

A reduced complexity numerical method for optimal gate synthesis

Srinivas Sridharan,¹ Mile Gu,² Matthew R. James,¹ and William M. McEneaney³

¹*School of Engineering, Department of Information Engineering,
Australian National University, Canberra, ACT 0200, Australia.**

²*Center for Quantum Technologies, National University of Singapore, Singapore*

³*Department of Mechanical and Aerospace Engineering University of California, San Diego, U.S.A*

Although quantum computers have the potential to efficiently solve certain problems considered difficult by known classical approaches, the design of a quantum circuit remains computationally difficult. It is known that the optimal gate design problem is equivalent to the solution of an associated optimal control problem, the solution to which is also computationally intensive. Hence, in this article, we introduce the application of a class of numerical methods (termed the max-plus curse of dimensionality free techniques) that determine the optimal control thereby synthesizing the desired unitary gate. The application of this technique to quantum systems has a growth in complexity that depends on the cardinality of the control set approximation rather than the much larger growth with respect to spatial dimensions in approaches based on gridding of the space, used in previous literature. This technique is demonstrated by obtaining an approximate solution for the gate synthesis on $SU(4)$ - a problem that is computationally intractable by grid based approaches.

PACS numbers: 03.67.Lx, 02.70.-c

I. INTRODUCTION

The advent of Shor's algorithm [1] demonstrated the potential for processors based on quantum operations to perform certain computational tasks exponentially faster than those limited to using classical operations. There has been much work devoted to solving the following problem of special interest: to determine the bounds on the number of one and two qubit gates required to perform a desired unitary operation—termed the gate complexity of the unitary. Yet, the explicit design of quantum algorithms has remained a challenging task.

One approach to this task of constructing an optimal circuit was highlighted in [2] where it was shown to be equivalent to finding a least path-length trajectory on a Riemannian manifold. This insight opened up the study of quantum circuit complexity to the use of tools from optimal control theory.

In [3] the method of dynamic programming was introduced to solve the control problem associated with quantum circuit complexity. The numerical computations of solutions using this technique proceeded via a widely used grid (mesh) based iteration approach [4–6] that requires the generation of a mesh in the region of the state space over which the solution is sought. This approach however, leads to the following issue. A grid (assumed for simplicity to be regular and rectangular) with K points along each of the N dimensions has K^N grid points, over which the solution must be propagated during each iteration. In addition, the dimension of an n qubit quantum system grows exponentially (as $4^n - 1$) thereby leading to a similar exponential growth in memory and time re-

quirements. This large growth in the resources required, arising from growth in the dimensions of the system, is termed the *curse of dimensionality* (COD). It renders the direct application of mesh based solution techniques unfeasible for systems larger than $SU(2)$ due to the large memory (in terabytes) and time (in centuries) required to solve problems of these dimensions via such methods.

In [7, 8] a COD-free technique was introduced for problems in Euclidean space. In this article we adapt these methods for quantum systems. Due to the structure of the control problem that we consider, we do not completely eliminate the COD. However we have a much more manageable growth related to the number of elements in the discretized control set used. This is managed via a pruning approach described in Sec. IV. The computational time of the resulting algorithm grows much slower than that in mesh based methods, thereby bringing us closer to the numerical study of larger systems. One particular application of interest for the numerical methods developed is the determination of whether a given unitary U in an n -qubit system can be approximately synthesized in an efficient manner (with respect to the growth in n) in a given time T .

The paper is structured as follows. Sec. II gives a brief introduction to the relevant concepts in quantum complexity and optimal control. We then introduce the reduced complexity algorithm in Sec. III, and in Sec. IV highlight the complexity growth in the application of this method and its management. The algorithm is then applied in Sec. V to the two qubit optimal gate synthesis problem on $SU(4)$. Such a complete analysis on $SU(4)$ - a problem in 15 dimensional space - has not been carried out in any previous literature. In Sec. VI we conclude with comments on various aspects of the technique introduced in this article.

*E-mail address to which correspondence should be sent:
srinivas.sridharan@anu.edu.au

II. PRELIMINARY CONCEPTS

In this section we recall the notion of gate complexity and introduce the cost function for an associated control problem as in [3, 9].

A. Gate complexity and control

In quantum computing an algorithm operating on an n qubit system can be represented as an element of the Lie group $SU(2^n)$ (denoted in this article by \mathbf{G}) and is termed a unitary. Every such unitary can be constructed by a sequence of available elementary unitaries $U_1, U_2 \dots U_n$. In practice, we synthesize a unitary \hat{U}_0 that approximates a desired computation U_0 with a required accuracy ϵ (i.e. $\|U_0 - \hat{U}_0\| \leq \epsilon$, where $\|\cdot\|$ denotes the standard matrix norm). This leads to the notion of approximate gate complexity $G(U_0, \epsilon)$ which is the minimal number of one and two qubit gates required to synthesize U_0 up to an accuracy of ϵ without ancilla qubits [9].

