Data-driven modeling and control of quantum dynamics

Andy J. Goldschmidt

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Jose Nathan Kutz, Chair
Lukasz Fidkowski
Arka Majumdar

Program Authorized to Offer Degree:
Department of Physics

University of Washington

Abstract

Data-driven modeling and control
of quantum dynamics

Andy J. Goldschmidt

Chair of the Supervisory Committee:
Jose Nathan Kutz
Department of Applied Mathematics

Control is the factor that delineates quantum science from quantum engineering. Model-based optimal control is a modern approach to practical control engineering. It designs control laws using optimization based on dynamical models of the system. If models are perfect, then successful control is realized by simply applying optimal control using the ideal predictions as needed. However, when models aren't perfect, directly applying optimal control produces undesired outcomes due to inaccurate predictions. In this case, we must turn to data-driven methods to improve modeling or control design. This thesis explores two novel ways to incorporate data to control quantum systems that are imperfectly modeled. First, we pursue data-driven modeling, and introduce a physics-informed regression-based approach to learn a model for the quantum control dynamics directly from time series measurements. Second, we pursue data-driven control design, and apply model predictive control to synthesize optimal controls for robust quantum state preparation. In addition to these two novel results, the necessary background in quantum mechanics is provided. Also, data-driven modeling and optimal control are reviewed and contextualized within quantum optimal control. We conclude by offering our perspective on future directions for data-driven approaches to model-based quantum optimal control.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

"Coffee, man, coffee's the drink
For fellows whom it hurts to think:
Look into the Moka pot
To see the world as the world's not.
And faith, 'tis pleasant till 'tis past:
The insight is that 'twill not last."

— A.E. Housman, *A Shropshire Lad*

There is a line in *Thinking, Fast and Slow* by Daniel Kahneman: "Optimistic people play a disproportionate role in shaping our lives." It is presented in a matter-of-fact way–more scientific fact than quip. I would like to thank my advisor, Nathan Kutz, for supplying an abundance of optimisim and coffee during my degree. Nathan is evidence of the disproportionate effect that an optimistic and enthusiastic person can have on the lives of those around them. When recreation and commiseration were the necessary order of the day, I would like to thank my friends for being there. In particular, for going above and beyond the call of duty, thanks to Elliott and Kade. My parents set the foundation for my entire formal and informal education. My successes are very much a direct result of their enduring support as my years in school continued to accummulate. Finally, I owe more than I can communicate here to my wife, Katie. I rely on her advice, patience, and friendship. Thank you for everything.

To Katie, and my parents:

Thank you for always supporting me.

Chapter 1

QUANTUM MECHANICS

> *"[Rolf] Landauer criticized this work very pointedly; he emphasized that computers are made out of physical apparatuses and not out of Hamiltonians, and... it does not constitute a serious model of computation if the imperfections in these apparatuses are not dealt with in the analysis."*
>
> — David DiVincenzo & Daniel Loss [47]

We introduce the mathematical description of quantum mechanics by asking four motivating questions[1]:

1. *How can we encode all our information?* (§1.1: States)

2. *How do we combine different pieces of information?* (§1.2: Compositions)

3. *How do we extract information from our encoding?* (§1.3: Measurements)

4. *How does this information change with time?* (§1.4: Dynamics)

[1] This approach comes courtesy of Daniel Lidar's helpful lecture notes on open quantum systems [120]. He attributes the idea to Michael Nielson and Isaac Chuang's *Quantum Computation* [149].

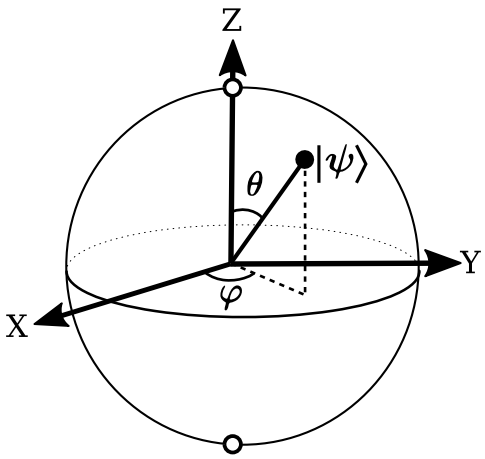


Figure 1.1: The Bloch sphere is a way to represent two-level pure states. Each pure state $|\psi\rangle$ corresponds to a point on the surface of the sphere. By construction, the X, Y, and Z axes point in the directions that correspond to the positive eigenstates of the σ_x, σ_y, and σ_z Pauli matrices.

1.1 States

How can we encode all our information?

1.1.1 Closed systems

In a closed quantum system, we study things isolated from the rest of the world. We call our isolated quantum states *pure states*. They can be understood as a model for idealized quantum systems with negligible interactions outside themselves. Quantum computers attempt to mitigate the effect of the environment as much as possible so that they can calculate using the dynamics of pure states. Of course, as emphasized in the quote introducing this chapter, pure quantum systems are an idealization–however, they are still a good starting point.

A pure quantum state is represented by an element $|\psi\rangle$ of a vector space over the complex numbers. This quantum vector space may have finitely many dimensions d, so that the state space is $\mathbb{C}^d$. More generally, this quantum vector space may be a space of functions, like how a periodic function is fully described by the coefficients of its Fourier series or how a distribution is defined by its value over each position on the real line. In this general case,

the state space is a Hilbert space, $\mathcal{H}$. We usually call quantum state spaces by this general term, Hilbert spaces, although in this chapter we presume the $\mathbb{C}^d$ case unless otherwise noted. Quantum states are equivalent up to constant factors, i.e. rays. This means that the physical part of a quantum state vector comes entirely from the direction it points in the Hilbert space. For convenience, the states are normalized so $\langle\psi|\psi\rangle \equiv 1$, with $\langle\cdot|\cdot\rangle$ the vector inner product. This leaves the representation of the state invariant up to an unphysical phase factor, $|\psi\rangle \equiv e^{i\varphi}|\psi\rangle$.

Working with vector spaces becomes more concrete once we pick a basis. For $\mathbb{C}^d$ choose the standard orthonormal basis, $\{|j\rangle\}_{j=1}^d$, where $|j\rangle$ is the length d vector with 1 at position j and 0 everywhere else. We can expand our quantum state in this basis: $|\psi\rangle = \sum_{j=1}^d a_j |j\rangle$. Now we can start to see how the state vector encodes information about our system: each $|j\rangle$ is a unique possible state for our system, orthogonal to the others. Moreover, the coefficients $a_j \equiv \langle\psi|j\rangle$ encode the probability of the measuring the outcome $|j\rangle$ via the rule $\mathbb{P}[j] \equiv |a_j|^2$. This is called the *Born rule*[2]. Observe that our previous choice of normalization, $\langle\psi|\psi\rangle \equiv 1$, corresponds to a statement about probabilities adding up correctly,

$$1 \equiv \langle\psi|\psi\rangle = \sum_{j=1}^{d} |a_j|^2 = \sum_{j=1}^{d} \mathbb{P}[j]. \tag{1.1}$$

1.1.2 Open systems

Multiple quantum states can be studied simultaneously, but some adjustments are required. If $\{|\psi_k\rangle\}_{k=1}^n$ is an ensemble of pure quantum states with relative contributions weighted by

[2]The Born rule is actually a consequence of assigning probabilities to the various components of our basis. That is, the Born rule emerges if measurements are described by independent projectors like $|j\rangle\langle j|$ as long as there are at least three such outcomes. This is called *Gleason's theorem* (see Box 1.2, Reference [195]).

Example 1: Qubits, Part I

A qubit is an element of a Hilbert space with two dimensions, $|\psi\rangle \in \mathbb{C}^2$. There are four real numbers parameterizing an arbitrary vector. After accounting for the unphysical overall phase and the normalization, we can parameterize the qubit with two angles, so

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)e^{i\phi}|1\rangle\,. \tag{1.2}$$

By defining a polar angle θ and azimuthal angle ϕ on a unit sphere, we can represent every unique state of the qubit with a unique point on the surface of this unit sphere. This representation is called the Bloch sphere (Figure 1.1). It is used to build intuition for two-level quantum dynamics and to help understand single qubit gates in quantum computation.

$w_k \equiv \mathbb{P}[k]$, notice that

$$\begin{aligned}
\mathbb{P}[j] &= \sum_{k=1}^{n} \mathbb{P}[j|k]\mathbb{P}[k] && (1.3)\\
&= \sum_{k=1}^{n} \langle j|\psi_k\rangle \langle\psi_k|j\rangle w_k && (1.4)\\
&= \sum_{k=1}^{n}\sum_{\ell=1}^{d} \langle \ell|j\rangle \langle j|\psi_k\rangle \langle\psi_k|\ell\rangle w_k && (1.5)\\
&= \mathrm{Tr}\left[|j\rangle\langle j|\left(\sum_{k=1}^{n} w_k |\psi_k\rangle\langle\psi_k|\right)\right]. && (1.6)
\end{aligned}$$

Define $\rho \equiv \sum_{k=1}^{n} w_k |\psi_k\rangle\langle\psi_k|$. This is called the *density operator* or *density matrix*, and it generalizes the quantum state to open quantum systems. For a pure state, notice $\rho_k \equiv |\psi_k\rangle\langle\psi_k|$. The mapping from density matrices to mixtures of pure states is not unique.

