

# Kuantum Weyl Dönüşümlerin Cirq Üzerinde Uygunlaması

## *Implementation of Quantum Weyl Transformations on Cirq*

Osman Semi Ceylan 

Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Mühendisliği, Çanakkale, Türkiye, e-mail: [osman.semi.ceylan@gmail.com](mailto:osman.semi.ceylan@gmail.com)

Öz

Kuantum hesaplama, kuantum algoritmaların çözüm kümesini bilinmeyen bir kuantum durumun evrimsel sürecinde kubitlere kodlanması prensibine dayanır. Bir başlangıç kuantum durumun çözüm durumuna geçmesi için bir dizi temel kuantum kapının uygulanması gerekmektedir ve matematiksel matrisler ile soyutlanırlar. Temel kuantum kapılarının matrisleri Weyl dönüşümleri ile elde edilirler. Her ne kadar ikili kuantum sistemlerde bu dönüşümler simülasyonu yapılabilir de yüksek boyut kuantum sistemlerde simülatörler için kullanılabilen hazır durumda bir kütüphane veya benzeri bir kodlama bulunmamaktadır. Bu çalışma ile kuantum yüksek boyutta Weyl dönüşümlerinin matrisleri kodlanmış olup bu dönüşümler Cirq kuantum simülatörü için kapılar olarak kodlanmıştır. Böylelikle araştırmacılar ve alana meraklıları için Cirq simülatöründe çalışan kuantum kapılar kolayca erişilebilir olmaktadır.

**Anahtar kelimeler:** Kuantum Bilgi, Kuantum Yüksek Boyut

Abstract

Quantum computing is based on the principle of encoding the solution space of quantum algorithms into qubits in the evolutionary process of an unknown quantum state. In order for an initial quantum state to transition to a solution state, a series of basic quantum gates must be applied and are abstracted by mathematical matrices. The matrices of fundamental quantum gates are obtained by Weyl transformations. Although these transformations can be simulated in binary quantum systems, there is no ready-made library or similar coding that can be used for simulators in high-dimensional quantum systems. In this study, the matrices of quantum high-dimensional Weyl transformations were coded and these transformations were coded as gates for the Cirq quantum simulator. Thus, quantum gates running in the Cirq simulator are easily accessible for researchers and field enthusiasts.

**Keywords:** Quantum Information, Quantum High Dimension

**Citation/Atf:** CEYLAN, O. S. (2023). Kuantum Weyl Dönüşümlerin Cirq Üzerinde Uygunlaması. *Kuantum Teknolojileri ve Enformatik Araştırmaları*. 1(1): 1-5, DOI: 10.5281/zenodo.10102956

**Corresponding Author/ Sorumlu Yazar:**  
Osman Semi Ceylan  
E-mail: [osman.semi.ceylan@gmail.com](mailto:osman.semi.ceylan@gmail.com)



Bu çalışma, Creative Commons Atif 4.0 Uluslararası Lisansı ile lisanslanmıştır.  
This work is licensed under a Creative Commons Attribution 4.0 International License.

## 1. GİRİŞ

Kuantum parçacıklar kullanılarak hesaplama yapılabileceği fikri ilk olarak Richard Feynman tarafından ortaya atıldı[1]. Bu öncü çalışma kuantum bilgisayarların teorikte klasik bilişim teknolojilerinden farklı olarak süperpozisyon fenomeninden dolayı üstel veri işleme kapasitesine sahip olduğunu öne sürdü. Günümüze kadar ise nükleer manyetik rezonans[2], süperiletken[3], foton[4], iyon tuzaklama[5] tabanlı gibi farklı prensiplerde çalışan küçük ölçekli kuantum bilgisayarlar geliştirilmiştir. Günümüzde geliştirilen genel amaçlı kuantum bilgisayarlar kapı tabanlı kuantum bilgisayarlar olarak bilinmektedir.

Kapı tabanlı kuantum bilgisayarlar başlangıçta temel durumda olan bir kuantumun kubitlerinde çözüm kümesi elde edilene kadar iteratifce bir seri dönüşümün uygulanması prensibi ile çalışmaktadır. Kuantum durumlar arasındaki evrimsel bu geçişi evrensel kuantum kapılar[6] sağlamaktadır. Bu kapılar kuantum bilgi biliminde matrisler olarak ifade edilmektedirler. Bu kapılar Weyl dönüşümü[7] kullanarak elde edilebilir. Weyl dönüşümleri ile ancak karmaşık uzayda kuditlerin bir temsil biçimi olan keyfi bir boyutun Fock durumunda işlem yapabilir.