Related to the gate synthesis problem is an optimal control problem (described below) on \mathbf{G} , such that the approximate gate complexity scales equivalently up to a polynomial in the optimal cost function for the control problem. This equivalence motivates the solution of the associated control problem.

We now describe the control problem and recall the solution process, via the dynamic programming principle, used in [3].

B. System Description

The system dynamics for the gate design problem is given by:

$$\frac{dU}{dt} = -i \left\{ \sum_{k=1}^M v_k(t) H_k \right\} U, \quad U \in \mathbf{G} \quad (1)$$

with control v (such that $v(t) \in \mathbb{R}^M, \forall t \geq 0$) and an initial condition $U(0) = U_0$. For the class of problems considered, v is taken to be an element of the set of piecewise continuous functions having a norm bound $\|v(\cdot)\| = 1$ (where $\|\cdot\|$ denotes the standard 2-norm on \mathbb{R}^M). We denote this class of controls by \mathcal{V} . The system equation contains a set of right invariant vector fields $-iH_1, -iH_2 \dots, -iH_M$, which correspond to the set of available one and two qubit Hamiltonians. The span of the set $\{-iH_1, -iH_2 \dots, -iH_M\}$ (and all brackets thereof) is assumed to be the Lie algebra \mathfrak{g} of the group \mathbf{G} ¹. Under these assumptions it follows from [10, Prop.

3.15] that the time to move the state, from the identity element to any other point on the group, is bounded (and hence, the minimum time to move between any two points on \mathbf{G} is finite). Given a control signal v and an initial unitary U_0 at time r the solution to Eq (1) at time t is denoted by $U(t; v, r, U_0)$.

The control problem involves generating a desired state U_0 of the system in Eq (1) starting from the identity element. By time reversal of the dynamics Eq (1) it can be seen that this is equivalent to the problem of reaching the identity element starting from U_0 . The optimal cost function for this control problem is given by the geodesic distance

$$C_0(U_0) = \inf_{v \in \mathcal{V}_g} \left\{ \int_0^{t_{U_0}(v)} \sqrt{v(s)^T R v(s)} ds \right\}, \quad (2)$$

$$\mathcal{V}_g := \{v(\cdot) \mid \|v\| = 1, \\ v_i : [0, \infty) \rightarrow \mathbb{R} \text{ is piecewise continuous} \\ \text{for all } i\},$$

where $t_{U_0}(v)$ is the time to reach the identity starting from U_0 and is defined by

$$t_{U_0}(v) = \inf\{t > 0 : U(0) = U_0, U(t) = I, \\ \text{dynamics in (1)}\}. \quad (3)$$

This time is taken to be $+\infty$ if the terminal constraint $U(t) = I$ is not attained.

The diagonal, symmetric and positive-definite weight matrix R in Eq (2) reflects the relative difficulty of generating each element of the control vector. For instance, on $SU(4)$, the two body unitary direction $\sigma_x \otimes \sigma_x$ may be weighted more than the single body unitary $\sigma_x \otimes I$ (as it is often harder to manipulate the former than the latter). The symbols $\sigma_x, \sigma_y, \sigma_z$ denote the standard Pauli matrices.

One approach to optimal control problems such as in Eq (2) is the *dynamic programming* method developed in the 1950's by R. Bellman (see [11]). It relates the optimal cost evaluated at an initial time r at a point U_0 to the optimal cost evaluated at a point $U(r+t; v, r, U_0)$ that is reached after applying the control signal v for a time duration of length t . This relation takes the form

$$C_r(U_0) = \inf_{v \in \mathcal{V}_g} \left\{ \int_r^{r+t} \sqrt{v(s)^T R v(s)} ds + \right. \\ \left. C_{r+t}(U(r+t; v, r, U_0)) \right\}, \quad (4)$$

for all points $U_0 \in \mathbf{G}$ and is termed the dynamic programming equation (DPE). For any function φ this dynamic programming relation can be expressed as

$$C_r(U_0) = S_{r, (r+t)}(C_{(r+t)})[U_0], \quad U_0 \in \mathbf{G} \quad (5)$$

¹ Note that we use the convention from mathematics where elements of the Lie algebra are skew Hermitian. This is also consistent with the fact that the Hamiltonians are Hermitian.

where

$$S_{r,(r+t)}(\varphi)[U_0] := \inf_{v \in \mathcal{V}_g} \left\{ \int_r^{r+t} \sqrt{v(s)^T R v(s)} ds + \varphi(U(r+t; v, r, U_0)) \right\}. \quad (6)$$

Note that for ease of notation we will denote the operator $S_{r,(r+t)}(\cdot)$ as $S_t(\cdot)$ when the value of r is clear from the context.