Density matrices are the set of all trace-one, positive semi-definite[3], linear operators acting

[3] A positive semi-definite operator acting on $\mathbb{C}^d$ is also Hermitian (see Exercise 2.24, Reference [149]).

on a Hilbert space, $\mathcal{H}$. We will denote this alternative state space by $\mathcal{D}(\mathcal{H})$. It is a convex set. The inner product of this state space is the Hilbert-Schmidt inner product, $\mathrm{Tr}[\rho_1\rho_2]$.

Example 2: Qubits, Part II

Example 1 discussed how a pure state could be represented by a point on the surface of the unit sphere. To represent the density matrix, ρ, by a vector, $\vec{v}$, we also need to use all the points on the inside of the unit sphere. Let $\vec{v}_k$ be the representation of the pure state ρ_k on the surface of the Bloch sphere. By the linearity demonstrated in Equation (1.6), the mixed state is $\vec{v} = \sum_{k=1}^{n} w_k \vec{v}_k$. It is bounded above by

$$\|\vec{v}\| = \left\| \sum_{k=1}^{n} w_k \vec{v}_k \right\| \leq \sum_{k=1}^{n} w_k \|\vec{v}_k\| = 1. \tag{1.7}$$

To find a lower bound, consider the pure-state mixture $\frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle$. On the Bloch sphere, this corresponds to averaging the north and south poles,

$$\vec{v} = \frac{1}{2}\vec{Z} + \frac{1}{2}(-\vec{Z}) = \vec{0}. \tag{1.8}$$

This is known as the *totally-mixed state*.

1.2 Compositions

How do we combine different pieces of information?

If we have two Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, we can combine them with the *tensor product*, $\mathcal{H}_1 \otimes \mathcal{H}_2$ (or *Kronecker product*, in the case of $\mathbb{C}^{d_1}$ and $\mathbb{C}^{d_2}$). The tensor product is used, instead of say, a direct sum, because in quantum mechanics we can have states of the combined system that are *entangled*–they cannot be written separately in the basis of the two original spaces. For example,

$$\frac{1}{\sqrt{2}}\left(|0\rangle_1 \otimes |0\rangle_2 + |1\rangle_1 \otimes |1\rangle_2\right) \in \mathbb{C}^2 \otimes \mathbb{C}^2 \tag{1.9}$$

is a valid entangled quantum state known as a Bell state. Said another way, this state cannot be represented by a pair of vectors on the pair of Bloch spheres for the two qubits. Compositions allow us to study interactions between a system and environment–we do so using the space $\mathcal{H}_S \otimes \mathcal{H}_E$.

Consider a lattice of n two-level systems. Each state is a vector in $\mathbb{C}^2$ so

$$\mathcal{H} \equiv \underbrace{\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{\text{n times}} \cong \mathbb{C}^{2^n} \tag{1.10}$$

This is how we would represent a *quantum register* comprised of n qubits with which we might intend to do some quantum computation; notice the dimension grows exponentially with each new qubit. In this context, we often leave out the $\otimes$, writing $|00\rangle = |0\rangle_1 \otimes |0\rangle_2$ for a two-qubit system. This is the quantum *curse-of-dimensionality* (albeit, to call it a curse is to lament some of the advantage offered by a quantum computation; we might even consider this quantum advantage a "blessing of dimensionality" [50]).

1.3 Measurements

How do we extract information from our encoding?

For our introduction to quantum measurement, we appeal to the textbook *Quantum Measurement and Control* by Howard Wiseman and Gerard Milburn [195].

1.3.1 Projective measurements (PVM)

A quantum *observable* is defined to be a Hermitian operator acting on the Hilbert space of the system. Because it is Hermitian, the spectral theorem tell us that an observable $\hat{\Lambda}$ can be written in a basis of projectors, $\hat{\Lambda} = \sum_\lambda \lambda \hat{\Pi}_\lambda$. The projectors may involve some degeneracy, $\hat{\Pi}_\lambda = \sum_{\ell=1}^{L} |\lambda, \ell\rangle\langle\lambda, \ell|$, but the important part is they are orthonormal, like the standard basis we used to decompose the quantum state space in Section 1.1. In a *projective measurement*, the outcome of measuring an operator is one of its eigenvalues, λ. The probability is given by

the Born rule, first mentioned in Section 1.1:

$$\mathbb{P}[\lambda] = \mathrm{Tr}\Big[\hat{\Pi}_\lambda \rho\Big]. \tag{1.11}$$

The posterior state of the system directly after the measurement is given by the *projection postulate*,

$$|\psi_\lambda\rangle = \frac{\hat{\Pi}_\lambda |\psi\rangle}{\sqrt{\mathbb{P}[\lambda]}}, \quad \rho_\lambda = \frac{\hat{\Pi}_\lambda \rho \hat{\Pi}_\lambda}{\mathbb{P}[\lambda]}. \tag{1.12}$$

The quantum state is transformed in a well-defined way by the quantum measurement. This formalism is called projective measurement, or a projection-valued measure measurement (PVM measurement).

Remark 1.3.2. Based on Equation (1.11), the expected value of an observable is

$$\mathbb{E}[\hat{\Lambda}] = \sum_\lambda \lambda \mathbb{P}[\lambda] = \sum_\lambda \lambda \,\mathrm{Tr}\Big[\hat{\Pi}_\lambda \rho\Big] = \mathrm{Tr}\Big[\hat{\Lambda}\rho\Big]. \tag{1.13}$$

1.3.3 POVM measurements

The projective measurements from Section 1.3.1 are often insufficient for real experiments. One major concern is that an experimentalist does not usually measure the quantum system under study in such a direct way. Instead, the experimentalist measures an environment coupled to the system, like an electromagnetic field coupled to an atom. Suppose the experimentalist starts by delineating a pure system and environment, $|S\rangle \otimes |E\rangle$. The experiment runs and the system entangles with the environment under a unitary transformation[4], $U(|S\rangle \otimes |E\rangle)$. Now, let the experimentalist measure the environment to find some result λ. The projection

[4]We mean for U to represent dynamics, see Section 1.4.

postulate (Equation (1.12)) states

$$|S_\lambda\rangle \otimes |\lambda\rangle = \frac{(\mathbb{1} \otimes |\lambda\rangle\langle\lambda|)\ U\ (|S\rangle \otimes |E\rangle)}{\sqrt{\mathbb{P}[\lambda]}} \tag{1.14}$$

$$= \frac{(\mathbb{1} \otimes |\lambda\rangle)\, \langle\lambda|\, U\, |E\rangle\, |S\rangle}{\sqrt{\mathbb{P}[\lambda]}} \tag{1.15}$$

where $\mathbb{1} \otimes |\lambda\rangle$ is a mapping of $|S'\rangle \in \mathcal{H}_S$ to $|S'\rangle \otimes |\lambda\rangle$, and $\hat{M}_\lambda \equiv \langle\lambda|\, U\, |E\rangle$ is a *measurement operator* that only acts on $\mathcal{H}_S$. Therefore, we have a generalized notion of measurement in $\mathcal{H}_S$,

$$|S_\lambda\rangle = \frac{\hat{M}_\lambda\, |S\rangle}{\sqrt{\mathbb{P}[\lambda]}}, \quad \rho_{S,\lambda} = \frac{\hat{M}_\lambda \rho_S \hat{M}_\lambda^\dagger}{\mathbb{P}[\lambda]} \tag{1.16}$$

with

$$\mathbb{P}[\lambda] = (\langle S| \otimes \langle E|)\, U^\dagger\, (\mathbb{1} \otimes |\lambda\rangle\langle\lambda|)\, U\, (|S\rangle \otimes |E\rangle) = \langle S|\, \hat{M}_\lambda^\dagger \hat{M}_\lambda\, |S\rangle \tag{1.17}$$

The process of obtaining this more general measurement on the system using a projective measurement on the environment is known as *Naimark's dilation theorem*. The operator $\hat{E}_\lambda \equiv \hat{M}_\lambda^\dagger \hat{M}_\lambda$ is known as an *effect*. Because they are connected to probabilities, we can say something interesting about effects. Observe

$$1 = \sum_\lambda \mathbb{P}[\lambda] = \sum_\lambda \mathrm{Tr}\left[\hat{E}_\lambda \rho_S\right] \tag{1.18}$$

for any state ρ_S; hence, effects are complete, $\sum_\lambda \hat{E}_\lambda = \mathbb{1}$. This is the only restriction on effects[5]. This formalism based on effects is called *Positive-Operator Valued Measure measurement*. This is almost always presented by its acronym, POVM measurement.

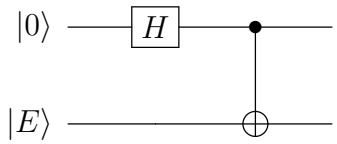
1.4 Dynamics

How does this information change with time?

[5]Effects are usually also noted as being positive semi-definite and Hermitian (see §2.2.6 in Reference [149]); we can quickly confirm that $\langle\varphi|\, E\, |\varphi\rangle = \langle\varphi|\, M^\dagger M\, |\varphi\rangle \geq 0$.

