A short introduction to unitary 2-designs

Olivia Di Matteo, CS 867/QIC 890, 4 November 2014

1 Everything is a design

Well, maybe not quite *everything*. But everything that I care about is a design. The most comprehensive definition of a design is that it is a finite set which is balanced, in the sense that it satisfies some symmetries. Designs have existed in the mathematical world for hundreds of years. Many mathematical and physical structures, such as Latin squares, affine and projective planes, mutually unbiased bases, error correcting codes, and so much more, can be ultimately be shown to be some type of design. In most of these cases, the associated designs are *combinatorial* designs. The theory and construction of combinatorial designs is a very rich, well-developed field, and would take an entire course (or a very large tome [1]) to even begin to do it justice.

Recently, new types of designs which are quantum in nature have emerged: quantum state designs, and unitary designs. The motivation for unitary designs is that they provide an answer to a very pertinent question: "How can we efficiently sample a random matrix from the unitary group?". The goal of this lecture and notes is to provide an introduction to unitary designs, and show that the familiar Clifford group is an exact unitary 2-design. In addition, we will investigate and implement a procedure which efficiently approximates unitary 2-designs by randomly generating unitary operations.

2 Preliminaries

2.1 Notation

- Let $\mathcal{U}(d)$ be the set of all $d \times d$ unitary matrices. We will work almost uniformly with n-qubit systems, so consider $d = 2^n$ unless otherwise specified.
- Let $\mathcal{S}(\mathbb{R}^d)$ be the unit sphere in $\mathbb{R}^d$ (often noted in other works as $\mathcal{S}^{d-1}$), and $\mathcal{S}(\mathbb{C}^d)$ the unit sphere in $\mathbb{C}^d$.
- We denote the Clifford group over n qubits as $\mathcal{C}_n$.
- Similarly, the Pauli group over n qubits is $\mathcal{P}_n$.
- Differentiating operators and superoperators can be tricky, so I define and stick to a consistent notation. $\hat{U}$ is a superoperator with some action $\hat{U}\rho = U(\rho)$; U is simply a matrix (which is an operator whose action is simply matrix multiplication). I will use $\hat{U}\rho$ and $U(\rho)$ somewhat interchangeably - mostly I will use the latter only when things may be ambiguous.
- I will assume the reader is familiar with the concept of measure. I will use μ to represent the normalized spherical measure, and η the normalized, unitarily invariant Haar measure.

2.2 Twirling quantum channels

Most of the operations performed in the proofs we will work through are twirling operations.

Let $\hat{\Lambda}$ be a quantum channel. Suppose we conjugate our channel by a unitary operation, $\hat{U}$. This sends the channel $\hat{\Lambda}$ to

$$\hat{\Lambda} \mapsto \hat{U}\hat{\Lambda}\hat{U}^{\dagger}, \tag{1}$$

where the action of $\hat{U}$ is

$$\hat{U}\rho = U\rho U^{\dagger}, \quad \hat{U}^{\dagger}\rho = U^{\dagger}\rho U, \qquad U \in \mathcal{U}(d), \tag{2}$$

and ρ is a density operator.

Now, suppose the operation $\hat{U}$ is chosen randomly with respect to some measure, for example the Haar distribution η. *Twirling* the quantum channel $\hat{\Lambda}$ with respect to η is defined as

$$\hat{\Lambda} \mapsto \int_{\mathcal{U}(d)} d\eta(U)\hat{U}^{\dagger}\hat{\Lambda}\hat{U}. \tag{3}$$

Intuitively, twirling a channel is equivalent to taking the average, or expected value, of conjugation with all possible $U \in \mathcal{U}(d)$.

The action of the twirled channel on a density operator ρ is

$$\rho \mapsto \int_{\mathcal{U}(d)} d\eta(U)U^{\dagger}\Lambda(U\rho U^{\dagger})U. \tag{4}$$

In general throughout these notes, our quantum channel $\hat{\Lambda}$ will take the explicit form

$$\hat{\Lambda}\rho = \Lambda(\rho) = A\rho B, \tag{5}$$

where A and B are arbitrary linear operators [2]. Then, twirling the channel sends ρ to

$$\begin{aligned}
\rho &\mapsto \int_{\mathcal{U}(d)} d\eta(U)\hat{U}^{\dagger}\hat{\Lambda}\hat{U}\rho && (6)\\
&\mapsto \int_{\mathcal{U}(d)} d\eta(U)\hat{U}^{\dagger}\hat{\Lambda}U\rho U^{\dagger} && (7)\\
&\mapsto \int_{\mathcal{U}(d)} d\eta(U)\hat{U}^{\dagger}AU\rho U^{\dagger}B && (8)\\
&\mapsto \int_{\mathcal{U}(d)} d\eta(U)U^{\dagger}AU\rho U^{\dagger}BU. && (9)
\end{aligned}$$

We can also twirl a channel over a discrete distribution of unitaries, rather than all of $\mathcal{U}(d)$. For some set $\Xi = \{U_1, \ldots, U_K\}$, twirling a channel $\hat{\Lambda}$ with respect to the uniform distribution over set Ξ sends

$$\rho \mapsto \frac{1}{K}\sum_{k=1}^{K} U_k^{\dagger}AU_k\rho U_k^{\dagger}BU_k. \tag{10}$$

3 Designs

3.1 Spherical designs

Spherical designs are inherently classical objects, however they provide excellent intuition for understanding quantum designs.