Weyl dönüşümleri ile elde edilen  $d$ -boyutlu matrisler kullanılarak yüksek boyutta kuditlerde kuantum algoritmalar çalıştırmak mümkündür. Fakat yüksek boyut kuantum simülatörleri ikili sistemler için halihazırda bu dönüşümleri entegre kullanılabilecek şekilde sunmakta iken yüksek boyutta ise kullanıcılardan gelen ek girdilere ihtiyacı vardır. Bizler bu çalışma ile keyfi bir  $d$ -boyutta Weyl dönüşümlerini Cirq kuantum devre simülatöründe[8] çalışabilecek biçime getirdik. Bunu gerçekleştirebilmek için önce Weyl dönüşüm matrislerini çıktısını verebilecek bir yordan yazdık. Sonra Cirq simülatörü üzerinde kapı olarak tanıtımını sağlayacak olan entegre sınıfı oluşturduk.

Bu makale 4 bölümde derlenmiştir. İlk bölümde çalışmaya giriş yapılmıştır. Bölüm 2 ise kullanılan yöntem anlatılmıştır. Bölüm 3 bir örnek üzerinden önerilen yöntem gösterilmiştir. Son bölümde bulgular ve sonraki çalışmalar tartışılmıştır.

## 2. YÖNTEM

Önerilen bu çalışmada öncelikle Weyl dönüşüm formülü bir Python yordamı yardımıyla kodu geliştirilmiştir.

$${}_{m\alpha}^d U_{pq} = \sum_{k=0}^{d-1} e^{\frac{2\pi\alpha i}{d^m} kp} |k\rangle \langle (k+q) \bmod d| \quad (1)$$

Denklem (1) ile  $d$ -boyutta Weyl dönüşümünün formülü verilmektedir. Burada  $p, q \in [0, 1, \dots, d-1]$  aralığında değerlerden oluşmaktadır. Denklemden görüleceği üzere bir  $d$ -boyutta  $d^2$  adet Weyl dönüşümü üretilebilmektedir.

Denklem (1)  $d$  boyutuna  $m$  köküne,  $\alpha$  dönme miktarına ve  $p$  ve  $q$  varyant parametreleri kullanarak bir dönüşüm matrisi oluşturmaktadır.

$${}_{11}^4 U_{00} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Denklem (2) ile  $d=4$  boyutta ve  $p, q$  ise 0 değerine eşit olduğunda elde edilen matris gösterilmektedir. Bu matris ikili sistemlerde evrensel kapılardan birisi olan birim ( $I$ ) kapısının yüksek boyuttaki matrisidir.

$${}_{11}^4 U_{01} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

Denklem (3) ile verilen matris ikili sistemlerde karşılığı evrensel  $X$  kapısıdır. Kuantum yüksek boyutta bilgi işlemede  $X$  kapısının  $d-1$  adet karşılığı bulunmaktadır. Bu denklemde  $q$  değerini keyfi bir değişken aldığında matrisin bir kudite etkisi

$${}_{11}^d U_{0q} |0\rangle = |q\rangle \quad (4)$$

biçiminde olmaktadır.

$$Y_p = i^{(p \bmod d)} {}_{11}^d U_{pp} \quad (5)$$

Denklem (5) ile verilen yüksek boyutta  $Y$  kapısı ise Weyl dönüşümünün üzerine karmaşık bir çarpan gerekmektedir.

$${}_{m\alpha}^d Z_q = {}_{m\alpha}^d U_{qq} \quad (6)$$

Denklem (6) ile verilen yüksek boyutta  $Z$  kapısı  $\alpha$  dönüşüm miktarı ve  $m$  kök değeri parametrelerini kullanmaktadır. Kök değeri  $m$  ile iki seviyeli kuantum sistemde karşılaşılan kapının kök

kuvvetleri,  $\alpha$  ile de faz dönüşüm miktarı elde edilebilir.

Denklemi (1)'i formüle ettiğimiz yordam kod ile kuantum yüksek boyut algoritmalarında kullanılan evrensel kapıların matrisleri kolayca hesaplanabilir. Fakat bu dönüşüm matrisleri ile yüksek boyutlu algoritma simüle etmek için yeterli değildirler. Bu nedenle kuantum yüksek boyutlu simülasyon yapabilme kapasitesi olan Cirq simülatörüne bu matrislerin sistematik kullanılmasına izin veren entegre bir sınıf geliřtirdik.