Example 3: Measuring a qubit

The point of this exercise is to see how measurement (of an entangled environment) can affect the state of a qubit. Start with a qubit in $|0\rangle \in \mathcal{H}_S$. Suppose that in order to measure the qubit, we entangle it with a qubit environment in some state $|E\rangle \in \mathcal{H}_E$ by way of the following circuit (shown here is a Hadamard and CNOT gate):



(1.19)

In the computational basis ($|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$), this circuit is equivalent to the action of the matrix

$$\mathbf{B} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{pmatrix}. \tag{1.20}$$

Now, suppose we measure the state of the environment. Looking at Equation (1.15), we can recall the definition of the measurement operator (acting on the system) which results from the process of measuring $|0\rangle$ in the environment: $\hat{M}_0 = (\mathbb{1} \otimes \langle 0|)\mathbf{B}(\mathbb{1} \otimes |E\rangle)$. Consider two cases: $|E\rangle = |0\rangle$, or $|E\rangle = |1\rangle$.

$$\text{Case I: } \langle 0|\,\mathbf{B}\,|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \tag{1.21}$$

$$\text{Case II: } \langle 0|\,\mathbf{B}\,|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} \tag{1.22}$$

Our system state is $|0\rangle \in \mathcal{H}_S$, or $\begin{pmatrix} 1 & 0 \end{pmatrix}^{\mathrm{T}}$. Inspecting Equation (1.15), we see that in Case I, the measurement does not change the state. Meanwhile, in Case II, we see that

measurement has the effect of mapping $\begin{pmatrix}1 & 0\end{pmatrix}^{\mathrm{T}}$ to $\begin{pmatrix}0 & -1\end{pmatrix}^{\mathrm{T}}$. Therefore, depending on the state of the environment, our measurement may or may not have an effect on the system qubit.

1.4.1 Closed systems

The time evolution of a closed quantum system is modeled by unitary operators acting on the system's Hilbert space,

$$|\psi(t)\rangle = U(t,0)\,|\psi(0)\rangle\,. \tag{1.23}$$

The infinitesimal dynamics are generated[6] by the quantum Hamiltonian $\hat{H}$; this can be expressed using the Schödinger equation for either the state or the unitary operators themselves,

$$i\hbar\frac{d}{dt}\,|\psi(t)\rangle = \hat{H}(t)\,|\psi(t)\rangle\,, \quad i\hbar\frac{d}{dt}U(t) = \hat{H}(t)U(t). \tag{1.24}$$

The quantum Hamiltonian can be obtained from a classical Hamiltonian using a procedure called *canonical quantization*, where canonically conjugate variables from Hamilton's equations, like position x and momentum p, are replaced by non-commuting observables, i.e. $[\hat{x},\hat{p}] = i\hbar$. The equivalent of Schrödinger's equation for the density matrix is the quantum Liouville equation,

$$i\hbar\frac{d}{dt}\rho(t) = [H,\rho(t)]. \tag{1.25}$$

Remark 1.4.2. Recall the expression for the expected value of an observable in Equation (1.13). Using the cyclic property of the trace, we can write

$$\mathbb{E}[\hat{\Lambda}] = \mathrm{Tr}\Big[\hat{\Lambda}\rho(t)\Big] = \mathrm{Tr}\Big[U^\dagger\hat{\Lambda}U\rho(0)\Big] \equiv \mathrm{Tr}\Big[\hat{\Lambda}(t)\rho(0)\Big] \tag{1.26}$$

[6]Unitary operators ($U^\dagger U = \mathbb{1}$) preserve probability. To do so, they must be generated ($\dot{U} = AU$) by a skew-Hermitian operator ($A^\dagger = -A$); writing $U(\mathrm{dt}) = e^{A\,\mathrm{dt}}U(0)$, we observe $U(\mathrm{dt})^\dagger U(\mathrm{dt}) = U(0)^\dagger e^{-A\,\mathrm{dt}}e^{A\,\mathrm{dt}}U(0) = \mathbb{1}$.

Placing the time dependence on the observables is called the *Heisenberg picture*, in contrast to placing the time dependence on the state which is called the *Schrödinger picture*.

Remark 1.4.3. In a quantum computer, gates are the unitary transformations acting on the qubit register. Standard one-qubit gates include the Pauli matrices: for example σ_x is a bit flip gate, $\sigma_x : \{|1\rangle \mapsto |0\rangle\,,\ |0\rangle \mapsto |1\rangle\}$. This can be a bit confusing because σ_x is also the x-spin measurement operator. Notice that the Pauli matrices are special because they correspond to both a Hermitian operator and a unitary transformation; equivalently, they have real eigenvalues on the unit circle, i.e. the eigenvalues are ± 1.

1.4.4 Open systems

In the case of a system coupled to an environment, it is not trivial to see what the dynamics of the system might be. Indeed, the first thing we need to understand is how to define the state of the system given $\rho(t) \in \mathcal{H}_S \otimes \mathcal{H}_E$. To start, suppose we have a separable state given by $\rho(t) = \rho_S(t) \otimes \rho_E(t)$. In this case, we can easily trace out the environmental degrees of freedom to find our system state,

$$\mathrm{Tr}_E[\rho_S(t) \otimes \rho_E(t)] = \rho_S(t)\,\mathrm{Tr}_E[\rho_E(t)] = \rho_S(t). \tag{1.27}$$

Moreover, we maintain that the system state should still be defined by $\rho_S(t) \equiv \mathrm{Tr}_E[rho(t)]$ even for entangled systems that cannot be separated into components exclusively in the system or environment. If M_S is a system-only observable, this definition of a system state uniquely provides that

$$\mathrm{Tr}_S[M_S\rho_S(t)] = \mathrm{Tr}_S[M_S\,\mathrm{Tr}_E[\rho(t)]] = \mathrm{Tr}_{SE}[(M_S \otimes \mathbb{1}_E)\rho(t)] = \mathbb{E}[M_S]. \tag{1.28}$$

Even though we have shown that $\rho_S(t)$ can be defined from $\rho(t)$, we still need to address what the dynamics of this system look like; if $\rho_S(t) = \Phi_S(\rho_S(0))$, what can we say about the system-only flow map[7] Φ_S? It turns out that even finding such a map can be impossible

[7]The term for maps like Φ_S is *superoperator*, meaning it acts linearly on operators, $\Phi_S : \mathcal{D}(\mathcal{H}_S) \to \mathcal{D}(\mathcal{H}_S)$.

(see Reference [120], Section VI); however, for most of the dynamics we study, we can show that Φ_S exists and find general rules regarding its structure. Suppose we again start with a system and environment, and make the critical choice that they are initially decoupled such that $\rho(0) = \rho_S(0) \otimes \sum_\nu \nu \left|\nu\right\rangle \left\langle\nu\right|$. We have enumerated the various states in the environment in an eigenbasis. We expect to find the system dynamics by tracing out the environment like in Equation (1.27),

$$\rho_S(t) = \mathrm{Tr}_E[U(t)\rho(0)U^\dagger(t)] \tag{1.29}$$

$$= \sum_\mu \left\langle\mu\right| U(t) \left(\rho_S(0) \otimes \sum_\nu \nu \left|\nu\right\rangle \left\langle\nu\right| \right) U^\dagger(t) \left|\mu\right\rangle \tag{1.30}$$

$$= \sum_{\mu\nu} K_{\mu\nu}(t)\rho_S(0)K_{\mu\nu}^\dagger(t) \equiv \Phi(\rho_S(0)) \tag{1.31}$$

with $K_{\mu\nu}(t) = \sqrt{\nu} \left\langle\mu\right| U(t) \left|\nu\right\rangle$ as the system-only *Krauss operators*. This general expression for the system dynamics is the *Krauss operator sum representation*. It is also known as a *quantum process* or *channel*. Notice the similarity to the generalized measurements and effects in Section 1.3.3. Measurement is a type of quantum dynamics. Axiomatically, a quantum process is any linear map $\mathcal{D}(\mathcal{H}) \to \mathcal{D}(\mathcal{H})$ that is *completely positive* and *trace-preserving*[8] (this pair of properties is often abbreviated CPTP). All CPTP maps can be expressed (though not uniquely) using the Krauss operator sum representation.

1.4.4.1 GKSL equation

Unitary evolution involves a single Krauss operator, $\rho(t) = U(t)\rho(0)U(t)^\dagger$. We know from Section 1.4.1 that unitary evolution is generated by the Hamiltonian. Is there something like a Hamiltonian which acts as a generator for all quantum processes? The answer turns out to be yes. Following Reference [120] Section IX, we consider the short-time expansion of

[8] Again, trace preserving is a statement about the conservation of probability and provides a normalization condition $\sum_j K_j^\dagger K_j = \mathbb{1}$.