A spherical design is a set of points on the sphere which "approximate it well" [3], in the following sense. Suppose we have some polynomial in d variables, and we would like to compute its average over the unit sphere $\mathcal{S}(\mathbb{R}^d)$. A spherical design is a set of representative points (i.e. d-dimensional unit vectors) on the surface of the sphere such that computing the average of the function only over these points is identical to taking the average over the entire unit sphere. We can define this more formally as follows [1, 3].

Definition Let $p_t : \mathcal{S}(\mathbb{R}^d) \to \mathbb{R}$ be a polynomial in d variables, with all monomial terms homogeneous in degree at most t. A set $X = \{x : x \in \mathcal{S}(\mathbb{R}^d)\}$ is a ***spherical* t-*design*** if

$$\frac{1}{|X|} \sum_{x \in X} p_t(x) = \int_{\mathcal{S}(\mathbb{R}^d)} p_t(u) d\mu(u) \tag{11}$$

for all possible p_t, where the integral is taken with respect to the normalized spherical measure.

Note that by definition, a t-design is also a $(t-1)$-design, since all monomials can have degree at most t (as long as all degrees are identical). Spherical designs exist for all t and d [1].

Let us take, for example, $\mathbb{R}^3$. A set of points is a 1-design only if its centre of mass is at the origin [1]. If we are averaging linear polynomials, we want to choose a set of points that is balanced. For example, choosing points from only one hemisphere would be a terrible approximation to the whole sphere, while choosing a couple equally-distributed points from each is ideal. Moving up, a tetrahedron is a 2-design, and a cube is a 3-design [1, 3]. The dodecahedron is a 5-design [1, 3]. This is quite intuitive - the more spherical the shape, the better it can approximate the sphere, thus t increases as we can use the design to approximate the integration of functions of higher and higher degree.

Spherical designs are directly related to combinatorial designs; see [4], one of the original papers on spherical designs, for more details (such as the lower bound on the size of spherical designs). Plenty more can be said about spherical designs, but for now we will move on and discuss their close relatives, complex projective designs.

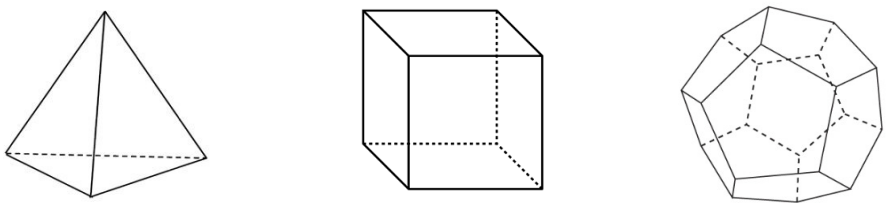

Figure 1: In $\mathbb{R}^3$, a tetrahedron is a spherical 2-design, a cube is a 3-design, and a dodecahedron is a 5-design. As t increases, the shape of the designs becomes rounder, as they are increasingly better approximations of the whole sphere.

3.2 Complex projective designs

Complex projective designs bridge the gap between classical spherical designs and the unitary designs we will soon discuss. They are defined similarly to spherical designs, but instead of a collection of points, or unit vectors in $\mathcal{S}(\mathbb{R}^d)$ we work with unit vectors in $\mathcal{S}(\mathbb{C}^d)$. Furthermore, we consider polynomials with homogeneous degree t, t, meaning they have degree at most t for all monomials in the entries of the vectors, and also homogeneous degree t in the monomials of the complex conjugates of these entries (i.e. we consider polynomials in $2d$ variables).

Definition Let $p_{t,t}$ be a polynomial with homogeneous degree t in d variables, and degree t in the complex conjugates of these variables. A ***complex projective* t*-design*** is a subset X of $\mathcal{S}(\mathbb{C}^d)$ such that

$$\frac{1}{|X|}\sum_{x\in X} p_{t,t}(x) = \int_{\mathcal{S}(\mathbb{C}^d)} p_{t,t}(u)d\eta(u) \tag{12}$$

holds for all possible $p_{t,t}$.

There is a direct correspondence between spherical and complex projective designs. A spherical t-design in $\mathcal{S}(\mathbb{R}^d)$ can be transformed into a $t/2$-design in $\mathcal{S}(\mathbb{C}^{\frac{d}{2}})$ [5]. As well, MUBs and SIC-POVMs, both very useful tools in quantum information theory, are complex projective 2-designs [6, 7].

Complex projective designs can also be written in terms of quantum states, since quantum states are just vectors in complex space. A quantum state design is a probability distribution over a finite set of quantum states such that t copies of a state chosen from this distribution is indistinguishable from t copies of a state chosen randomly from uniform distribution over all possible states [5].

Definition Let $\{p_1, \dots p_K\}$ be a probability distribution over quantum states $\{|\phi_1\rangle, \dots |\phi_K\rangle\}$. Such a distribution is called a ***quantum state design*** if

$$\sum_i p_i \left(|\phi_i\rangle\langle\phi_i|\right)^{\otimes t} = \int_\psi \left(|\psi\rangle\langle\psi|\right)^{\otimes t} d\psi. \tag{13}$$

3.3 Unitary designs

3.3.1 Unitary t-designs

Unitary t-designs are analogous to complex projective designs in the following sense. The polynomials we defined for complex projective designs act on $2d$ variables, the elements of vectors in $\mathcal{S}(\mathbb{C}^d)$ and their complex conjugates. For unitary t-designs, the polynomials are functions on the elements of matrices in $\mathcal{U}(d)$.

Consider a polynomial $P_{t,t}(U)$ which is homogeneous with degree at most t in the matrix elements of U, and at most degree t in the complex conjugates of these elements (i.e. $P_{t,t}$ is a polynomial in $2d^2$ variables). Just as a spherical t-design was some representative set of vectors such that averaging any degree t polynomial over the set was equivalent to averaging over the entire sphere, a unitary t-design is a representative subset of $\mathcal{U}(d)$ such that averaging a degree t, t polynomial over the set is equivalent to averaging over all of $\mathcal{U}(d)$.

