 Z_2)$ is equivalent to entangled gates: $CX.(I \otimes R_Z(2(\pi - x_1)(\pi - x_2))).CX$.

V. Results

To validate the proposed quantum kernel for the SVR and SVC, and compare its performance with the classic SVR and the LS in channel estimation, and coherent demodulation and classic SVC for symbol detection in OFDM systems, bit error rate (BER) and mean square error (MSE), i.e. $MSE = E[|H - \hat{H}|^2]$, curves were created, considering the following simulation parameters:

TABELA I
SIMULATION PARAMETERS

number of subcarriers [K]	16
subcarrier modulation	QPSK 16-QAM
cyclic prefix length in number of subcarriers	4 and zero
number of pilot subcarriers [S_p]	5

The tests were performed on a frequency selective channel with a delay profile given by $h = [1 \quad 0 \quad 0.3 + 0.3j]$.

We consider a packet-based transmission, where each packet consists of a header at the beginning of the packet with a known training sequence or preamble to carry out channel estimation, followed by the OFDM data symbols. At the preamble, there are two OFDM symbol with pilot subcarriers. After the estimation (with either QSVR, SVR or LS) of channel coefficients at pilot positions $\hat{H}_p(k)$, we use them to compute the interpolation of the channel. Next, we perform zero forcing (ZF) equalization [33] using the interpolated channel. Detection is carried out with a hard-decision slicer over the equalized data in the first teste and, SVM and quantum kernel SVM classifier (QSVM) in the second test. For each estimator, 100 packets were transmitted to calculate the average and raising the BER.

We studied the performance variation in the system due to changes in the kernel and free parameters of the SVR and SVC. We tested the linear, radial and polynomial kernels and varied the C parameter from 1 to 1000. The optimal parameter found for C was $C = 100$, and RBF kernel. We also utilized the Qiskit Python package [32] for the quantum tasks with a local quantum simulator. The classical kernel-based method for SVR and SVC was run on a classical computer with a regular CPU. Figs. 2 and 3 show, respectively, the MSE and BER performance as a function of the signal-to-noise ratio (E_b/N_0) for the first test with 16-QAM.

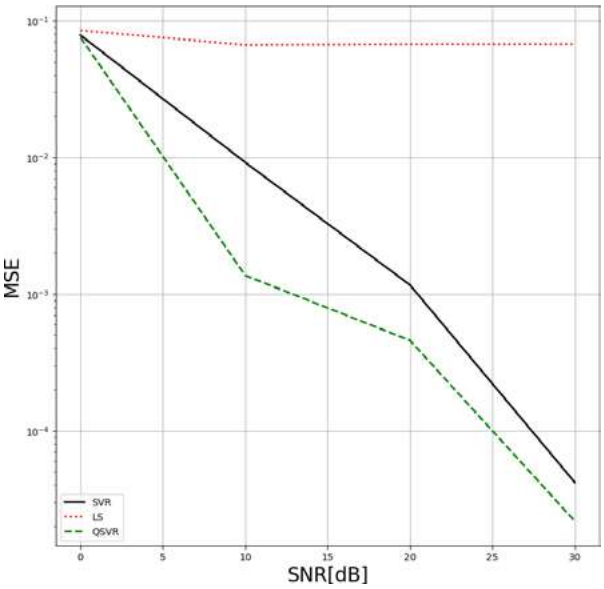


Fig. 2. MSE performance comparison.

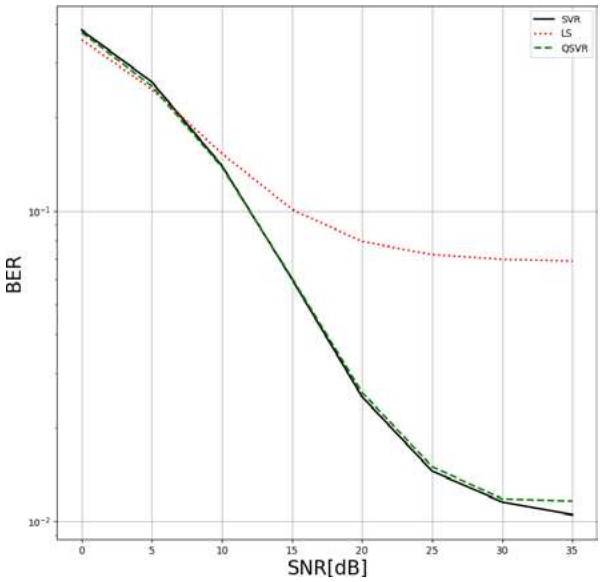


Fig. 3. BER performance comparison.