Example 4: Error channels acting on qubits

Recall Example 3 where a measurement of an entangled environment could flip a qubit, depending on the state of the environment. Suppose that the state of the environment was 0 with probability p (recall this does not change the system state) and 1 with probability $1-p$ (recall this flips the system state). The dynamics of the system are now

$$\rho_S' = \begin{cases} \rho_S & \text{with prob. } p \\ \sigma_x \rho_S \sigma_x & \text{with prob. } 1-p \end{cases}. \tag{1.32}$$

This is a type of quantum dynamics which captures the idea of random qubit flips. It evokes the bit flip error of classical computing, so we call this type of dynamics an error channel and denote using $\mathcal{E}(\cdot)$. Notice that it is necessary to represent the system dynamics using the Krauss operator sum,

$$\rho_S' = \mathcal{E}(\rho) = p\rho_S + (1-p)\sigma_x \rho_S \sigma_x. \tag{1.33}$$

Observe that this error channel does indeed satisfy the normalization condition, $p\mathbb{1} + (1-p)\sigma_x^2 = \mathbb{1}$.

quantum channels:

$$\rho(\mathrm{dt}) = \rho(0) + \frac{\partial \rho(t)}{\partial t}\Big|_{t=0} \mathrm{dt} + \mathcal{O}(\mathrm{dt}^2) \tag{1.34}$$

$$\rho(\mathrm{dt}) = \sum_\alpha K_\alpha(\mathrm{dt})\rho(0)K_\alpha^\dagger(\mathrm{dt}) \tag{1.35}$$

Define[9] $K_0 \equiv \mathbb{1} + L_0\mathrm{dt}$ and $K_\alpha \equiv \sqrt{\mathrm{dt}}L_\alpha$ for $\alpha \geq 1$. With these definitions in hand,

$$\rho(\mathrm{dt}) = K_\alpha(\mathrm{dt})\rho(0)K_\alpha^\dagger(\mathrm{dt}) \tag{1.36}$$

$$= \rho(0) + \mathrm{dt}(L_0\rho(0) + \rho(0)L_0^\dagger + \sum_{\alpha\geq 1} L_\alpha \rho(0) L_\alpha^\dagger) + \mathcal{O}(\mathrm{dt}^2) \tag{1.37}$$

Decomposing $L_0 = A - iH$ into Hermitian and skew-Hermitian parts, we have

$$\rho(\mathrm{dt}) = \rho(0) + \mathrm{dt}(-i[H,\rho(0)] + \{A,\rho(0)\} + \sum_{\alpha\geq 1} L_\alpha \rho(0) L_\alpha^\dagger) + \mathcal{O}(\mathrm{dt}^2) \tag{1.38}$$

We can use the Krauss operator normalization to find A to 1st order, so

$$= \rho(0) + \mathrm{dt}(-i[H,\rho(0)] + \sum_{\alpha\geq 1} L_\alpha \rho(0) L_\alpha^\dagger - \frac{1}{2}\{L_\alpha^\dagger L_\alpha, \rho(0)\}) + \mathcal{O}(\mathrm{dt}^2) \tag{1.39}$$

To get unitless operators, replace $L_\alpha \to \sqrt{\gamma_\alpha}L_\alpha$. Assuming that initial time was arbitrary (equivalently, that the dynamics can only depend on the current initial condition), we replace our short-time expansion around $t = 0$ with any time t. This plays the role of the Markov

[9] We need one operator to capture the Krauss operator normalization, and at least one other to find something beyond unitary evolution. The factors of time are set so that each operator appears at order dt in the expansion.

assumption for this dynamical system. We now state our final result:

$$\begin{aligned}\frac{d}{dt}\rho(t) = & \underbrace{-\frac{i}{\hbar}[H,\rho(t)]}_{\mathcal{L}_H[\rho(t)]} + \underbrace{\sum_{\alpha\geq 1}|\gamma_\alpha|\left(L_\alpha\rho(t)L_\alpha^\dagger - \frac{1}{2}\{L_\alpha^\dagger L_\alpha,\rho(t)\}\right)}_{\mathcal{L}_D[\rho(t)]} \\ & \equiv \mathcal{L}[\rho(t)].\end{aligned} \tag{1.40}$$

This is the *GKSL equation* (an abbreviation for its namesakes Gorini, Kossakowski, Sudarshan, and Lindblad), or the Lindblad master equation. We call $\mathcal{L}$ the *Lindbladian*, which is known as a superoperator because it acts on operators, $\mathcal{L}:\mathcal{D}(\mathcal{H})\to\mathcal{D}(\mathcal{H})$. The Lindbladian can be decomposed into a unital part and a dissipative part, $\mathcal{L}=\mathcal{L}_H+\mathcal{L}_D$. The unital part is equivalent to the Hamiltonian part of the quantum Liouville equation (Equation (1.25)). The dissipative part captures Markovian effects from the environment. It is worth noting that we can derive the same equation while allowing for time-dependent $H(t)$ and $\gamma(t)\geq 0$ (see Reference [120]).

Chapter 2

QUANTUM CONTROL

> *"The difference between science and technology is the ability to engineer your surroundings to your own ends, and not just explain them. In the Second Quantum Revolution, we are now actively employing quantum mechanics to alter the quantum face of our physical world."*
>
> — Jonathan Dowling & Gerard Milburn [52]

In a control model, our purpose is to modify the natural behavior of a system to our liking. We act on a physical system using whatever forces are at our disposal. Let $\mathbf{x}(t) \in \mathbb{F}^N$ be a state vector over some field $\mathbb{F}$. Perhaps this state vector is a real vector of positions and momenta governed by Hamilton's equations of motion. Alternatively, perhaps the state vector contains the complex coefficients of a finite-dimensional quantum wavefunction. In either case, the state $\mathbf{x}(t)$ may evolve over time on its own accord due to some intrinsic dynamics. Our intent is to act on such a system. We represent our actuation using a time-dependent control function $\mathbf{u}(t) \in \mathbb{R}^M$. If the change in $\mathbf{x}(t)$ is entirely determined by the current state and the current control, then we can write down an ordinary differential equation for the control dynamics:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t). \tag{2.1}$$

Linear control dynamics are the simplest non-trivial class of control dynamics,

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t). \tag{2.2}$$

There is a closed form solution for the state at all times for a given initial condition and control function. The linear case is the place to start when coming up with theoretical concepts

about control. We will come back to it later when we need to introduce basic concepts in control theory.

Arguably the next simplest class of systems is bilinear control dynamics,

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \sum_{j=1}^{M} u_j(t)\mathbf{B}_j\mathbf{x}(t). \tag{2.3}$$

It turns out that quantum control dynamics are bilinear. In particular, the Hamiltonian of a quantum system is often $H = H_0 + u(t)H_I$ with H_0 called a drift or internal Hamiltonian and H_I a control or interaction Hamiltonian. This chapter will follow the emergence of bilinear control dynamics in quantum mechanics for a few representative systems. These systems will be derived in a regime called the *semiclassical approximation*. In this approximation, the quantum degrees of freedom are reserved for the system being controlled, and the fields doing the controlling are kept as classical variables. In the last section of this chapter, we look at a class of classical nonlinear control dynamics from an operator theoretic perspective. We show that this perspective also leads us to bilinear control dynamics.

2.1 Molecules and atoms

The first derivation we will make is the interaction of a quantum particle with a classical electric field. The quantum particle might be an atom or molecule. We will find the interaction term is governed by the coupling of the quantized particle's dipole to the incident electromagnetic field. We will use Reference [41] and Reference [80] to guide us.

To start, let's consider a handful of charged particles in free space. The only fields in this problem are the ones produced by the particles. We will add a subscript s to emphasize system. The classical Hamiltonian for this system combines the kinetic energy of the particles with the energy of the fields produced by the particles. It ends up looking similar to a bunch of charged particles in a potential with a Coulomb (self-)interaction $H_{Coul.} = \sum_j q_j \Phi_s(\mathbf{x}_j)$ in

a potential Φ_s, except for the addition of a radiation term, $H_{Rad.}$, i.e.

$$H_s = \sum_{j=1}^{N} \frac{1}{2m_j} \left(\mathbf{p}_j - q_j \mathbf{A}_s(\mathbf{x}_j)\right)^2 + H_{Coul.} + H_{Rad.} \tag{2.4}$$

The details are explained in the following remark.

Remark 2.1.1. **Derivation of Equation (2.4).** In H_s, the first term is the kinetic energy in terms of canonical momentum, and the second term is the Coulomb potential energy. The latter emerges from the energy of the field produced by the particles. Consider the field energy, $\frac{1}{2}\int d\mathbf{x}\ \left(\varepsilon_0 \mathbf{E}_s^2 + \frac{1}{\mu_0}\mathbf{B}_s^2\right)$. The electric field can be decomposed into two parts, $\mathbf{E}_s = \mathbf{E}_{sL} + \mathbf{E}_{sT}$. The first is a longitudinal part that is irrotational ($\boldsymbol{\nabla} \times \mathbf{E}_{sL} = 0$), and the second is a transverse part that is sourceless ($\boldsymbol{\nabla} \cdot \mathbf{E}_{sT} = 0$). The two parts are orthogonal under reasonable boundary conditions at infinity ($\int d\mathbf{x}\ \mathbf{E}_{sT} \cdot \mathbf{E}_{sL} = 0$). We can transfer the decomposition to the level of the vector and scalar potentials, $\mathbf{E}_{sL} = -(\partial_t \mathbf{A}_s + \boldsymbol{\nabla}\Phi_s)_L$. We choose the Coulomb gauge so $\boldsymbol{\nabla} \cdot \mathbf{A}_s = 0$ is transverse. Hence, $\mathbf{E}_{sL} = -\boldsymbol{\nabla}\Phi_s$. Upon integrating by parts, the longitudinal part of the electric field can be shown to be the Coulomb component of the energy, $H_{Coul.} \equiv \frac{1}{2}\int d\mathbf{x}\ \varepsilon_0 (\mathbf{E}_{sL})^2 = \frac{1}{2}\int \left(\sum_{j=1}^{N} q_j \delta(\mathbf{x} - \mathbf{x}_j\right) \Phi_s(\mathbf{x})$. Notice we have introduced a distribution of point charges. In we are careful and neglect self-energies, $H_{Coul.} = \frac{1}{8\pi\varepsilon_0}\sum_{k\neq j} q_j q_k |\mathbf{x}_j - \mathbf{x}_k|^{-1}$. Finally, define the leftover transverse part to be the radiative component of the energy, $H_{Rad.} \equiv \frac{1}{2}\int d\mathbf{x}\ \left(\varepsilon_0 (\mathbf{E}_{sT})^2 + \frac{1}{\mu_0}(\mathbf{B}_s)^2\right)$. □