Definition A ***unitary* t-*design*** is a set of unitary matrices $\{U_k\}, k = 1, ..., K$ such that

$$\frac{1}{K}\sum_{k=1}^{K} P_{t,t}(U_k) = \int_{\mathcal{U}(d)} d\eta(U) P_{t,t}(U) \tag{14}$$

holds for *every* polynomial $P_{t,t}(U)$.

3.3.2 Unitary 2-designs

Let us return to the process of conjugating a quantum channel by a random unitary (a.k.a twirling):

$$\rho \mapsto \int_{\mathcal{U}(d)} d\eta(U) U^\dagger \Lambda(U\rho U^\dagger) U. \tag{15}$$

Choosing a random unitary over the Haar measure is very difficult - an n-qubit quantum circuit engineered to perform this task would need $\mathcal{O}(n^2 2^{2n})$ 1- and 2-qubit gates [8]. Simply choosing one such unitary, never mind integrating over all possible unitaries is an infeasible task.

Instead, what if, in the spirit of designs, we could just average over a representative set? Namely, does there exist some set of unitaries $\{U_k\}, k = 1, \ldots, K$ such that

$$\frac{1}{K}\sum_{k=1}^{K} U_k^\dagger \Lambda(U_k \rho U_k^\dagger) U_k = \int_{\mathcal{U}(d)} d\eta(U) U^\dagger \Lambda(U\rho U^\dagger) U \ ? \tag{16}$$

A set $\{U_k\}$ satisfying this property is called a unitary 2-design [2, 9]. It turns out that such a set does exist, and it is in fact a very familiar set: the Clifford group. This is an extremely useful result, because Clifford group elements can be synthesized by circuits using only $\mathcal{O}(n^2)$ 1- and 2-qubit gates.

There are many ways of defining unitary 2-designs, two of which were shown here, and all are equivalent (see [9], for example). Unitary 2-designs are also directly connected to complex projective 2-designs. Consider the vectorization of a unitary matrix $\mathrm{vec}(U)$, a function defined by stacking all the columns of U one atop the other in order. Then, for the elements $\{U_1, \ldots, U_K\}$ of a 2-design in $\mathcal{U}(d)$, $\{\mathrm{vec}(U_1), \ldots, \mathrm{vec}(U_K)\}$ is a complex projective 2-design in $\mathcal{S}(\mathbb{C}^{d^2})$ [9].

4 The Clifford group is a 2-design

The main goal of this lecture is to prove that the Clifford group is indeed a 2-design, as claimed. The proof shown here is akin to that of the original authors in [2], but with some of the additional details filled in.

We will consider our channel $\hat{\Lambda}$ to be a linear mapping of the form $\hat{\Lambda}\rho = A\rho B$ where $A, B \in L(\mathbb{C}^d)$. The original paper which proved this fact used a general linear operator X instead of ρ, however I feel using a density matrix is more pertinent to quantum information applications. Furthermore, we could just as well show this for channels where A, B are just Pauli matrices, but I prefer showing the general case, since A and B will just be written as linear combinations of the Paulis anyways.

We would like to show that, for all ρ,

$$\frac{1}{|\mathcal{C}_n|}\sum_{k=1}^{|\mathcal{C}_n|} C_k^\dagger A C_k \rho C_k^\dagger B C_k = \int_{\mathcal{U}(d)} d\eta(U) U^\dagger A U \rho U^\dagger B U. \tag{17}$$

In other words, averaging over a uniform distribution over the Clifford group is equivalent to averaging over the uniform Haar distribution. Furthermore, randomly sampling a unitary from the Clifford group is indistinguishable from sampling from the uniform Haar distribution with respect to this twirling operation (ultimately, for randomly sampling a unitary, a higher t-design will better cover the whole space, in a way analogous to example provided for the spherical design).

We will prove this by evaluating both sides of (17), and showing they are equal.
Fortunately, the RHS of (17) has already been evaluated for us, and has a closed form [10]:

$$\int_{\mathcal{U}(d)} d\eta(U) U^\dagger A U \rho U^\dagger B U = \frac{\mathrm{Tr}(AB)\mathrm{Tr}(\rho)}{d}\frac{\mathbb{1}}{d} + \left(\frac{d\mathrm{Tr}(A)\mathrm{Tr}(b) - \mathrm{Tr}(AB)}{d(d^2-1)}\right)\left(\rho - \mathrm{Tr}(\rho)\frac{\mathbb{1}}{d}\right) \quad (18)$$

The evaluation requires some representation theory and the application of Schur's lemmas, and is outside the scope of this lecture. Here, we will work only with the LHS, and see that with a couple twirling operations, we can show it has the same form as (18).

Consider our channel $\hat{\Lambda}\rho = A\rho B$, and let us perform a twirl with respect to the uniform distribution over the Pauli group, $\mathcal{P}_n$. This sends

$$\rho \mapsto \frac{1}{d^2}\sum_{k=1}^{d^2} P_k^\dagger A P_k \rho P_k^\dagger B P_k = \frac{1}{d^2}\sum_{k=1}^{d^2} P_k A P_k \rho P_k B P_k, \quad (19)$$

since Paulis are Hermitian.