In order to completely characterize the solution to the DPE (4) we require boundary conditions given by

$$C(U) \begin{cases} = 0, & U = I \\ > 0, & U \in \mathbf{G} \setminus \{I\}. \end{cases} \quad (7)$$

These conditions reflect the fact that if we start at the identity I , then the cost to reach the identity is zero. Furthermore, if $U_0 \neq I$, then a non-zero amount of time is needed to reach the identity as the control values are bounded.

The solution (i.e. the optimal cost function) to the control problem can be obtained by solving a specific partial differential equation (a differential version of the DPE), termed the Hamilton-Jacobi-Bellman (HJB) equation [5], given by

$$H(U, DC) = 0, \quad U \in \mathbf{G} \setminus \{I\} \quad (8)$$

with

$$H(U, p) := \sup_{\|v\|=1} \left\{ \left[-i \left\{ \sum_{k=1}^M v_k H_k \right\} U \right] \cdot p - \sqrt{v^T R v} \right\}.$$

with the boundary conditions in Eq (7). The DPE/HJB equation encodes considerable information about the problem, and can be solved for the optimal cost function. The DPE can then be used to construct/verify optimal control strategies via the *verification theorem*, [5, Sec. 1.5]).

Various approaches exist to obtain the solution of the HJB equation (8). The most common are grid based methods [4, 5, 12, 13] which require a mesh to be generated over the state space. Due to the COD which leads to infeasible memory and time requirements these methods are unsuitable for larger dimension systems. Hence alternative approaches to this problem are required.

III. THE REDUCED COMPLEXITY ALGORITHM

Recently, a class of algorithms that are not subject to the curse of dimensionality were introduced in [7, 8, 14] to

solve first order HJB equations in Euclidean space. This method, termed the *max-plus curse of dimensionality free* approach, to solve the dynamic programming equations involves a propagation of the solution of the HJB equation forward in fixed time steps without discretization in the spatial dimensions. The dramatic speed up of this approach stems from an invariant structure that the cost function possesses, which is preserved under the above propagation. This invariant form helps reduce the amount of information that must be stored while solving the control problem. In this section we introduce this method and describe how the invariant structure arises from the cost function. We then apply this method to obtain a numerical procedure to determine the solution to the control problem.

The COD-free max-plus theory [8] to obtain and use an invariant form of the cost function does not currently deal with cost functions containing terminal constraints (as is the case in this problem where the trajectory must reach the identity element such as in Eq (2)). Hence we formulate a relaxed version of the problem, that can be solved via this theory.

A. Relaxation of the optimal cost function

One possible relaxation of the cost function in Eq (2) proceeds by introducing a fixed terminal time T and a terminal penalty cost to yield the expression

$$V_s^\epsilon(U_0) = \inf_{v \in \mathcal{V}_g^\epsilon} \left\{ \int_s^T \sqrt{v(t)^T R v(t)} dt + \frac{1}{\epsilon} \phi(U(T; v, s, U_0)) \right\}, \quad (9)$$

where $\phi(\cdot)$ is a real valued non-negative function (that is zero only at the identity element). This function penalizes terminal states away from the identity.

The extended control set \mathcal{V}_g^ϵ above, is defined as

$$\mathcal{V}_g^\epsilon := \mathcal{V}_g \cup \{v \equiv 0\}, \quad (10)$$

where $\{v \equiv 0\}$ denotes the control signal that is identically zero. We note that due to the fixed time horizon in Eq (9) this extension to the control set ensures that once the target set is reached, the cost function does not increase further.

In this article we take the terminal cost to be of the form

$$\begin{aligned} \phi(U) &= \text{tr}[(I - U) \times (I - U)^\dagger] \\ &= \text{tr}[2I - U - U^\dagger], \end{aligned} \quad (11)$$

$$= 2 \text{tr}[I] - 2 \text{Re}(\text{tr}(U)). \quad (12)$$

where ‘tr’ denotes the trace operation and $\text{Re}(\cdot)$ denotes the projection onto the real axis. The above relaxation

is valid since, as the penalty increases, the approximation $V_0^\epsilon(U)$ converges to $V(U)$ for all points $U \in \mathbf{G}$ (for a sufficiently large time horizon T).

We recall that due to the assumptions outlined in Sec. II B, the time to move between any 2 points in the group \mathbf{G} is bounded. The minimum time to move from U_0 to I is denoted by

$$t_{U_0} = \inf_{v \in \mathcal{V}_g^\epsilon} [t_{U_0}(v)]. \quad (13)$$

We define

$$T_0 := \sup_{U_0 \in \mathbf{G}} [t_{U_0}] \quad (< \infty) \quad (14)$$

to be the maximum value of this time over all points in the group. Hence choosing a T to be T_0 in Eq (9) ensures that it is applicable for any initial point $U_0 \in \mathbf{G}$.