Suppose we introduce additional fields produced by some other charges and currents. Suppose further that these new sources are fixed by us and do not otherwise interact. As a result, the external fields (denoted by subscript e) add linearly to the field terms interacting with the particles. This is due to the linearity of Maxwell's equations. The Hamiltonian becomes

$$H_{se} = \sum_{j=1}^{N} \frac{1}{2m_j} \left(\left(\mathbf{p}_j - q_j \mathbf{A}_s(\mathbf{x}_j) - q_j \mathbf{A}_e(\mathbf{x}_j)\right)^2 + q_j \Phi_e(\mathbf{x}_j)\right) + H_{Coul.} + H_{Rad.}. \tag{2.5}$$

Our first approximation is that the particles are close to each other, so the Coulomb

interaction dominates the interaction of the particles and their field. Thus, the contribution of $\mathbf{A}_s$ can be neglected, and the only remaining fields coupled to the particles are the external fields,

$$H_{se} = \sum_{j=1}^{N} \frac{1}{2m_j} \left((\mathbf{p}_j - q_j \mathbf{A}_e(\mathbf{x}_j))^2 + q_j \Phi_e(\mathbf{x}_j) \right) + H_{Coul.} + H_{Rad.}. \tag{2.6}$$

Our second approximation is to apply first quantization, in which we model the system of particles as quantum mechanical and leave external fields as classical variables. This is a semiclassical approximation. Going forward, we intend to restrict ourselves to the Hilbert space of the particles. Notice that the radiative term acts only on the field, a separate Hilbert space from the one which describes the particles. We exchange the canonical variables for operators, and arrive at our quantum Hamiltonian for the particles:

$$\hat{H}_s = \sum_{j=1}^{N} \frac{1}{2m_j} \left((\hat{\mathbf{p}}_j - q_j \mathbf{A}_e(\hat{\mathbf{x}}_j))^2 + q_j \Phi_e(\hat{\mathbf{x}}_j) \right) + \hat{H}_{Coul.}. \tag{2.7}$$

Further simplification is possible using a third approximation, the *long-wavelength approximation*. We suppose that our particles are localized around the origin and that our external fields do not vary across the length scale of the particle system. Consider a gauge transformation $\chi(\mathbf{x}, t) = -\mathbf{x} \cdot \mathbf{A}(\mathbf{0}, t)$ (note we still remain in the Coulomb gauge). Compute:

$$\mathbf{A}'(\mathbf{x}, t) = \mathbf{A}(\mathbf{x}, t) + \boldsymbol{\nabla} \chi(\mathbf{x}, t) = \mathbf{A}(\mathbf{x}, t) - \mathbf{A}(\mathbf{0}, t) \tag{2.8}$$

$$\Phi'(\mathbf{x}, t) = \Phi(\mathbf{x}, t) - \partial_t \chi(\mathbf{x}, t) = \Phi(\mathbf{x}, t) + \mathbf{x} \cdot \partial_t \mathbf{A}(\mathbf{0}, t) \tag{2.9}$$

For simplicity, fix the potential such that $\Phi(\mathbf{0}, t) = 0$. Expanding to first order and applying the long-wavelength approximation,

$$\mathbf{A}'(\mathbf{x}, t) \approx 0 \tag{2.10}$$

$$\Phi'(\mathbf{x}, t) \approx \Phi(\mathbf{0}, t) + \mathbf{x} \cdot \boldsymbol{\nabla} \Phi(\mathbf{0}, t) + \mathbf{x} \cdot \partial_t \mathbf{A}(\mathbf{0}, t) = -\mathbf{x} \cdot \mathbf{E}. \tag{2.11}$$

Define the dipole operator $\hat{\mathbf{d}} \equiv \sum_j q_j \hat{\mathbf{x}}_j$. The particle Hamiltonian is now

$$\hat{H} = \sum_{j=1}^{N} \frac{1}{2m_j} \hat{\mathbf{p}}_j^2 + \hat{H}_{Coul.} - \hat{\mathbf{d}} \cdot \mathbf{E}(\mathbf{0}, t) \tag{2.12}$$

Fix the direction of the electric field and attach a control such that $\mathbf{E}(\mathbf{0}, t) = u(t)\boldsymbol{E}_0$, and define

$$\hat{H}_0 \equiv \sum_{j=1}^{N} \frac{1}{2m_j} \hat{\mathbf{p}}_j^2 + \hat{H}_{Coul.} \tag{2.13}$$

$$\hat{H}_I \equiv -\hat{\mathbf{d}} \cdot \mathbf{E}_0 \tag{2.14}$$

so

$$\hat{H} = \hat{H}_0 + u(t)\hat{H}_I. \tag{2.15}$$

The internal $\hat{H}_0$ describes the structure of the atom or molecule and $\hat{H}_I$ couples the energies.

In the next example, we consider the common case where the multilevel approximation involves only two levels. This gives us a binary representation which can be used to encode information similar to a classical bit. In quantum information, this two-level state is referred to as a qubit.

2.2 Particles with spin

The second example we study is a quantum particle interacting with a classical magnetic field. Recall that the magnetic field interacts with charged particles based on their angular momentum. This can be the orbital angular momentum of an electron bound to a nucleus. In quantum mechanics, this might also be the intrinsic angular momentum of the electron itself–a distinctly quantum property of fundamental particles evocatively referred to as spin. The spin of a fundamental point particle does not emerge out of any rotation or orbital motion. Instead, it is a defining property of the particle. Electrons, protons, and neutrons are examples of spin-$1/2$ particles. The Hilbert space of a spin-$1/2$ particle is intrinsically two-dimensional. Example 6 introduced a situation where we restrict our quantum control

Example 5: Multilevel approximation

Starting from Equation (2.15), consider solving the spatial wavefunction of the bare system. This is found via the time-independent Schrödinger wave equation (with $\hat{\mathbf{x}} \mapsto \mathbf{x}$ and $\hat{\mathbf{p}} \mapsto -i\hbar\nabla$),

$$i\hbar\partial_t\phi(\mathbf{x}, t_0) = H_0\phi(\mathbf{x}, t_0). \tag{2.16}$$

Because H_0 is a linear differential operator it has an eigenbasis $\{\phi_k(\mathbf{x}, t_0)\}$ with corresponding energies $\{\varepsilon_k\}$. The time-dependence is $\phi_k(\mathbf{x}, t_0) = \phi_k(\mathbf{x}, t_0)\exp\{-i\varepsilon_k(t - t_0)\}$. We can expand any wavefunction in this basis so $\psi(\mathbf{x}, t) = \sum_k c_k(t)\phi_k(\mathbf{x}, t_0)$. We can also expand $\hat{H}_I$ in this basis. If we can assume our initial population is limited to a finite range of energies and that the dynamics retains this truncation, we can make the *multilevel approximation* to obtain a bilinear matrix differential equation for a finite list of coefficients,

$$i\hbar\, d_t\mathbf{c}(t) = \mathbf{A}\mathbf{c}(t) + u(t)\mathbf{B}\mathbf{c}(t). \tag{2.17}$$

system to two levels. Spin can couple to the magnetic field like a spatial angular momentum. The result for one particle is a control system of the form

$$\hat{H}(t) = \frac{\gamma}{2}\left(\mathbf{B}_e(t)\cdot\hat{\sigma}\right) \tag{2.23}$$

where $\mathbf{B}_e$ is a time-dependent external magnetic field and $\hat{\sigma} = [\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z]^{\mathrm{T}}$ is a vector containing the Pauli operators.

Spins provide a natural opportunity to consider a coupled lattice or network of particles interacting with an external magnetic field. One experimental realization is a molecule where each constituent ion has a spin as shown in Figure 2.1 This type of system is analogous to a network of coupled qubits on a quantum chip. Coupled spins live in a Hilbert space combined by tensor products, so

$$\hat{\sigma}_z^{(j)} \equiv \mathbb{1}\otimes\cdots\otimes\mathbb{1}\otimes\hat{\sigma}_z\otimes\mathbb{1}\otimes\cdots\otimes\mathbb{1} \tag{2.24}$$