We can simplify this expression by recalling that the Pauli matrices form a basis, so we can rewrite A and B as

$$A = \sum_{a=1}^{d^2} \alpha_a P_a, \quad B = \sum_{b=1}^{d^2} \beta_b P_b. \quad (20)$$

Substituting these into (19), we obtain

$$\rho \mapsto \frac{1}{d^2}\sum_{k=1}^{d^2} P_k \left(\sum_{a=1}^{d^2}\alpha_a P_a\right) P_k \rho P_k \left(\sum_{b=1}^{d^2}\beta_b P_b\right) P_k \quad (21)$$

$$\mapsto \frac{1}{d^2}\sum_{a=1}^{d^2}\sum_{b=1}^{d^2}\alpha_a\beta_b \left(\sum_{k=1}^{d^2} P_k P_a P_k \rho P_k P_b P_k\right) \quad (22)$$

Consider the product $P_k P_a P_k$ for some fixed a (the same argument applies for fixed b). There are two possible results for this product:

$$P_k P_a P_k = \begin{cases} P_a & \text{if} \quad [P_a, P_k] = 0 \\ -P_a & \text{if} \quad \{P_a, P_k\} = 0 \end{cases}, \quad (23)$$

where [] and {} indicate the commutator and anti-commutator respectively. Two Paulis commute if the symplectic inner product of their binary symplectic representations is 0, i.e. $[P_a, P_k] = 0$ if $\langle k, a\rangle_S = 0$.

Thus, we can rewrite the above as

$$\rho \mapsto \frac{1}{d^2}\sum_{a=1}^{d^2}\sum_{b=1}^{d^2}\alpha_a\beta_b\left(\sum_{k=1}^{d^2}(-1)^{\langle k,a\rangle_S}P_a\rho(-1)^{\langle k,b\rangle_S}P_b\right) \tag{24}$$

$$\mapsto \frac{1}{d^2}\sum_{a=1}^{d^2}\sum_{b=1}^{d^2}\alpha_a\beta_b\left(\sum_{k=1}^{d^2}(-1)^{\langle k,a\oplus b\rangle_S}P_a\rho P_b\right), \tag{25}$$

where we make use of the distributivity of the symplectic inner product.

We once again have two cases to consider, for evaluation of the inner product.

1. $\boldsymbol{a = b}$: In this case, $a \oplus b = 0$, so $\langle k, 0\rangle_S = 0$. This means all d^2 of the terms in the sum will have a $+1$ coefficient, so
$$\sum_{k=1}^{d^2}(-1)^{\langle k,0\rangle_S}P_a\rho P_a = d^2 P_a\rho P_a. \tag{26}$$

2. $\boldsymbol{a \neq b}$: Each Pauli commutes with exactly half of the Paulis in $\mathcal{P}_n$, including the identity and itself. Suppose we fix Paulis a and b. When we sum over all k, the $a \oplus b$ will 'commute' with exactly half of the k and lead to a coefficient of $+1$, but it also doesn't commute with the other half, which will have a coefficient of -1. Since we will have an equal number of $+1$ and -1 coefficients,
$$\sum_{k=1}^{d^2}(-1)^{\langle k,a\oplus b\rangle}P_a\rho P_b = 0. \tag{27}$$

Thus, the only terms that contribute are those when $a = b$, and we obtain the following result for our Pauli twirl:

$$\rho \mapsto \frac{1}{d^2}\sum_{a=1}^{d^2}\alpha_a\beta_a d^2 P_a\rho P_a \tag{28}$$

$$\mapsto \sum_{a=1}^{d^2} r_a P_a\rho P_a \tag{29}$$

$$= \hat{\Lambda}_p\rho, \tag{30}$$

where we have set $r_a = \alpha_a\beta_a$, and we denote the Pauli channel as $\hat{\Lambda}_p$.

Now, this doesn't look like anything close to the form of Eq. (17) yet. We will have to perform an additional twirling operation on the Pauli channel.

Recall that the Clifford group is the normalizer of the Pauli group, i.e. Cliffords map Paulis to Paulis under conjugation. In group theoretic terms, $\mathcal{P}_n$ is a *normal subgroup* of $\mathcal{C}_n$. We can thus consider the quotient group $\mathcal{C}_n/\mathcal{P}_n$, which is the Clifford group with the Pauli elements 'modded out'. Let $\{Q_a\}, i = 1, ...$ be a full set of coset representatives of $\mathcal{C}_n/\mathcal{P}_n$, chosen in any way. We note that each element of the larger Clifford group $C \in \mathcal{C}_n$ can be written as the product of a non-Pauli Clifford and a Pauli, $C = Q_i P_j$.

Let us twirl $\hat{\Lambda}_p$ by $\mathcal{C}_n/\mathcal{P}_n$:

$$\rho \mapsto \frac{1}{\frac{|\mathcal{C}_n|}{|\mathcal{P}_n|}} \sum_{a=1}^{|\mathcal{C}_n|/|\mathcal{P}_n|} Q_a^\dagger \Lambda_p(Q_a \rho Q_a^\dagger) Q_a \tag{31}$$

$$\mapsto \frac{|\mathcal{P}_n|}{|\mathcal{C}_n|} \sum_{a=1}^{|\mathcal{C}_n|/|\mathcal{P}_n|} \sum_{s=1}^{d^2} r_s Q_a^\dagger P_s Q_a \rho Q_a^\dagger P_s Q_a \tag{32}$$

$$\tag{33}$$

It is in this form that, having twirled the channel twice, we have molded it into the form of the original LHS of the equation; a uniform distribution over the Clifford group.