Analyzing the MSE and BER performance obtained from the three tested estimators, we observe in Fig. 2 that the MSE curve obtained with the LS estimator does not decay due to ISI caused by the lack of the cyclic prefix. The BER curve, Fig. 3, also decays very slowly up to 20 dB of SNR and presents a plateau from this value onwards, indicating its sensitivity to ISI. In contrast, the curves obtained with the SVR and QSVR estimators demonstrate robustness to ISI, with slightly better performance for QSVR in terms of MSE. This can be attributed to the better suitability of the quantum kernel for linearizing the input space data.

The plateau trend observed in the BER curves for the SVR and QSVR from 30 dB of SNR can be explained by the ISI generated by the lack of CP, impacting coherent detection while leaving channel estimation unaffected. To mitigate ISI and enhance robustness, we explored replacing the coherent detector with an SVM and QSVM classifier. Despite observing a reduction in BER with tests using a multiclass SVM classifier

the plateau trend persisted. The inability to implement a multiclass QSVM classifier led us to transition to QPSK modulation, enabling testing with a binary QSVM classifier. Figs. 4 and 5 illustrate the BER performance versus signal- to-noise ratio (E_b/N_0) for the first QPSK test and the second test solely with the classic SVC classifier, respectively. We can see in Fig. 4 that the switch to QPSK, maintaining coherent detection, was not enough to eliminate the plateau after 30 dB of SNR. However, in Fig. 5, it is possible to observe that the SVC is robust to ISI caused by the absence of the CP, independent of the regressor type used in channel estimation (LS, SVR or QSVR).

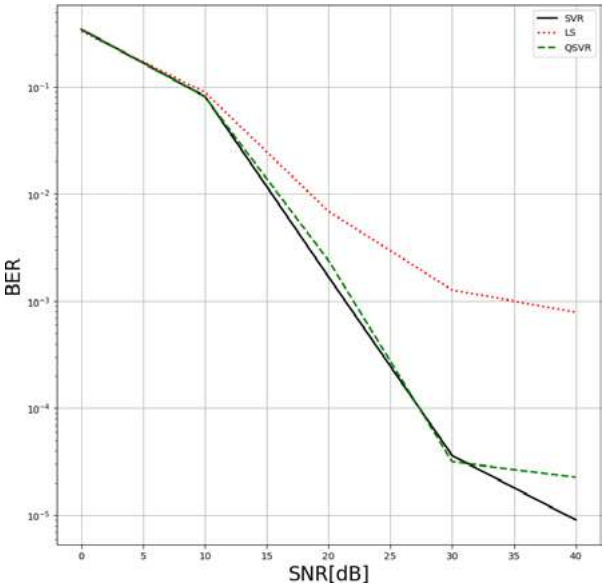


Fig. 4. BER performance comparison for the first test with QPSK modulation.

Several experiments are being carried out with the aim of adjusting the QSVC parameters, including increasing the number of steps performed during the training process (τ), varying the number of samples for training and testing, and varying the parameter C. The best performance obtained so far was with 10000 samples, 6000 for training and 4000 for testing, τ =1000 and C = 1000, using "*ZfeatureMap*" coding. However, the performance of QSVC is still slightly worse than that of classic SVC, as can be seen in Fig. 6, requiring further investigation. Additionally, the development of effective multiclass quantum classifiers is needed, as current results indicate inferior performance compared to classical multiclass SVMs.

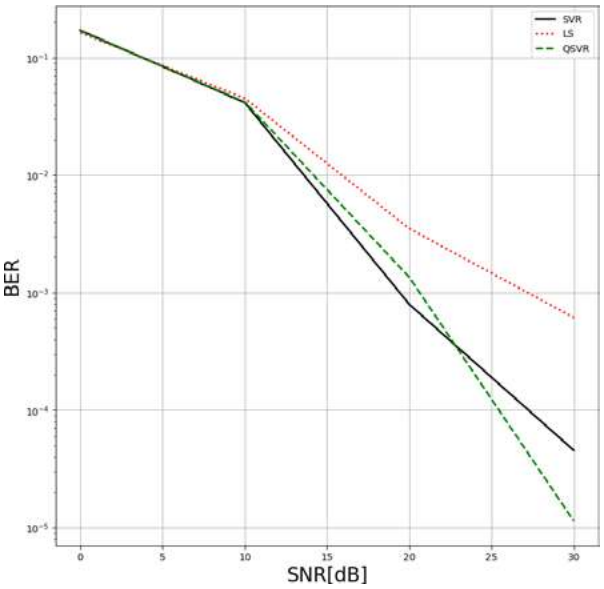


Fig. 5. BER performance comparison for the second test with QPSK modulation and SVC classifier.