Example 6: Control of two-level dynamics

Consider Equation (2.15) for a single two-level atom with ground and excited states denoted by $|g\rangle$ and $|e\rangle$. For convenience, set the energy difference to be such that $\hat{H}_0 = \frac{\hbar\omega_0}{2}\hat{\sigma}_z$. In this basis, the dipole operator is

$$\hat{\mathbf{d}} = \begin{pmatrix} \langle g|\, q\hat{\mathbf{x}}\,|g\rangle & \langle g|\, q\hat{\mathbf{x}}\,|e\rangle \\ \langle e|\, q\hat{\mathbf{x}}\,|g\rangle & \langle e|\, q\hat{\mathbf{x}}\,|e\rangle \end{pmatrix} = \begin{pmatrix} 0 & d\boldsymbol{\epsilon}_d^* \\ d\boldsymbol{\epsilon}_d & 0 \end{pmatrix} = d(\boldsymbol{\epsilon}_d\hat{\sigma}_- + \boldsymbol{\epsilon}_d^*\hat{\sigma}_+) \tag{2.18}$$

where $\langle e|\, q\hat{\mathbf{x}}\,|g\rangle \equiv d\boldsymbol{\epsilon}_d$ with d the real-valued magnitude and $\langle g|\, q\hat{\mathbf{x}}\,|g\rangle = \langle e|\, q\hat{\mathbf{x}}\,|e\rangle = 0$ by parity[a]. Consider a single mode ω of some driving electric field with polarization $\boldsymbol{\epsilon}_f$ and amplitude modulated by $f(t)$,i.e. $\mathbf{E}(t) = f(t)\left(\boldsymbol{\epsilon}_f e^{-i\omega t} - \boldsymbol{\epsilon}_f^* e^{i\omega t}\right)$ so that the full dipole term is

$$-\hat{\mathbf{d}}\cdot\mathbf{E} = -df(t)(\boldsymbol{\epsilon}_d\cdot\boldsymbol{\epsilon}_f\hat{\sigma}_- e^{-i\omega t} - \boldsymbol{\epsilon}_d\cdot\boldsymbol{\epsilon}_f^*\hat{\sigma}_- e^{i\omega t} + \boldsymbol{\epsilon}_d^*\cdot\boldsymbol{\epsilon}_f\hat{\sigma}_+ e^{-i\omega t} - \boldsymbol{\epsilon}_d^*\cdot\boldsymbol{\epsilon}_f^*\hat{\sigma}_+ e^{i\omega t}). \tag{2.19}$$

This is rather unwieldy; we can simplify by moving to a rotating frame. Let the state counter-rotate according to the isolated effect of the bare Hamiltonian, i.e. $|\tilde{\psi}\rangle \equiv \hat{U}_r\,|\psi\rangle \equiv \exp\{it\frac{\omega_0}{2}\hat{\sigma}_z\}\,|\psi\rangle$. The dynamical operator in the rotating frame can be found using the chain rule,

$$\begin{aligned} i\hbar\, d_t\,|\tilde{\psi}\rangle &= i\hbar\, d_t(\hat{U}_r)\,|\psi\rangle + i\hbar\hat{U}_r d_t\,|\psi\rangle \\ &= \left(i\hbar\, d_t(\hat{U}_r)\hat{U}_r^\dagger + \hat{U}_r H\hat{U}_r^\dagger\right)\hat{U}_r\,|\psi\rangle \equiv H_r\,|\tilde{\psi}\rangle\,. \end{aligned} \tag{2.20}$$

The rotating frame eliminates the bare Hamiltonian, $i\hbar\, d_t(\hat{U}_r)\hat{U}_r^\dagger + \hat{U}_r H_0\hat{U}_r^\dagger = 0$. Before addressing the remaining dipole term $\hat{U}_r(-\hat{\mathbf{d}}\cdot\mathbf{E})\hat{U}_r^\dagger$, the following lemma about rotations of raising and lowering operators will be useful:

Lemma 2.1.1.

$$\exp\left\{i\frac{\phi}{2}\hat{\sigma}_z\right\}\hat{\sigma}_\pm\exp\left\{-i\frac{\phi}{2}\hat{\sigma}_z\right\} = \exp\{\pm i\phi\}\hat{\sigma}_\pm$$

In the rotating frame, two terms in the full dipole term pick up an additive phase. Neglecting these fast-rotating terms is known as the *rotating-wave approximation* (RWA)[b]. Using Lemma 2.1.1 and the RWA, the full dipole contribution is:

$$\begin{aligned}\hat{U}_r(-\hat{\mathbf{d}}\cdot\mathbf{E})\hat{U}_r^\dagger &= -d\,f\big(\boldsymbol{\epsilon}_d\cdot\boldsymbol{\epsilon}_f\hat{\sigma}_- e^{-i(\omega+\omega_0)t} - \boldsymbol{\epsilon}_d\cdot\boldsymbol{\epsilon}_f^*\hat{\sigma}_- e^{i(\omega-\omega_0)t} \\ &\qquad + \boldsymbol{\epsilon}_d^*\cdot\boldsymbol{\epsilon}_f\hat{\sigma}_+ e^{-i(\omega-\omega_0)t} - \boldsymbol{\epsilon}_d^*\cdot\boldsymbol{\epsilon}_f^*\hat{\sigma}_+ e^{i(\omega+\omega_0)t}\big) \\ &\approx -d\,f\left(\boldsymbol{\epsilon}_d\cdot\boldsymbol{\epsilon}_f^*\hat{\sigma}_- e^{i(\omega-\omega_0)t} - \boldsymbol{\epsilon}_d^*\cdot\boldsymbol{\epsilon}_f\hat{\sigma}_+ e^{-i(\omega-\omega_0)t}\right) \\ &= -d\,f\left|\boldsymbol{\epsilon}_d\cdot\boldsymbol{\epsilon}_f^*\right|\,\left(\hat{\sigma}_x\cos(\omega_r t) - \hat{\sigma}_y\sin(\omega_r t)\right) \end{aligned} \tag{2.21}$$

where $\phi \equiv \arg\{\boldsymbol{\epsilon}_d\cdot\boldsymbol{\epsilon}_f^*\}$ and $\omega_r \equiv \omega - \omega_0 + \phi$. If we are careful about our choice of drive ω, we can activate either $\hat{\sigma}_x$ ($\omega_r = 0 \mod \pi$) or $\hat{\sigma}_y$ ($\omega_r = \frac{\pi}{2} \mod \pi$). By superposition, we can even include both, permitting an in-phase control about $\hat{\sigma}_x$ and out-of-phase control about $\hat{\sigma}_y$. Define the control over the envelope of these drives by $a_j(t) = df_j(t)\left|\boldsymbol{\epsilon}_d\cdot\boldsymbol{\epsilon}_{f_j}^*\right|$ where $j \in \{x, y\}$. The full Hamiltonian in the rotating frame within the RWA is

$$\hat{H}_r = \hat{U}_r(t)(-\hat{\mathbf{d}}\cdot\mathbf{E})\hat{U}_r^\dagger(t) = a_x(t)\hat{\sigma}_x - a_y(t)\hat{\sigma}_y. \tag{2.22}$$

[a] The Coulomb potential is symmetric with respect to position, so an expectation value of position will vanish over the full space.

[b] e.g., Reference [80] §3

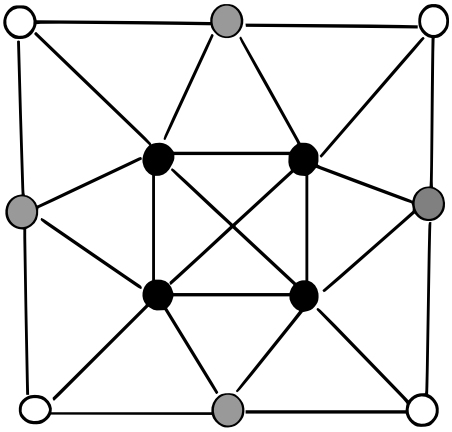


Figure 2.1: (Adapted from Figure 2.8 in Reference [41]). The molecule $Mn12$ involves three different kinds of Manganese ions (shown as black, gray, and white circles). The ions have intrinsic spin and interact pairwise along the illustrated couplings (shown as lines). The system can be described by a Hamiltonian for a network of spins.

acts non-trivially only on the j-th site of the lattice or network. There are many ways that particles with spin might interact among each other. Two of the more common choices for interactions are [41]

Definition 2.2.1. Heisenberg interaction. The *Heisenberg interaction* is given by the Hamiltonian

$$\hat{H}_I = \frac{1}{4}\sum_{j<k} J_{jk}\hat{\sigma}^{(j)} \cdot \hat{\sigma}^{(k)}.$$

where J_{jk} are the coupling constants.

Definition 2.2.2. Ising interaction. The *Ising interaction* is given by the Hamiltonian

$$\hat{H}_I = \frac{1}{4}\sum_{j<k} J_{jk}\hat{\sigma}_z^{(j)}\hat{\sigma}_z^{(k)}.$$

2.3 Circuits

In Section 2.1 and Section 2.2 we used examples based on atoms to motivate dipole and spin control. In this section, we consider circuits which have a quantized energy spectrum

Example 7: Transverse-field Ising model

We can use the Ising interaction to write down a quantum version of the classical Ising model. To make the model quantum, two separate non-commuting observables must be introduced for the on-site and coupling terms of the model (w.l.o.g. we choose $\hat{\sigma}_x$ and $\hat{\sigma}_z$, respectively). Consider nearest-neighbor interactions with fixed on-site energy and coupling strength. We have

$$\hat{H}_{TFI} = -h \sum_j \hat{\sigma}_x^{(j)} - J \sum_j \hat{\sigma}_z^{(j)} \hat{\sigma}_z^{(j+1)} \tag{2.25}$$

which is called the *transverse-field Ising model* (also, the Heisenberg model). The model is bilinear whether we treat the on-site coefficient h or the coupling coefficient J as the control.

like an atom. The material for this section is from Reference [168] and Reference [106]. The advantage of a circuit is they are made of engineered components like capacitors and inductors. The physical properties of these artificial atoms can be tuned to suit our needs. An isolated atom or spin system is entirely described by a few degrees of freedom. It is easy to see that quantum dynamics provides an appropriate description for such a system. Meanwhile, in a mesoscopic circuit there are many electrons in the metal wires. How can we isolate a few degrees of freedom for such a system to make it quantum? The key idea is to use a superconductor where there is a gap separating the many-body ground state of the electrons from the next energy level. If the superconductor is cold enough ($\sim 10mK$), then normal thermal processes cannot excite the system out of this ground state.