We now propagate the relaxation V_s^ϵ , that satisfies the DPE in Eq (5), in time steps of τ . Let $T = N\tau$, for some $N \in \mathbb{N}$. Using the notation

$$\tilde{V}_k^\epsilon(U) = V_{T-k\tau}^\epsilon(U)$$

and Eq (5) we rewrite Eq (9) as the propagation of the function \tilde{V}^ϵ forward in time through a time interval τ due to the action of the operator $S_\tau[\cdot]$:

$$\tilde{V}_{k+1}^\epsilon(U) = S_\tau[\tilde{V}_k^\epsilon(\cdot)](U). \quad (15)$$

Hence we repeatedly apply $S_\tau[\cdot]$ to move towards the desired value function $V_0^\epsilon(\cdot)$ ($\cong \tilde{V}_N^\epsilon(\cdot)$).

From Eq (9), (15) it follows that for all $\tilde{U} \in \mathbf{G}$,

$$\tilde{V}_{k+1}^\epsilon(\tilde{U}) = \inf_{v \in \mathcal{V}_\tau^\epsilon} \left\{ (\sqrt{v^T R v}) \times \tau + \tilde{V}_k^\epsilon(U(\tau; v, k\tau, \tilde{U})) \right\}, \quad (16)$$

which indicates the action of the DPE operator $S_{k\tau, (k+1)\tau}[\cdot]$. We now describe how the properties of this operator lead to the invariant structure of the optimal cost function Eq (9), and the implications resulting therefrom.

B. Invariant structure of the optimal cost function

The terminal cost in Eq (11) can be written as

$$\phi(U) = c_0 + P_0(U), \quad U \in \mathbf{G}, \quad (17)$$

where

$$c_0 = 2n, \quad P_0(U) = -\text{tr}[U] - \text{tr}[U^\dagger]. \quad (18)$$

Hence c_0, P_0 encode the initial costs corresponding to a control I (i.e. no control action). Assume that for a given ‘ k ’ ($k \in [1, 2, \dots, N-1]$), the cost function \tilde{V}_k^ϵ can be written as

$$\tilde{V}_k^\epsilon(U) = c_k + P_k(U). \quad (19)$$

From Eqns (15) and (16), after one time step τ the value function becomes

$$\tilde{V}_{k+1}^\epsilon(U_0) = S_\tau[\tilde{V}_k^\epsilon](U_0) \quad (20)$$

$$= \min_{v \in \mathcal{V}_\tau^\epsilon} \{ (\sqrt{v^T R v}) \tau + c_k + P_k(U((k+1)\tau; v, k\tau, U_0)) \}, \quad (21)$$

$$= \min_{v \in \mathcal{V}_\tau^\epsilon} \{ (\sqrt{v^T R v}) \tau + c_k + P_k(\Psi[(k+1)\tau, k\tau, v] \cdot U_0) \}, \quad (22)$$

$$:= \min_{v \in \mathcal{V}_\tau^\epsilon} p_k^v(U_0), \quad (23)$$

where $\forall U_0 \in \mathbf{G}$

$$p_k^v(U_0) := \{ (\sqrt{v^T R v}) \tau + c_k + P_k(\Psi[(k+1)\tau, k\tau, v] \cdot U_0) \}, \quad (24)$$

$$\Psi[t, s, v] \cdot U_0 := U(t; v, s, U_0). \quad (25)$$

In the equations above, $\Psi(\cdot)$ denotes the propagator for the system dynamics in Eq (1) under the action of a control signal v .

Hence

$$\tilde{V}_{k+1}^\epsilon(U_0) = c_{k+1} + P_{k+1}(U_0), \quad (26)$$

where

$$c_{k+1} := c_k + (\sqrt{\bar{v}^T R \bar{v}}) \tau, \quad (27)$$

$$P_{k+1}(U_0) := P_k(\Psi[(k+1)\tau, k\tau, \bar{v}] \cdot U_0), \quad (28)$$

$$\text{where } \bar{v} = \arg \min_{v \in \mathcal{V}_\tau^\epsilon} [p_k^v(U_0)]. \quad (29)$$

By the principle of induction, from Eqns. (17), (19), (26) it may be seen that $S_\tau[\cdot]$ preserves the structure of the cost function. This invariance of the structure is a key aspect of the class of techniques introduced, as it helps obtain the optimal cost function at desired points without having to discretize along the spatial dimensions. This optimal cost function for any point U_0 is

$$\tilde{V}_N^\epsilon(U_0) = \min_{k \in \{1, 2, \dots, N\}} \min_{v \in \prod_k \mathcal{V}_\tau^\epsilon} p_k^v(U_0), \quad (30)$$

where \prod_k denotes the k fold product of the control set $\mathcal{V}_\tau^\epsilon$. Hence once a computationally efficient parameterization of the control signals in terms of the set of p values is obtained as described above, Eq (30) easily yields the cost function. We note that the computation of P_k and $\Psi[t, s, v] \cdot U_0$ can be performed efficiently as they can be reduced to matrix multiplications and trace operations on matrices. The generation of a set of parameters, by using the invariant structure of the cost, and its application to determine the optimal cost function is the essence of the max-plus COD free technique.