We can remove the identity Pauli P_1 from this to obtain the following

$$\hat{\Lambda}_p \rho \mapsto r_1 \rho + \frac{|\mathcal{P}_n|}{|\mathcal{C}_n|} \sum_{a=1}^{|\mathcal{C}_n|/|\mathcal{P}_n|} \sum_{s=2}^{d^2} r_s Q_a^\dagger P_s Q_a \rho Q_a^\dagger P_s Q_a \tag{34}$$

When we consider the action of the entire Clifford group on a single non-identity Pauli, it maps that Pauli to each of the $d^2 - 1$ other possible Paulis an equal number of times. Since we have $|\mathcal{C}_n|/|\mathcal{P}_n|$ possible Cliffords, we get mapped to each Pauli $\frac{|\mathcal{C}_n|/|\mathcal{P}_n|}{d^2-1}$ times. Let us rearrange the previous equation and evaluate this:

$$\rho \mapsto r_1 \rho + \frac{|\mathcal{P}_n|}{|\mathcal{C}_n|} \sum_{s=2}^{d^2} r_s \left(\sum_{a=1}^{|\mathcal{C}_n|/|\mathcal{P}_n|} Q_a^\dagger P_s Q_a \rho Q_a^\dagger P_s Q_a \right) \tag{35}$$

$$\mapsto r_1 \rho + \frac{|\mathcal{P}_n|}{|\mathcal{C}_n|} \sum_{s=2}^{d^2} r_s \left(\frac{|\mathcal{C}_n|}{|\mathcal{P}_n|} \frac{1}{d^2-1} \sum_{c=2}^{d^2} P_c \rho P_c \right) \tag{36}$$

$$\mapsto r_1 \rho + \frac{1}{d^2-1} \left(\sum_{s=2}^{d^2} r_s \right) \sum_{c=2}^{d^2} P_c \rho P_c \tag{37}$$

Now, we need simply to massage this into the form of (18). We will do that by applying a couple useful identities (see Appendix A for their derivation):

1. $r_1 = \frac{\mathrm{Tr}(A)\mathrm{Tr}(B)}{d^2}$
2. $\sum_{k=1}^{d^2} r_k = \frac{1}{d}\mathrm{Tr}(AB)$
3. $\sum_{s=1}^{d^2} P_s \rho P_s = d\mathrm{Tr}(\rho)\mathbb{1}$

Let us substitute these values into (37) above:

$$
\begin{aligned}
& r_1\rho \quad + \frac{1}{d^2-1}\left(\sum_{s=2}^{d^2} r_s\right)\sum_{c=2}^{d^2} P_c\rho P_c \\
&= \frac{\mathrm{Tr}(A)\mathrm{Tr}(B)}{d^2}\rho + \frac{1}{d^2-1}\left(\frac{1}{d}\mathrm{Tr}(AB) - \frac{\mathrm{Tr}(A)\mathrm{Tr}(B)}{d^2}\right)(d\mathrm{Tr}(\rho)\mathbb{1} - \rho) \\
&= \left[\frac{\mathrm{Tr}(A)\mathrm{Tr}(B)}{d^2} - \frac{d\mathrm{Tr}(AB) - \mathrm{Tr}(A)\mathrm{Tr}(B)}{d^2(d^2-1)}\right]\rho + \left(\frac{d\mathrm{Tr}(AB) - \mathrm{Tr}(A)\mathrm{Tr}(B)}{d^2-1}\right)\frac{\mathrm{Tr}(\rho)}{d}\mathbb{1} \\
&= \left[\frac{d^2\mathrm{Tr}(A)\mathrm{Tr}(B) - d\mathrm{Tr}(AB)}{d^2(d^2-1)}\right]\rho - \left(\frac{d\mathrm{Tr}(A)\mathrm{Tr}(B) - d^2\mathrm{Tr}(AB)}{d(d^2-1)}\right)\frac{\mathrm{Tr}(\rho)}{d}\mathbb{1} \\
&= \left[\frac{d\mathrm{Tr}(A)\mathrm{Tr}(B) - \mathrm{Tr}(AB)}{d(d^2-1)}\right]\rho - \left(\frac{d\mathrm{Tr}(A)\mathrm{Tr}(B) - \mathrm{Tr}(AB) - (d^2-1)\mathrm{Tr}(AB)}{d(d^2-1)}\right)\frac{\mathrm{Tr}(\rho)}{d}\mathbb{1} \\
&= \quad \frac{(d^2-1)\mathrm{Tr}(AB)}{d(d^2-1)}\frac{\mathrm{Tr}(\rho)}{d}\mathbb{1} + \left[\frac{d\mathrm{Tr}(A)\mathrm{Tr}(B) - \mathrm{Tr}(AB)}{d(d^2-1)}\right]\left(\rho - \frac{\mathrm{Tr}(\rho)}{d}\mathbb{1}\right) \\
&= \quad \frac{\mathrm{Tr}(AB)\mathrm{Tr}(\rho)}{d}\frac{\mathbb{1}}{d} + \left[\frac{d\mathrm{Tr}(A)\mathrm{Tr}(B) - \mathrm{Tr}(AB)}{d(d^2-1)}\right]\left(\rho - \frac{\mathrm{Tr}(\rho)}{d}\mathbb{1}\right)
\end{aligned}
$$

This is now identical to (18)! Therefore, the Clifford group is an exact unitary 2-design.

□

5 Approximating 2-designs

5.1 Uniformization procedure

The fact that the Clifford group is an exact 2-design is quite useful, however for cases of large qubits this still impractical - the number of elements in the Clifford group scales ludicrously with the number of qubits. Algorithms do exist to randomly sample a single Clifford [11], however after a certain point, surely 'approximating' the integration over all unitaries using the Cliffords will end up being just as cumbersome as doing the original integration.