Kirchhoff's laws for an LC circuit can be used to obtain the classical Hamiltonian

$$H = \frac{1}{2L}\Phi^2 + \frac{1}{2C}Q^2 \tag{2.26}$$

where Φ is the magnetic flux in the inductor and Q is the charge on the capacitor. A sketch

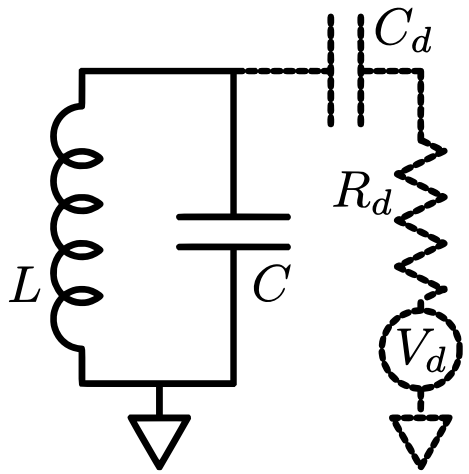


Figure 2.2: (Adapted from Figure D.1 in Reference [168].) An LC circuit is drawn in solid lines. At the right, the dotted lines show how the LC circuit can be coupled to a classical drive line using a parallel capacitor.

of this type of circuit is shown with the solid lines in Figure 2.2. This is the Hamiltonian of a harmonic oscillator with a natural frequency of $\omega_{LC} = 1/\sqrt{LC}$. Therefore, the charge and flux in the circuit oscillate collectively just like a mass moving on a spring. Indeed, the charge and flux are a canonically conjugate pair with Φ the position and Q the momentum. In a normal circuit, there are many electrons which interact to spoil a clean oscillation at this natural frequency. However, for a cold isolated superconductor, there is only one many-body electron ground state for the circuit, and its oscillations between charge and flux are the only degrees of freedom. We therefore quantize this harmonic oscillator in the usual way, i.e.

$$\hat{H} = \frac{1}{2L}\hat{\Phi}^2 + \frac{1}{2C}\hat{Q}^2 = \hbar\omega_{LC}\left(\hat{a}^\dagger\hat{a} + \frac{1}{2}\right) \tag{2.27}$$

where $\hat{a}^\dagger$ and $\hat{a}$ are called the circuit raising and lowering operators, respectively, with

$$\hat{a} \equiv \frac{1}{\sqrt{2\hbar}}\left(\frac{1}{\sqrt{Z_{LC}}}\hat{\Phi} + i\sqrt{Z_{LC}}\hat{Q}\right) \tag{2.28}$$

and $Z_{LC} \equiv \sqrt{\frac{L}{C}}$ is the characteristic impedance. Note $[\hat{a}, \hat{a}^\dagger] = 1$.

Now we just need a way to control our artificial atom. We do not want to directly attach a classical drive line to the circuit. The classical drive line has a resistance (denoted R_d). The bigger this resistance, the quicker the oscillations in the circuit are dampened by energy loss. In terms of the circuit quality factor, $Q_{\text{direct}} = R_d/Z_{LC}$. We need a high circuit quality factor to maintain our carefully constructed quantum coherence. To accomplish this, suppose we attach our drive line in parallel using a capacitor C_d. This lowers the current flowing through the resistor and therefore the energy loss. In particular, in the *weak-coupling limit* we choose a small capacitance ($C_d R_d \ll 1$) so that the quality is larger: $Q_{\text{indirect}} \approx (C/C_d)^2 Z_{LC}/R_d$. Setting $V_d(t)$ as the drive voltage, we can apply Kirchoff's laws to this new circuit. The result is a classical Hamiltonian,

$$H = \frac{1}{2(C+C_d)}\tilde{Q} + \frac{1}{2L}\Phi + \frac{C_d}{C+C_d}V_d(t)\tilde{Q} \approx H_{LC} + \frac{C_d}{C+C_d}V_d(t)Q \tag{2.29}$$

where the new canonical momentum is $\tilde{Q} = (C+C_d)\dot{\Phi} + C_d V_d(t)$ (renormalized by $C_d V_d(t)$) and $\tilde{Q} \approx Q$ for weak-coupling. Finally, we quantize the charge and flux and write everything in terms of raising and lowering operators,

$$\hat{H} = \hbar\omega_{LC}\left(\hat{a}^\dagger\hat{a} + \frac{1}{2}\right) - gV_d(t)\left(\hat{a} - \hat{a}^\dagger\right) \tag{2.30}$$

with $g \equiv i(\sqrt{\hbar/2Z_{LC}})C_d/(C+C_d)$. Suppose we are able to make a multilevel approximation and truncate this system to the first two levels. Then

$$\hat{H}_{\text{trunc.}} = \frac{\hbar\omega_0}{2}\hat{\sigma}_z + gV_d(t)\hat{\sigma}_y \tag{2.31}$$

is our familiar bilinear two-level Hamiltonian where the properties of the system are given by artificial circuit parameters.

Example 8: Transmon qubits

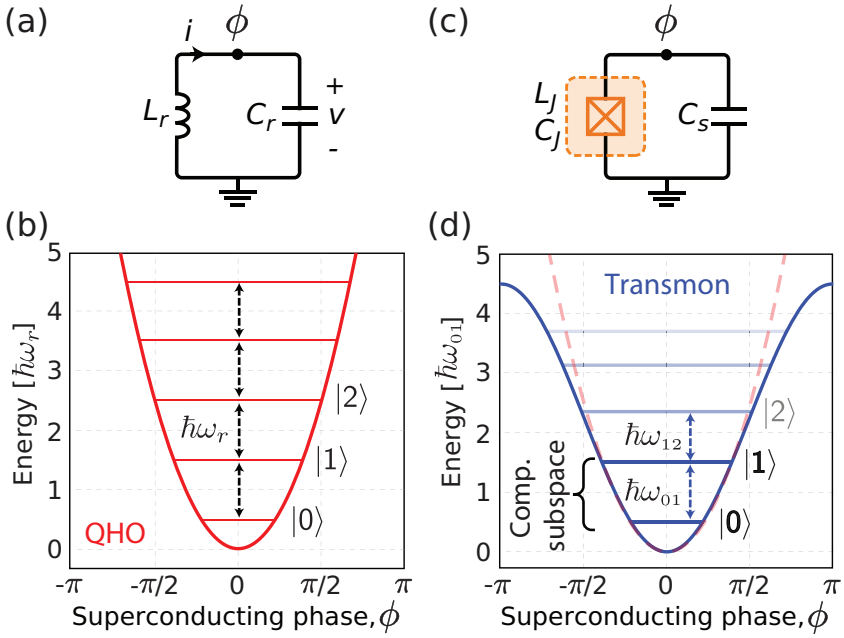


Figure 2.3: (Adapted from Figure 1 in Reference [106]) **(a)** An LC circuit with a single resonance parameterized by L_r and C_r with circuit voltage v and current i is shown. The variable ϕ (*superconducting phase*) is proportional to the quantized magnetic flux. **(b)** The energy levels of the quantum Harmonic oscillator (QHO) for the quantized LC circuit are equally spaced according to the single resonance frequency of the circuit, ω_r. **(c)** The transmon is an LC circuit modified by a Josephson junction (dashed and shaded orange box). This modification introduces a nonlinear inductance L_J among a new set of circuit parameters. **(d)** The nonlinearity changes the shape of the energy potential from a quadratic function (dashed red) to a cosine function (solid blue). Now $\hbar\omega_{01}$ is different from the other values, so the computational subspace (the qubit) can be driven independent of the other energy levels. The amount of isolation is parameterized by the anharmonicity α of this transmon, $\alpha = \hbar\omega_{12} - \hbar\omega_{01}$.

The Harmonic oscillator in this section is a good way to introduce the ideas of a superconducting qubit, but it is not actually feasible. The issue is the truncation of the Harmonic oscillator to just the first two levels. A harmonic oscillator has equally spaced energy levels given by the number of quanta (the eigenstates of the number

operator $\hat{N} = \hat{a}^\dagger \hat{a}$). This means that driving the system at its resonance will involve an infinite number of energy levels. To change this, a nonlinear inductance is introduced by replacing the linear inductor with a nonlinear *Josephson junction* [106]. The truncation in Equation (2.31) is now permissible because the qubit space is individually addressable. For a visualization of this distinction, refer to Figure 2.3. Common qubit frequencies ω_0 are in the range $3 - 6$ GHz. The *anharmonicity* is the term for the offset between the energy difference of the first two levels and the next difference. An infinite anharmonicity means the qubit subspace is perfectly decoupled from the other energy levels. For transmon qubits, the magnitude of the anharmonicity ranges between $100 - 300$ MHz. This value is small enough that it is still possible to have coupling to higher energy levels outside the qubit subspace.

2.4 Control-affine models

In this final section, we change tack slightly and introduce a control-affine nonlinear system,

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \sum_{j=1}^{M} u_j(t)\mathbf{g}_j(\mathbf{x}(t)) \tag{2.32}$$

where $\mathbf{f}, \mathbf{g}_1, \ldots, \mathbf{g}_M$ are smooth nonlinear functions. Our purpose will be to demonstrate that control-affine models are in fact bilinear models, under a suitable transformation of the problem into the language of observables. This will be similar to the mathematics of quantum mechanics, but will emerge from a classical (nonlinear) dynamical system. Koopman theory is reviewed in Reference [27, 154]. In addition, References [76, 159] emphasize the control-affine (bilinear) case.