From Eqns (23),(24) it is clear that during each time step there is an increase in the size of the number of

candidate controls to be considered during the minimization. Specifically, the number of elements $p_k^e[\cdot]$ that result from each P_k is the cardinality of the control set \mathcal{V}_τ^e . Thus, as in the COD free method in Euclidean space, due to the avoidance of spatial discretization the problem is free of the growth in dimensionality arising from spatial terms; however due to the structure of the quantum control problem *there is now a geometric growth in complexity*. The details of this growth and methods to reduce its impact are now described.

IV. CONTROL SPACE GROWTH AND PRUNING

For the purpose of implementation let the control space be discretized as follows: the control signal is held constant over each particular time period of duration τ . Furthermore at most one component of the control vector is set to a value of 1 over any time period, while the others are kept at 0. This class of control signals is denoted by $\tilde{\mathcal{V}}_\tau^e$. Note from Eq (22) that after each time step there is a factor of $[\#(\tilde{\mathcal{V}}_\tau^e)]$ growth in the number of control sequences in the set to be considered, where $\#(A)$ indicates the cardinality of a set A . Hence after N time steps the number of possible control sequences is $[\#(\tilde{\mathcal{V}}_\tau^e)]^N$.

To manage this growth we introduce a selective removal (termed pruning) of some of these control sequences. To describe this pruning procedure we first introduce the required notation. The set of control sequences of length k (i.e. k time step sequence) is denoted by Λ_k . The set of all such control sequences of all possible lengths $\{1, 2, \dots, N\}$ is

$$\Lambda := \bigcup_{k=1}^N \Lambda_k.$$

As indicated in the previous section there is a cost function $p_\lambda(\cdot)$, associated with each control $\lambda \in \Lambda$ such that the discretized cost function $\tilde{V}_N^\epsilon(U) \cong V_0(\cdot)$ can be determined for any point $U \in \mathbf{G}$ by

$$\tilde{V}_N^\epsilon(U) = \min_{\lambda \in \Lambda} p_\lambda(U), \quad (31)$$

To decide upon pruning some of the control sequences $p_\lambda(\cdot)$ we first determine its contribution to the minimization of the cost function. A control $\bar{\lambda}$ (and the corresponding function $p_{\bar{\lambda}}$) contributes to the minimization of the cost function Eq (31) iff

$$\exists U \in \mathbf{G} \text{ such that } p_{\bar{\lambda}}(U) < p_\lambda(U), \quad \forall \lambda \in \bar{\Lambda}, \quad (32)$$

where $\bar{\Lambda}$ denotes the set of control sequences $\Lambda \setminus \{\bar{\lambda}\}$ which are different from the sequence $\bar{\lambda}$. This idea can be used to measure the contribution of any control sequence towards the minimization of the cost in Eq (31). From [14] one such function that quantifies this contribution is

$$W(\bar{\lambda}) \doteq \max\{h(\zeta, U) \mid \bar{p}_\lambda(U) - \zeta \geq 0, \forall \lambda \in \bar{\Lambda}, U \in \mathbf{G}, \zeta \in \mathbb{R}\}, \quad (33)$$

where

$$\begin{aligned} \bar{p}_\lambda(U) &\doteq p_\lambda(U) - p_{\bar{\lambda}}(U) \quad \forall \lambda \in \bar{\Lambda}, \\ h(\zeta, U) &\doteq \zeta, \quad \zeta \in \mathbb{R}, \end{aligned} \quad (34)$$

If $W(\bar{\lambda}) \leq 0$, then $p_{\bar{\lambda}}$ never achieves the minimum for any point U in \mathbf{G} , and consequently, the control sequence $\bar{\lambda}$ can be pruned without any effect on the cost function. For those $\bar{\lambda}$ such that $W(\bar{\lambda}) > 0$, pruning would remove control sequences that do contribute to the minimization. However this is unavoidable in order to manage the growth in computational resources required. Therefore to reduce the errors in the optimal cost function arising from this pruning, we selectively eliminate control sequences with relatively small values of W . This minimizes the impact of pruning on the optimality of the resulting control strategy.