Bu entegre oluşturulan matrisleri kullanarak Cirq üzerinde kullanılacak kapılara çevirmektedir. Bunu yapabilmek için dönüşüm matrisi, çalışılan boyut ve devre diyagram bilgisini bulmak yeterlidir.

```
class Gate(cirq.Gate):
    def __init__(self,
                 unitary_matrix: typing.Iterable,
                 dimension: int,
                 diag_info: typing.Union[cirq.CircuitDiagramInfo,
                                         typing.Tuple[str, str, None]] = None):
```

Miras alınan bu sınıf kullanılabilmesi için tanımlanması gereken sınıf yordamları bulunmaktadır.

```
def _qid_shape_(self) -> typing.Tuple[int]:
    return (self.dimension, ) * self.qubit_count
```

Verilen yoram `_qid_shape_` ile boyut ve kudit sayısına bağlı bir set elde eder. Bu set Cirq simülatörünün kapıları devreye yerleştirme aşamasında önem teşkil eder.

```
def _num_qubits_(self) -> int:
    return self.qubit_count

def _unitary_(self) -> numpy.ndarray:
    return self.unitary_matrix
```

Verilen yordamlar ve `Gate` ile Cirq kuantum simülatörünün çalışma zamanında kullanılan kapının işlenmesi sırasında kullanılmaktadır.

Verilen yordam `_circuit_diagram_info_` ile Cirq'de programlanan tüm devrelerinin görselleştirilmesi gerektiğinde kapıların etiketleri, konumları ve etkileşimlerini ifade eder.

### 3. Kuantum Fourier Dönüşümü Uygulaması

Bu çalışmada önerilen entegre kod ile kuantum yüksek boyutta kolayca simülasyonlar gerçekleştirilebilir. Bu bölümde uygulamayı göstermek adına kuantum Fourier dönüşümü algoritması kullanarak gerçekleştirilmiş devre simülasyonu bulunmaktadır.

Kuantum Fourier dönüşümü(QFT) iki seviyeli sistemlerde de olduğu üzere *Hadamard* ve *kontrollü-Z* kapıları kullanılarak gerçekleştirilmektedir. Yüksek boyutta kuantum hesaplamada da bu kapıların denklileri kullanılarak gerçekleştirilebilir. Bunu gerçekleştirebilmek için gerekli matrisler elde edildikten sonra entegre kapı sınıfına tanıtmak ve sonrasında kuantum devreye yerleştirmek yeterli olacaktır.

```
dimension = 4
H_matrix = general_Hadamard(dim=dimension)
Z_matrix = general_Z(1, root=1, dim=dimension)
S_matrix = general_Z(1, root=2, dim=dimension)
T_matrix = general_Z(1, root=3, dim=dimension)
X_matrix = general_X(1, dimension)
swap_matrix = swap(dimension)
```

Verilen bu QFT örneği  $d=4$  boyutta hazırlanan *Hadamard*, *Z*, *S*, *T*, *X* ve *Swap* matrisleri entegre yordamlar yardımıyla hazırlanırlar. Burada kullanılan *X* temel durumdaki bir kudit birinci seviye Fock bazına çıkaran varyantıdır ve QFT'de kayda değer bir değişikliğin görülmesi için eklenir. Çünkü kuantum devre temel durumda başlar ise QFT sadece süperpozisyona dönüşür.

```
H_gate = Gate(H_matrix, dimension, f'H_{{dimension}}')
Z_gate = Gate(Z_matrix, dimension, f'Z_{{dimension}}')
S_gate = Gate(S_matrix, dimension, f'S_{{dimension}}')
T_gate = Gate(T_matrix, dimension, f'T_{{dimension}}')
X_gate = Gate(X_matrix, dimension, f'X_{{dimension}}')
swap_gate = Gate(swap_matrix, dimension, ('x', 'x'))
CZ_gate = Z_gate.controlled(1, (1, ), (dimension, ))
CS_gate = S_gate.controlled(1, (1, ), (dimension, ))
CT_gate = T_gate.controlled(1, (1, ), (dimension, ))
```

QFT için gereken kapılar elde edilen matrisler kullanılarak hazırlanırlar. Verilen kontrollü kapılar ise Cirq içerisinde `.controlled` hazırda bulunan kullanılarak hazırlanırlar. Fakat yüksek boyut kuantum hesaplamada iki seviyeli sistemlerden farklı olarak sadece birinci seviye Fock bazı kontrol etmez. Multi-value Controlled Gate(MVCG) olarak adlandırılan bu kontrollü kapıların kontrol değerleri kümesi kullanarak olu-

şan bir alt kümedir.

```

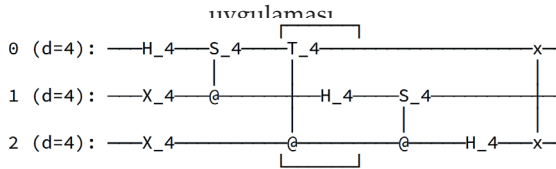
qudits = circ.LineQid.range(3, dimension=dimension)
circuit = circ.Circuit()

circuit.append(X_gate.on(qudits[1]))
circuit.append(X_gate.on(qudits[2]))
circuit.append(H_gate.on(qudits[0]))
circuit.append(CS_gate.on(qudits[1], qudits[0]))
circuit.append(CT_gate.on(qudits[2], qudits[0]))
circuit.append(H_gate.on(qudits[1]))
circuit.append(CS_gate.on(qudits[2], qudits[1]))
circuit.append(H_gate.on(qudits[2]))
circuit.append(swap_gate.on(qudits[0], qudits[2]))
print(circuit)