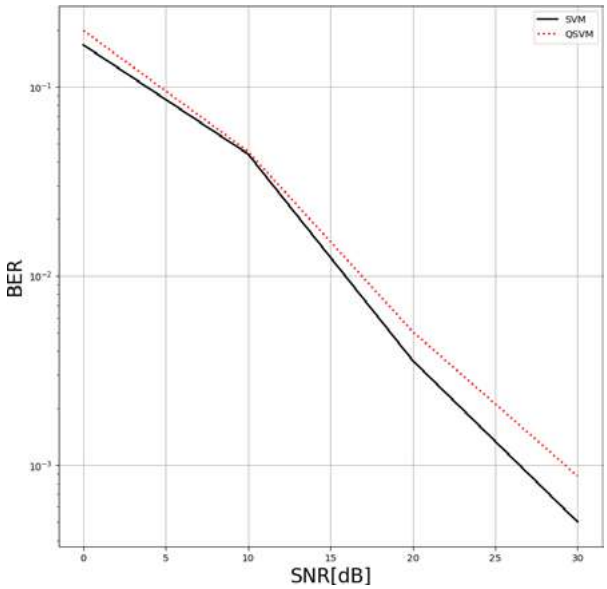


Fig. 6. BER performance comparison for the second test with QPSK modulation and QSVC classifier.

VI. CONCLUSIONS

In this work, an SVR and an SVC algorithms with a quantum kernel for channel estimation and data detection in OFDM systems were proposed. Therefore, the structure of the adopted OFDM system, the channel estimation process in the time domain, by the SVR, the frequency domain, by the LS, the data detection model and the SVC, in addition to the fundamentals of quantum computing for generating the quantum kernel were described. Several tests were carried out in search of the optimal parameters of the SVR, SVC and the OFDM system. The simulations confirmed the robustness of the QSVR in the presence of ISI and the results show that the proposal outperforms the classic SVR and the LS for channel estimation. Following this work, we intend to investigate the parameters and adjustments to improve the performance of multiclass QSVC in data detection in OFDM systems.

REFERENCES

[1] N. Marchetti, M. I. Rahman, S. Kumar, R. Prasad, *New Directions in Wireless Communications Research.* cap. 2, OFDM-Principles and Challenges, 2009.

[2] A. F. Molisch. *Orthogonal Frequency Division Multiplexing (OFDM).* in Wireless Communications , IEEE, 2011.

[3] B. R. Ballal, A. Chadha, N. Satam. *Orthogonal Frequency Division Multiplexing and its Applications.* International Journal of Science and Research (IJSR), 2319-7064 Volume 2 Issue 1, 2013.

[4] B. Wang, K. J. R. Liu. *Advances in cognitive radio networks: A survey.* in IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 1, pp. 5-23, Feb. 2011.

[5] Z. Du, X. Song, J. Cheng and N. C. Beaulieu. *A channel estimation technique for OFDM systems in dispersive time-varying channels.* 11th Canadian Workshop on Information Theory, Ottawa, ON, Canada, 2009.

[6] D. Shrestha, X. Mestre and M. Payaró, *On channel estimation for power line communication systems in the presence of impulsive noise*, Computers & Electrical Engineering, Volume 72, Pages 406-419, 2018.

[7] M. P. Sánchez-Fernández, M. de Prado-Cumplido, J. Arenas-García, and F. Pérez-Cruz, *SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems.* IEEE Trans. Signal Process., vol. 52, no. 8, pp. 2298–2307, 2004.

[8] D. Sebald and A. Buclew, *Support vector machine techniques for nonlinear equalization.* IEEE Trans. Signal Process., vol. 48, no. 11, pp.3217–3226, Nov. 2000.

[9] L. V. Nguyen, D. H. N. Nguyen, and A. L. Swindlehurst, *SVM-based channel estimation and data detection for massive MIMO systems with one-bit ADCs.* to appear at IEEE Int. Conf. Commun., 2020.