Approximate 2-designs can help. Let us define $\mathbb{E}_\chi(\Lambda)$ as the twirled channel with action

$$\mathbb{E}_\chi(\Lambda): \quad \rho \mapsto \int_{\mathcal{U}(d)} d\chi(U) U^\dagger \Lambda(U\rho U^\dagger) U, \tag{38}$$

where χ is any measure. If χ is defined as the probability distribution over a discrete subset of $\mathcal{U}(d)$, then the integral takes the form of a sum, akin to the sums we saw when twirling by Paulis and Cliffords.

When we proved that the Clifford group was a unitary 2-design, we were essentially showing the following: for a measure ζ which is the uniform distribution over the Clifford group, then

$$\mathbb{E}_\zeta(\Lambda) = \mathbb{E}_\eta(\Lambda). \tag{39}$$

In other words, the channel obtained by twirling over the Cliffords is equivalent to that obtained by a twirl with respect to the entire Haar measure.

Using the diamond norm to quantify the measure of distance between two channels, we can define an ε-approximate 2-design as follows.

Definition Let χ be a measure over a finite subset of $\mathcal{U}(d)$. An ε-approximate 2-design is a channel $\mathbb{E}_\chi(\Lambda)$ such that

$$||\mathbb{E}_\chi(\Lambda) - \mathbb{E}_\eta(\Lambda)||_\diamond \leq \varepsilon. \tag{40}$$

We can construct approximate 2-designs using randomly generated circuits. By repeatedly applying sequences of conjugation and twirling operations to Pauli channels, we obtain a probability distribution over the possible resultant circuits that approximates a 2-design. Namely, to be within ε of a 2-design, we can use circuits $O(n \log 1/\varepsilon)$ gates, which scales significantly better than the $O(n^2)$ required to simulate an arbitrary Clifford group element.

Let $\{p_1, \ldots, p_m\}$ be a probability distribution for quantum circuits $\{\Phi_1, \ldots, \Phi_m\}$. If we twirl our channel $\hat{\Lambda}$ with such a distribution, we are sending

$$\rho \mapsto \sum_{i=1}^{m} p_i \Phi_i^\dagger \Lambda(\Phi_i \rho \Phi_i^\dagger) \Phi_i. \tag{41}$$

The circuits Φ_i will still consist of Clifford gates - what this approximate construction does is produce a *different*, non-uniform probability distribution over the Clifford group, rather than the original distribution seen in Eq. (17). Furthermore, it allows for a random circuit from $\{\Phi_i\}$ to be quickly and efficiently sampled.

Suppose we are working with n qubits. We will make use of the following two operations:

Defintion Let $R = SH$, where S is the single-qubit phase gate, and H is the Hadamard gate. A $\mathcal{C}_1/\mathcal{P}_1$ ***twirl*** is defined as conjugating a qubit by R^j, where j is randomly chosen from $\{0, 1, 2\}$.

Defintion Let each qubit from $2, \ldots n$ execute a CNOT gate with qubit 1 as the target with independent probability 3/4. This operation is called ***conjugation by a random XOR***.

Using these two operations, we can sample from an approximate 2-design as follows [2]:

1. Perform a Pauli twirl on the channel.
2. Perform a $\mathcal{C}_1/\mathcal{P}_1$ twirl on each of the qubits.
3. Conjugate the first qubit by a random XOR.
4. Conjugate the first qubit by H, and $\mathcal{C}_1/\mathcal{P}_1$ twirl the rest.
5. Conjugate the first qubit by a random XOR.
6. Conjugate the first qubit by H, and $\mathcal{C}_1/\mathcal{P}_1$ twirl the rest.
7. Conjugate the first qubit by S with probability 1/2.
8. Conjugate the first qubit by a random XOR.

9. $\mathcal{C}_1/\mathcal{P}_1$ twirl the first qubit

10. To obtain an ε-approximate 2-design, repeat steps 2-9 $\mathcal{O}(\log(1/\varepsilon))$ times.

The Pauli twirl consists of $\mathcal{O}(n)$ gates, and there are $\mathcal{O}(n)$ twirling operations which follow (every step involves operations on at most n qubits). Since we repeat the procedure $\log(1/\varepsilon)$ times, the number of gates is on the order of $\mathcal{O}(n \log(1/\varepsilon))$. The depth of the circuits is $\mathcal{O}(\log n \log(1/\varepsilon))$, as the only contributors are the CNOTs, which by the divide-and-conquer technique can be done in depth $\mathcal{O}(\log n)$.

The distribution over all possible circuits is an ε-approximate 2-design. This procedure essentially serves to 'mix' the Paulis. Consider only one component of the initial Pauli twirl, $\rho \to P_a \rho P_a$, where P_a is not the identity. If the distribution of circuits is ε close to a 2-design, a circuit randomly chosen according to this uniformization procedure should be equally likely to send P_a to any of the other non-identity Paulis.

5.2 The 2-Designer

It is fine to believe that the theoretical procedure above works, and a proof is provided in [2]. However, things are always much more convincing when you can actually *see* them work. To that end, I wrote a computer program called 'The 2-Designer' in Python to perform the random generation of circuits using the above procedure. The code can be downloaded from the Github repository `https://github.com/glassnotes/The-2-Designer`. It is written using the open source library QuaEC, which contains efficient means of manipulating Pauli and Clifford operations [12].

What I sought to do is to generate a number of circuits using the uniformization procedure above, and test the resultant Pauli distribution when all circuits are applied to the same initial Pauli. If the procedure works as stated, the resultant distribution on the Paulis should be uniform.