There are several numerical methods [15, 16] that can efficiently solve pruning problems of the form in Eq (33). The more involved mathematical details of the procedure will be addressed in a subsequent article. By adjusting the upper limit on the number of control sequences stored in each set Λ_k , we may arrive at an acceptable tradeoff between speed and accuracy. This approach has enabled a dramatic improvement in the time required to solve problems with 2-qubits (Sec. V), while using standard computing resources

A. Description of computational complexity of the algorithm

We now compare the complexity of the algorithm outlined in this article with that of mesh based solution methods such as in [3]. In the reduced complexity method without pruning, the computational complexity grows as

$$\mathcal{M}^K \quad (35)$$

where \mathcal{M} is the number of elements in the control set and K is the number of time steps in the simulation. For the assumptions on controllability to hold, at-most \mathcal{M} ($\in O(n^2)$) directions of control (i.e. control Hamiltonians) are required. Hence, from Eq (35), the complexity of the algorithm (without pruning) for a simulation of K time steps is

$$[O(n^2)]^K = \text{poly}(n), \quad (36)$$

where $\text{poly}(n)$ denotes a polynomial in n . With pruning, the complexity growth depends on the storage limits chosen in the pruning process. Hence there exists a tradeoff, influenced by these storage limits, between the accuracy of the solution and the growth in complexity of the procedure required to obtain it.

In contrast the computational cost in Eq (36), for mesh based methods with Γ mesh points along each dimension, is

$$\Gamma^{[4^n - 1]} \times O(\Gamma).$$

The two terms in this expression arise from the number of spatial dimensions and number of iterations required respectively.

An important application of the approach described herein is that, given a fixed time horizon T it is possible to efficiently (with respect to the scaling of n) check if a desired gate U can be synthesized within this time. The complexity of the algorithm to perform this check this would be $[O(n^2)]^K$ (*without pruning*) where K is the number of iterations in the algorithm (which is fixed for a given T).

Thus it may be observed that the COD-free approximation technique offers a potentially large order of magnitude reduction in computational complexity.

V. EXAMPLE ON $SU(4)$

We now proceed to apply the theory introduced, to an example on $SU(4)$. The dynamics for this system is given by Eq (1) with $M = 5$ and a control set generated from Hamiltonians of the form $H_k \in \{I \otimes \sigma_x, I \otimes \sigma_z, \sigma_x \otimes I, \sigma_z \otimes I, \sigma_x \otimes \sigma_z\}$ i.e., a set of four 1-body terms and one 2-body term. The associated control directions are sufficient to generate the entire Lie algebra $\mathfrak{su}(4)$, thereby ensuring controllability.

To help highlight the performance improvements of the methods introduced herein, we note that a grid based solution approach with a conservative mesh of 50 points in each of the 15 dimensions of the space $SU(4)$, with a total of 20 iterations over the space and an estimated time of 0.0001 seconds to propagate the cost function via value iteration [3] at each point in the mesh would require 1.69×10^{19} hours and a few terabytes of memory. Using the reduced complexity theory described in the previous section, this problem was solved in 15 hours to yield a solution for the final time horizon problem, with a horizon T of 4 seconds and a discretization step size τ of 0.2 seconds (i.e., 20 propagation steps). The simulation was carried out on a standard desktop computer without any exhaustive efforts to optimize the code. Hence there is a strong potential for further improvements to this procedure.

A. Simulation results

The simulation results obtained from the reduced complexity technique provide the optimal cost function Eq (31) for the gate synthesis problem on the two qubit system. In order to visualize the cost function on the group, we require a mapping between points on the group and points in Euclidean space (as the latter can be easily plotted via conventional graphs). For this purpose we make use of the exponential map from the literature on differential geometry [17, 18]. This is an onto map that takes points in the Lie algebra (the tangent space at the

identity element) to points in the group. As the Lie algebra \mathfrak{g} is isomorphic to the Euclidean space, we can thus obtain a function in the Euclidean co-ordinates at points of interest. The exponential map acts on the algebra of any matrix Lie group as follows

$$\exp(X) := \sum_{j=0}^{\infty} \frac{X^j}{j!}, \quad \forall X \in \mathfrak{g}. \quad (37)$$

For the connected group $SU(2^n)$ the exponential map of the algebra ($\exp(\mathfrak{g})$) generates all of the group (\mathbf{G}) [17, Thm 4.6], thereby ensuring that a valid visualization can be generated for all points in the group.

Thus, in order to visualize the cost function on \mathbf{G} , we evaluate the optimal cost function at a desired set of points on this group that correspond to a 2-dimensional slice of interest in \mathfrak{g} . The plots obtained indicate the approximate optimal cost function at points chosen in the plane of interest (on the algebra) e.g., the $\sigma_x \otimes \sigma_x$ vs $\sigma_y \otimes \sigma_y$ plane (used in this article). The value of the cost function is mapped to the shading used, in order to illuminate the behavior of the function. For instance in Fig 1, regions of darker shading indicate unitaries which are easier (lower cost) to generate while the lighter areas show gates which are costly to synthesize.