[10] A. W. Harrow, A. Hassidim, and S. Lloyd. *Quantum algorithm for linear systems of equations.* Physical Review Letters, vol. 103, no. 15, oct 2009.

[11] Schuld, M.; Petruccione, F. *Supervised learning with quantum computers*. Cham: Springer, 2018.

[12] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash. *q-means: A quantum algorithm for unsupervised machine learning.* arXiv:1812.03584, 2018.

[13] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum algorithms for supervised and unsupervised machine learning arXiv:1307.0411, 2013.

[14] J. Preskill. *Quantum computing in the NISQ era and beyond.* Quantum, vol. 2, p. 79, aug 2018.

[15] P. Botsinis, S. X. Ng and L. Hanzo. *Quantum Search Algorithms, Quantum Wireless, and a Low-Complexity Maximum Likelihood Iterative Quantum Multi-User Detector Design* in IEEE Access, vol. 1, pp. 94- 122, 2013, doi: 10.1109/ACCESS.2013.2259536.

[16] P. Botsinis, D. Alanis, Z. Babar, S. X. Ng and L. Hanzo. *Joint Quantum-Assisted Channel Estimation and Data Detection* in IEEE Access, vol. 4, pp. 7658-7681, 2016, doi: 10.1109/ACCESS.2016.2591903.

[17] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary and M. Asaduzzaman. *Quantum Machine Learning for 6G Communication Networks: State-of-the-Art and Vision for the Future* in IEEE Access, vol. 7, pp. 46317-46350, 2019.

[18] V. Havlíc˘ek, A. D. Córcoles, K. Temme, Et al. *Supervised learning with quantum-enhanced feature spaces*. Nature 567, 209–212 (2019).

[19] J. E. Park, B. Quanz, S. Wood, H. Higgins, and R. Harishankar. *Practical application improvement to quantum svm: theory to practice.* 2020. [Online]. Available: https://arxiv.org/abs/2012.07725

[20] A. Viladomat Jasso, A. Modi, R. Ferrara, C. Deppe, J. Nötzel, F. Fung, M. Schädler. *Quantum and quantum-inspired stereographic k nearest-neighbour clustering.* Entropy, vol. 25, no. 9, 2023.

[21] M. Schuld and N. Killoran. *Is quantum advantage the right goal for quantum machine learning?* PRX Quantum, vol. 3, p. 030101, Jul 2022.

[22] M. J. Fernández-Getino García, J. M. Páez-Borrallo, and S. Zazo, *DFT-based channel estimation in 2D-pilot-symbol-aided OFDM wireless systems.* In: Proc IEEE Vehicular Technology Conf., 2001, vol. 2, pp. 815–819.

[23] A. Papoulis. *Probability Random Variables and Stochastic Processes.* McGraw-Hill, NY, 3 edition, 1991.

[24] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. *Support vector regression machines.* In: Advances in Neural Information Processing Systems, M. Mozer, M. Jordan, and T. Petsche, Eds., vol. 9. MIT Press, 1996.

[25] Smola, A.J., Schölkopf, B. *A tutorial on support vector regression*. Statistics and Computing 14, 199–222 (2004).

[26] V. Vapnik. *The Support Vector Method of Function Estimation.* Boston, MA: Springer US, 1998, pp. 55–85.

[27] R. Khanna and M. Awad. *Efficient Learning Machines: Theories, Con- cepts, and Applications for Engineers and System Designers.* Apress, 04 2015.

[28] Ali Grami. *Introduction to Digital Communications* Academic Press, Elsevier, 2016.

[29] Yao, Y., Frasconi, P., Pontil, M. *Fingerprint classification with combi- nations of support vector machines*. InAVBPA 2001, LNCS 2091 (pp. 253-258).

[30] M. Schuld. *Supervised quantum machine learning models are kernel methods* in arXiv preprint arXiv:2101.11020, 2021.

[31] M. Schuld, N. Killoran. *Quantum machine learning in feature hilbert spaces* in Physical review letters, vol. 122, 2019.

[32] A. Asfaw, L. Bello, Y. Ben-Haim, Et al. *Qiskit: An Open-Source Framework for Quantum Computing*. ArXiv, 2020.

[33] U. Katare, P. Patidar e A.C. Tiwari, *Comparative Analysis of ZF and MMSE Receiver for Multicode MC-CDMA Downlink Channels*. International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 3, Issue 4, Julho 2014.


CEFET/RJ