I ran the code for 2 different values of ε, on the same initial 2-qubit input Pauli (ZZ). For $\varepsilon = 1$ I generated 10,000 circuits randomly using the uniformization procedure, and for $\varepsilon = 0.1$ I generated 1,000,000 circuits. This difference in size is due to the large amount of circuits that are possible in the $\varepsilon = 0.1$ case. For $\varepsilon = 1$, there were just under 3,000 possible circuits recorded. On the other hand, for 1,000,000 iterations at $\varepsilon = 0.1$, there were 999911 distinct circuits - if we just ran 10,000 iterations here, then it would not even be close to a good sampling of the space, so the number of circuits generated had to be higher.

The resulting distributions on the non-identity output Paulis are shown in Table 1; for the two qubit case, we can see that the distributions are fairly uniform, with most values not deviating very much from the ideal value of 1/15, or 0.0667. Of course, the results for $\varepsilon = 0.1$ are more uniform than the $\varepsilon = 1$ case. Sampling from a distribution with higher ε would take too much time to be practical, unfortunately.

In order to quantize the (non-)uniformity of the distribution, we can compute the Kullback-Leibler divergence - this is a measure of the *relative entropy* between the two probability distributions [13]. The relative entropy is of course 0 if two distributions are the same; otherwise, it quantizes the amount of information (in bits) that would be lost were we to use one distribution to approximate another. The Python package SciPy contains an implementation of the Kullback-Leibler divergence [14], which was very easy to integrate into the code. The output values are recorded in Table 1, and we can see that the distributions produced by The 2-Designer, for both values of ε sampled, are quite good. For $\varepsilon = 1$ less than 1/1000 of a bit of information is lost, and for $\varepsilon = 0.1$ less than one 1/100000.

	$\varepsilon = 1$	$\varepsilon = 0.1$
Circuits generated	10000	1000000
Distinct circuits	2894	999911
Pauli		
IX	0.0628	0.066290
IZ	0.0690	0.066828
IY	0.0639	0.066339
XI	0.0670	0.067035
XX	0.0640	0.066785
XZ	0.0687	0.066785
XY	0.0663	0.066493
ZI	0.0676	0.066476
ZX	0.0690	0.066731
ZZ	0.0692	0.066622
ZY	0.0641	0.067049
YI	0.0664	0.066502
YX	0.0700	0.066340
YZ	0.0686	0.066818
YY	0.0634	0.066907
Rel. Entropy (bits)	$6.357 \cdot 10^{-4}$	$6.542 \cdot 10^{-6}$

Table 1: Resultant distribution of Paulis after running the uniformization procedure for a given ε. 10,000 separate circuits were generated fpr $\varepsilon = 1$ and 1,000,000 for $\varepsilon = 0.1$ (to better cover the space) and were applied to the initial Pauli ZZ to obtain the results. Relative entropy is computed using the Kullback-Leibler divergence.

Thus, we have explicitly shown that The 2-Designer, and the underlying method, are an excellent means of sampling random Clifford circuits, and approximating 2-designs.

6 Some properties of unitary designs

6.1 When do unitary designs exist?

Some combinatorial designs, and even quantum designs do not exist in every dimension. One can only find a set of MUBs, for example, in prime and prime power dimensions. Similarly, projective and affine planes exist only in prime and prime power dimensions [1]. It is likely that this can be attributed to their underlying construction based on finite fields, as it is well-known that such fields exist only for primes and powers of primes. Particular forms of SICPOVMs, on the other hand, are conjectured to exist in all dimensions [15].

A natural question that arises then, is "In what dimensions do unitary t-designs exist?". It turns out that it has been shown that these designs exist for all possible combinations of t and d, a result which may be somewhat surprising [16]. Despite proof of their existence, actual construction methods for designs have not been found for every case.

6.2 How big are designs?

A very pertinent question that might have come up while reading these notes, is how many unitaries are there in a 2-design? This question was addressed in [9] and produced the following lower bound.

Definition $d^4 - 2d^2 + 2$ is a lower bound for the size of any unitary 2-design in dimension d [9, 16].

Let us consider the case of 2 qubits. The Clifford group has 11520 elements - the lower bound, however, evaluates to $4^4 - 2 \cdot 4^2 + 2 = 226$, which is significantly smaller. Finding such a design, however, may be cumbersome.

A lower bound has also been computed for a general t-design in dimension d, based on the sizes of the irreducible representations of $\mathcal{U}(d)$. As we are most interested in 2-designs for the purpose of these notes, such a calculation is outside our scope, but can be found in [16], where they note that their result collapses to the bound above for the $t = 2$ case.

References

[1] C. J. Colbourn and J. H. Dinitz, eds. *Handbook of Combinatorial Designs,* 2nd ed. (2007) CRC Press.

[2] C. Dankert, R. Cleve, J. Emerson, and E. Livine (2009) Phys. Rev. A **80** 012304

[3] E. Bannai and E. Bannai (2009) European J. Combin. **30** 1392-1425

[4] P. Delsarte, J. M. Goethals and J.J. Seidel (1977) Geometriae Dedicata **6** 363-388

[5] A. Ambainis and J. Emerson (2007) Twenty-Second Annual IEEE Conference on Computational Complexity. 129-140

[6] A. Klappenecker and M. Rötteler (2005) Proceedings of SPIE Vol. 5914

[7] A. Klappenecker and M. Rötteler (2005) Proceedings of International Symposium on Information Theory. 1740-1744

[8] J. Emerson, Y. S. Weinstein, M. Saraceno, S. Lloyd and D. G. Cory (19 December 2003) *Pseudo-Random Unitary Operators for Quantum Information Processing.* Science **302** 2098-2100

[9] D. Gross, K. Audenaert, and J. Eisert (2007) arXiv:quant-ph/0611002v2

[10] J. Emerson, R. Alicki, and K. Życzkowski (2005) J. Opt. B: Quantum Semiclass. Opt. **7** S347-S352

[11] R. Koenig and J. A. Smolin (2014) http://arxiv.org/abs/1406.2170

[12] `https://github.com/cgranade/python-quaec`

[13] Chris Granade, online conversation, 28/10/14.