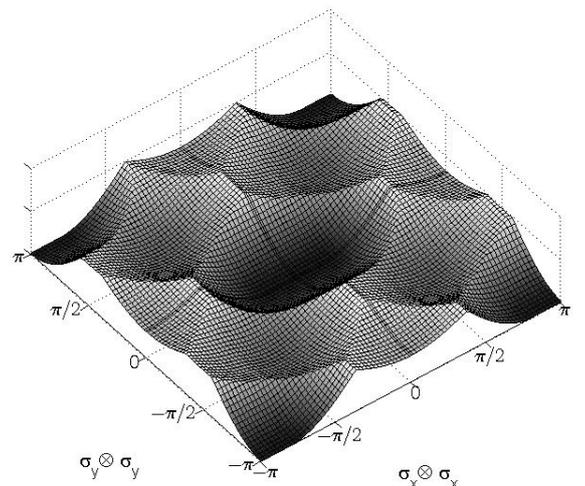


FIG. 1: Plot of the cost function in the plane $\sigma_x \otimes \sigma_x$ vs $\sigma_y \otimes \sigma_y$. Here the cost of applying any available 1-body Hamiltonian is taken to be only slightly less than the cost of generating the 2 body Hamiltonian $\sigma_x \otimes \sigma_x$ (i.e., $\mathbf{r} = 1/1.3$). The lighter areas indicate unitary operations that are harder to synthesize.

In order to understand the effects, on the cost function, of the difficulty in synthesizing the 2-body unitary compared to the 1-body terms (which we denote by the ratio \mathbf{r} of the cost of one body interactions to that of the available two body interaction $\sigma_x \otimes \sigma_x$), simulations were performed that varied \mathbf{r} (using the R matrix in Eq (2))

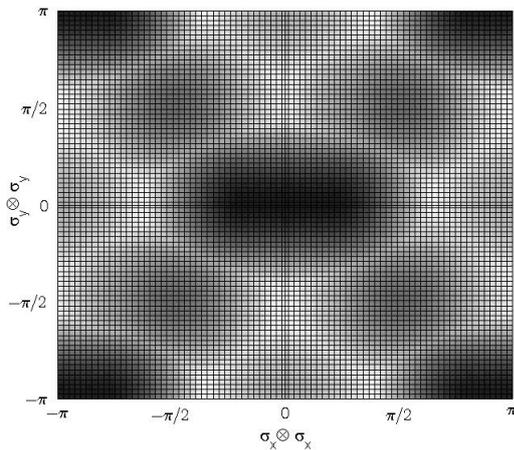
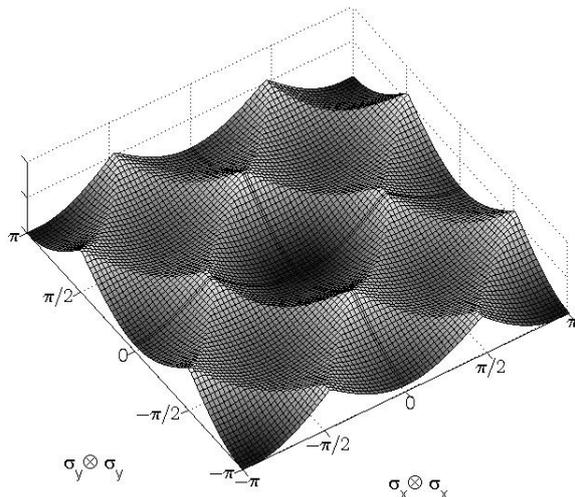


FIG. 2: Top view of the plot in figure 1.

starting from slightly less than one and proceeding upto a value of one-third.

FIG. 3: Plot of the cost function in the plane $\sigma_x \otimes \sigma_x$ vs $\sigma_y \otimes \sigma_y$ with \mathbf{r} set to $1/3$.

The generation of unitaries in the $\sigma_y \otimes \sigma_y$ direction (which is not directly accessible) requires the alternating application (i.e., bracketing operation) of $\sigma_x \otimes \sigma_x$ and one of the available single body Hamiltonians. Hence if $\mathbf{r} = 1/1.3$, then the cost of moving along the $\sigma_y \otimes \sigma_y$ direction is substantially larger than that along the one and two body control directions, leading to an elliptical shape of the level set (as shown in Fig. 2), where the cost increases faster in the $\sigma_y \otimes \sigma_y$ direction compared to the $\sigma_x \otimes \sigma_x$ direction. However as the cost imposed on moving along the $\sigma_x \otimes \sigma_x$ direction is increased, the relative cost of moving along $\sigma_y \otimes \sigma_y$ compared to $\sigma_x \otimes \sigma_x$ decreases, thereby leading to level