```

4 boyutta 3 adet kudit içeren bir Cirq devresi oluşturulduktan sonra öncesinde hazırlanmış kapılar gerekli kubitlere etki ettirilmelirler. Verilen kod parçasında görüleceği üzere X kapıları ile devrenin başlangıç durumu değiştirilmiştir. *Hadamard* ile süperpozisyona evrilen kuditler sonrasında, *CS*, *CT* kontrollü kapıları kullanmıştır. QFT devresi kubitlerin yerlerinin değiştirilmesi ile *.swap* sonlanmaktadır. Kullanılan kapısı kudit sıralamayı tersten hazırlayan devre simülatörleri için gerekli olmaktadır. Böylelikle hazırlanan Weyl dönüşümü yordamları ve Cirq kapı entegrasyonu sayesinde yüksek boyutta bir kuantum algoritma kuantum devreye eklenmiş ve simülasyona hazır durumda olur.

Şekil-1. 3 kuditlik QFT algoritmasının Cirq devre



Şekil-1 ile kuantum yüksek boyutta hazırlanan 3 kuditlik QFT algoritması örneğinin devre görseli sunulmaktadır. Şekilden görüleceği üzere daha öncesinde hazırlanan entegre kodlar sayesinde QFT için gereken kapılar ilgili konumlarındadır.

Cirq üzerinde  $d=4$  boyutta hazırlanan bu devre simülase edildiğinde elde edilen kuantum durum ile QFT matrisinin başlangıçtaki kuantum durum vektörüyle çarpımı karşılaştırıldığında tüm elemanlarının bire bir eşit olduğu saptanmıştır. Böylelikle oluşan Weyl dönüşüm matrislerinin ve onlardan oluşturulan evrensel kuantum kapıların doğruluğu sağlanmaktadır.

## 4. SONUÇ VE GELECEK ÇALIŞMALAR

Bu çalışma ile önerilen Weyl dönüşüm matrislerinin ve onların Cirq kütüphanesi üzerinde entegrasyonu sayesinde araştırmacılar ve alana meraklıları için hazırda bulunan yöntemler sunulmaktadır. Bu çalışma ile kuantum algoritmaların kodlanmasına ve kolayca kullanılmasına yardımcı olacağı düşünülmektedir. Özellikle kuantum yüksek boyutun iki seviyelere göre gürültü duyarlılığı, az kudit üzerinde daha geniş veri kodlama ve işleme hacmi ve devre karmaşıklığının daha az olması gibi çeşitli avantajları[9] ve alana olan artan ilgi göz önüne alındığında önerilen çalışmanın kuantum hesaplama alanına fayda sağlayacağı aşikardır.

Bizler bu çalışmamızı taban alarak daha fazla yüksek boyut kuantum kapı matrisleri oluşturabilecek yordamlar üzerinde çalışmalarda bulunacağız. Bunun yanı sıra belirtilen yordamları diğer yüksek boyut simülatörlerine entegre ederek kullanıcıları matematiksel matris altyapısından soyutlayarak geliştirdikleri yüksek boyut kuantum algoritmalara odaklanmasını sağlamayı hedefliyoruz.

## KAYNAKÇA

- [1] Feynman, R. P. *Simulating physics with computers*. CRC Press, 2018, pp. 133 – 153.
- [2] Vandersypen, L., Matthias S., Breyta, G., Contantino, Y., Mark, S., Isaak, C. *Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance*. Nature, 2001.
- [3] Ballon, A. *Quantum computing with superconducting qubits – PennyLane*. PennyLane Demos. Retrieved 2022-12-13.
- [4] Svetlana, K., Felix, P., Karoline, S., Simone, F., Oliver, K., Vadim, K., Patrik, R., Andreas, V., Frank, H. Manfred, M. K. *Fully integrated quantum photonic circuit with an electrically driven light source*. Nature Photonics, 2016, pp. 727-732.
- [5] Juan, I. C., Peter, Z. *Quantum computations with cold trapped ions*. Physical Review Letters, 1995.
- [6] Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., Weinfurter, H. *Elementary gates for qu-*

- antum computation*. Physical Review A. American Physical Society (APS), pp. 3457–3467.
- [7] Berkani, M. *On the equivalence of Weyl thorem and generalized Weyl theorem*. Acta Mathematica Sinica, 2007, pp. 103-110.
- [8] Cirq Developers. *Cirq*. Zenodo. Doi: 10.5281/zenodo.8161252.
- [9] Daniele, C., Beatrice, D. L., Davide, B., Leif, K. O. *High-dimensonal quantum communication: benefits, progress, and future challanges*. Advanced Quantum Technologies, 2019.