[14] `http://docs.scipy.org/doc/scipy-dev/reference/generated/scipy.stats.entropy.html`

[15] J. M. Renes, R. Blume-Kohout, A. J. Scott, and C. M. Caves (2004) J. Math. Phys. **45** (6) 2171-2180

[16] A. Roy and A. J. Scott (2009) Des. Codes Cryptogr. **53** 13 - 31

I would like to acknowledge Richard Cleve, who helped point me in the right direction and clarified many things, especially regarding the uniformization procedure for approximate 2-designs. Chris Granade was also very helpful on the software side, and showed me how to best use his library QuaEC for the purposes of The 2-Designer. Finally, Peter Turner referred me to some useful resources for understanding the difference between spherical designs and complex projective designs.

A Mini proofs

These short, simple proofs of the identities used to prove (17) were relegated to this Appendix to de-clutter the main text.

A.1 $r_1 = \mathbf{Tr}(A)\mathbf{Tr}(B)/d^2$

Recall that we expanded

$$A = \sum_{a=1}^{d^2} \alpha_a P_a, \quad B = \sum_{b=1}^{d^2} \beta_b P_b. \tag{42}$$

and defined $r_i = \alpha_i \beta_i$ after performing the Pauli twirl.

We can compute α_1 by taking the trace:

$$\begin{aligned} \mathrm{Tr}(A) &= \mathrm{Tr}\left(\sum_{a=1}^{d^2} \alpha_a P_a\right) && (43)\\ &= \sum_{a=1}^{d^2} \alpha_a \mathrm{Tr}(P_a) && (44)\\ &= \alpha_1 d && (45) \end{aligned}$$

since the trace of all non-identity Paulis is 0. Thus, $\alpha_1 = \mathrm{Tr}(A)/d$. Similary, $\beta_1 = \mathrm{Tr}(B)/d$. We easily combine the two to obtain

$$r_1 = \alpha_1 \beta_1 = \frac{\mathrm{Tr}(A)\mathrm{Tr}(B)}{d^2}. \tag{46}$$

□

A.2 $\sum_{a=1}^{d^2} r_a = \mathbf{Tr}(AB)/d$

Begin with the RHS and expand:

$$\begin{aligned} \frac{\mathrm{Tr(AB)}}{d} &= \frac{\mathrm{Tr}\left(\sum_{a=1}^{d^2} \alpha_a P_a \sum_{b=1}^{d^2} \beta_b P_b\right)}{d} && (47)\\ &= \frac{\sum_{a=1}^{d^2}\sum_{b=1}^{d^2} \alpha_a \beta_b \mathrm{Tr}(P_a P_b)}{d} && (48) \end{aligned}$$

Of course, unless $P_a P_b = \mathbb{1}$ the trace of their product will 0. Since each Pauli is it's own inverse, terms will contribute only if $a = b$, so we obtain

$$\begin{aligned} \frac{\mathrm{Tr(AB)}}{d} &= \frac{\sum_{a=1}^{d^2} \alpha_a \beta_a \mathrm{Tr}(\mathbb{1})}{d} && (49)\\ &= \sum_{a=1}^{d^2} \alpha_a \beta_a && (50)\\ &= \sum_{a=1}^{d^2} r_a. && (51) \end{aligned}$$

□

A.3 $\sum_{s=1}^{d^2} P_s \rho P_s = d\mathbf{Tr}(\rho)\mathbb{1}$

Let us begin by expanding ρ in the Pauli basis: $\rho = \sum_{x=1}^{d^2} \alpha_x P_x$.

We can evaluate $\mathrm{Tr}(\rho)$ using the linearity of the trace operation:

$$\begin{aligned} \mathrm{Tr}(\rho) &= \mathrm{Tr}\left(\sum_{x=1}^{d^2} \alpha_x P_x\right) && (52)\\ &= \sum_{x=1}^{d^2} \alpha_x \mathrm{Tr}(P_x) && (53)\\ &= \alpha_1 d, && (54) \end{aligned}$$

since the trace of all Paulis operators except the identity is 0, and $\mathrm{Tr}(\mathbb{1}) = d$ of course.

Now, let us evaluate the sum of ρ conjugated by all the Paulis:

$$\begin{aligned} \sum_{s=1}^{d^2} P_s \rho P_s &= \sum_{s=1}^{d^2} P_s \left(\sum_{x=1}^{d^2} \alpha_x P_x\right) P_s && (55)\\ &= \sum_{s=1}^{d^2}\sum_{x=1}^{d^2} \alpha_x P_s P_x P_s && (56)\\ &= \sum_{x=1}^{d^2} \alpha_x \left(\sum_{s=1}^{d^2} (-1)^{\langle s,x\rangle_S} P_x\right) && (57) \end{aligned}$$

Using the same argument as before, summing over s, the symplectic inner product will be $+1$ and -1 an equal amount of times, unless P_x is the identity. Thus, we obtain

$$\sum_{s=1}^{d^2} P_s X P_s = \alpha_1 d^2 \mathbb{1} \tag{58}$$

If we substitute in the value of $\mathrm{Tr}(\rho)$ from (54), we obtain

$$\sum_{s=1}^{d^2} P_s \rho P_s = d\mathrm{Tr}(\rho)\mathbb{1}, \tag{59}$$

which completes the proof. □