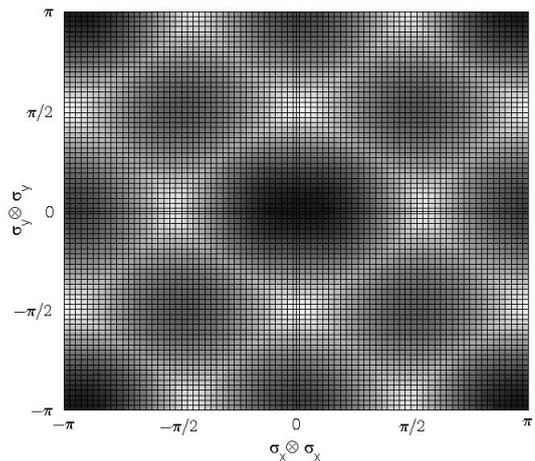


FIG. 4: Top view of the plot in figure 3.

sets that become more circular (Fig. 4). This agrees with analytical results such as in [19]. Note that there are two axis of symmetry in these plots, namely both $\sigma_x \otimes \sigma_x$ and $\sigma_y \otimes \sigma_y$. This arises due to the symmetry in the cost function equations (2) and (9) with respect to the application of controls along either the positive or negative direction of the available control Hamiltonians in the system equation (1).

VI. CONCLUSIONS

In this article we have demonstrated a reduced complexity method may be used to obtain substantial improvements in computational speed in solving optimal control problems arising in closed quantum systems. This technique was used to obtain a numerical solution for an example gate complexity problem in a 2 qubit system. Instead of the curse of dimensionality in the spatial dimension, we now have a much more manageable growth in dimensionality that depends on the number of elements in the discretization of the control set.

The approach outlined in this article can deal with a very general class of problems and can, in principle, be used for systems with any number of spins. Furthermore, the methods described can be extended to other systems of interest. For instance, a control problem on a system with drift can be solved under the current system framework (Eq (1)) by taking the cost of moving along the negative direction of the drift term to be much larger than that in the positive direction.

At present, the techniques outlined yield preliminary solutions whose error bounds, rate of convergence and other properties must be determined via further research. It is hoped that the method introduced (and refinements thereof) would enable the accurate solution of problems on spin systems of larger dimensions than has been pos-

sible until now.

Acknowledgments

S. Sridharan and M.R. James wish to acknowledge the support for this work by the Australian Research

Council. M. Gu acknowledges the support from the National Research Foundation and Ministry of Education of Singapore. W. McEneaney acknowledges support from AFOSR and NSF. The authors would like to thank the reviewer for helpful comments.

-
- [1] P. Shor, *SIAM Review* **41**, 303 (1999).
 - [2] M. Nielsen, M. Dowling, M. Gu, and A. Doherty, *Science* **311**, 1133 (2006).
 - [3] S. Sridharan, M. Gu, and M. James, *Phys. Rev. A* **78**, 52327 (2008).
 - [4] H. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time* (Springer Verlag, Berlin-NY, 1992).
 - [5] M. Bardi and I. C. Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations* (Birkhäuser, Boston, 1997).
 - [6] Crandall, *Trans. AMS* **282**, 487 (1984).
 - [7] W. McEneaney, *Max-plus methods for nonlinear control and estimation* (Birkhäuser, Boston, 2006).
 - [8] W. McEneaney, *SIAM Journal on Control and Optimization* **46**, 1239 (2008).
 - [9] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty, *Phys. Rev. A (Atomic, Molecular, and Optical Physics)* **73**, 062323 (2006).
 - [10] H. Nijmeijer and A. Van der Schaft, *Nonlinear dynamical control systems* (Springer, New York, 1990).
 - [11] R. Bellman, *Dynamic Programming* (Courier Dover Publications, 2003).
 - [12] P. Dupuis, *SIAM Journal on Numerical Analysis* **36**, 667 (1999).
 - [13] W. Fleming and H. Soner, *Controlled Markov Processes and Viscosity Solutions* (Springer Verlag, Berlin-NY, 2006).
 - [14] W. McEneaney, A. Deshpande, and S. Gaubert, in *American Control Conference, 2008* (Washington 2008), pp. 4684–4690.
 - [15] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, MPS-SIAM Series on Optimization (Philadelphia 2001).
 - [16] J. Löfberg, in *Proceedings of the CACSD Conference 2004* (Taipei, Taiwan, 2004).
 - [17] M. R. Sepanski, *Compact Lie groups* (Springer, New York 2007).
 - [18] B. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction* (Springer, New York 2003).
 - [19] M. Gu, A. Doherty, and M. A. Nielsen, *Phys. Rev. A* **78**, 032327 (2008).