Introduce the flow map $\mathbf{F}$ as the solution to the autonomous ($\mathbf{u}(t) = 0 \, \forall t$) initial value problem defined by Equation (2.32),

$$\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(0), t). \tag{2.33}$$

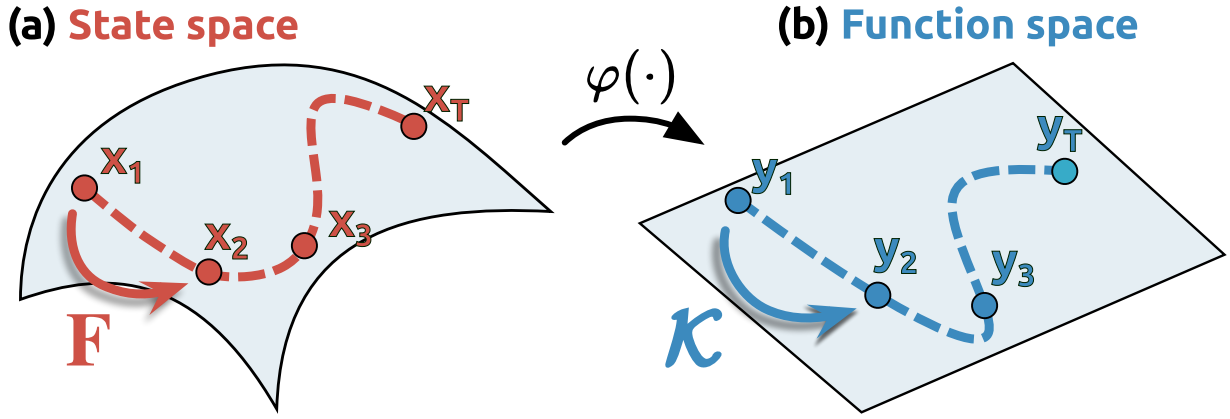


Figure 2.4: (Adapted from Figure 1 of Reference [94]) **(a)** The trajectory of a system with nonlinear dynamics in state space is given by a flow map $\mathbf{F}(\cdot, t)$. Nonlinear dynamics defined over a state space can be exchanged for linear dynamics defined over an infinite dimensional function space using Koopman theory. This is represented by the right arrow overset with an observable function, $\varphi(\cdot)$. **(b)** In function space the dynamics of observables are given by the Koopman operator, $\mathcal{K}$. By judicious choice of observables, it may be possible to find a new state $\mathbf{y}$ in terms of a finite list of observables which evolves according to the linear Koopman dynamics.

An *observable* of the state space is a mapping from the state space of $\mathbf{x}$ to a scalar field like $\mathbb{R}$ or $\mathbb{C}$. The *Koopman operators* $\mathcal{K}_t$ are a family of linear operators parameterized by t that act on observables φ according to

$$\mathcal{K}_t \varphi(\cdot) = (\varphi \circ \mathbf{F}(\cdot, t))(\cdot). \tag{2.34}$$

The Koopman operators advance observables of a dynamical system forward in time. This transforms the nonlinear dynamics defined in the state space into linear dynamics defined over the infinite-dimensional function space spanned by observables. This is visualized in Figure 2.4. For smooth functions of the dynamics, the generator of the Koopman family of operators (often referred to as *the* Koopman operator) is the directional derivative of an

observable along the flow, i.e.

$$\mathcal{K}\varphi(\mathbf{x}(0)) \equiv \lim_{t\to 0}\frac{1}{t}(\mathcal{K}_t - \mathbb{1})\varphi(\mathbf{x}(0)) \tag{2.35}$$

$$= \lim_{t\to 0}\frac{1}{t}(\varphi(\mathbf{F}(\mathbf{x}(0),t)) - \varphi(\mathbf{x}(0))) \tag{2.36}$$

$$\left(\approx \lim_{t\to 0}\frac{1}{t}(\varphi(\mathbf{x}(0) + t\dot{\mathbf{x}}(0)) - \varphi(\mathbf{x}(0)))\right) \tag{2.37}$$

$$= \boldsymbol{\nabla}\varphi(\mathbf{x}(0))\cdot\dot{\mathbf{x}}(0) \tag{2.38}$$

$$= (\mathbf{f}\cdot\boldsymbol{\nabla})\varphi(\mathbf{x}(0)). \tag{2.39}$$

The Koopman generator is like the Hamiltonian in quantum mechanics and the Koopman operators are like unitary matrices[1]. We have that $\frac{d}{dt}\mathcal{K}_t = \mathcal{K}\mathcal{K}_t$. Define $\phi(\mathbf{x},t) \equiv \mathcal{K}_t\varphi(\mathbf{x})$. This is a time-varying observable with reference to a fixed state of the system. Note

$$\frac{\partial}{\partial t}\phi(\mathbf{x},t) = \frac{\partial}{\partial t}\left(\mathcal{K}_t\varphi(\mathbf{x})\right) = \left(\frac{d}{dt}\mathcal{K}_t\right)\varphi(\mathbf{x}) = \mathcal{K}\mathcal{K}_t\varphi(\mathbf{x}) = \mathcal{K}\phi(\mathbf{x},t). \tag{2.40}$$

An initial condition $\phi(\mathbf{x},0) = \varphi(\mathbf{x}(0))$ completes this picture.

This same procedure can be repeated for each of the smooth functions $\mathbf{f}, \mathbf{g}_1, \ldots, \mathbf{g}_M$ to obtain the associated Koopman operators $\mathcal{K}_\mathbf{f}, \mathcal{K}_{\mathbf{g}_1}, \ldots, \mathcal{K}_{\mathbf{g}_M}$. Then

$$\frac{\partial}{\partial t}\phi(\mathbf{x},t) = \left(\mathcal{K}_\mathbf{f} + \sum_{j=1}^{M} u_j(t)\mathcal{K}_{\mathbf{g}_j}\right)\phi(\mathbf{x},t). \tag{2.41}$$

This is a linear equation in an infinite-dimensional function space. Like in the quantum case, we can seek out ways to make a finite-dimensional truncation of this space. It's worth emphasizing that the (bi)linearization we have pursued here through the use of the Koopman operator is global. We traded our nonlinear dynamics defined over our state space for linear dynamics defined over an infinite dimensional function space. We did not rely on Taylor series or other local approximations of our dynamical system.

---

[1]Except in quantum mechanics, there is not an underlying state space $\mathbf{x}$ to come back to.

Chapter 3

DATA-DRIVEN MODELING

> *"[T]he belief that we can start with pure observation alone, without anything in the nature of a theory, is absurd... I tried to bring home the same point to a group of physics students in Vienna by beginning a lecture with the following instructions: 'Take pencil and paper; carefully observe, and write down what you have observed!' They asked, of course, what I wanted them to observe."*
>
> — Karl Popper, *Conjectures and Refutations*

This chapter introduces data-driven modeling, and explores some applications to quantum dynamics. Data-driven modeling (e.g., Steve Brunton and Nathan Kutz's textbook, *Data-Driven Science and Engineering* [28]) is a catch-all term for a new paradigm in the analysis of complex dynamical systems. The spirit of this new paradigm is optimization[1], which we leverage to turn data into meaningful scientific models. This is the shift of the new paradigm: it is the abstraction of model design into algorithms. This abstraction enables modern science and engineering to reach for problems which were previously inaccessible because of their scale or complexity. This new approach was made possible by new technology: data-driven modeling has emerged in the 21st century thanks to the ready availability of computational resources and the ease of data collection and manipulation.

This chapter is a review of some basic concepts and algorithms in data-driven modeling. Throughout this chapter, we contextualize these new concepts using examples from quantum mechanics. Our setting for this chapter will be autonomous dynamical systems–that is, systems evolving under their own dynamics, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$. This precludes any control of the system after the initial condition has been specified. Our focus is on the twin concepts of

[1] Machine learning

model discovery and *model reduction*, which we introduce in Section 3.1. The dynamic mode decomposition (DMD) is an algorithm combining model discovery and reduction, which will remain important through this thesis (in particular, it is integral to the material presented in Chapter 4). We review DMD in Section 3.2. In our discussion, we emphasize why the dynamic mode decomposition is a natural candidate for discovering models for quantum dynamics. We conclude this chapter with Section 3.3 which talks briefly about collecting data from quantum experiments to enable algorithms like DMD.

3.1 Model discovery and reduction

By the 1960s, the control theory based on state-space models of linear time-invariant systems (see Chapter 5) was well-developed. The next step was to implement practical control to meet the engineering needs of the mid-century, including grand challenges like the Apollo space program. This required models that could be implemented on computers. For this type of problem, Rudolf Kálmán [84, 95] and others sought minimal realizations of linear time-invariant systems. A minimal realization was the lowest-rank model obtainable from an existing first-principles description of a dynamical system or, notably, a measurement record of that system. At this time, the focus was linear or approximately linear systems, for which this process is known as *system identification*. Over time, these methods of reduced-order modeling were expanded to more types of systems and eventually incorporated into the modern data-driven paradigm (see [28] §7); today, techniques like the proper orthogonal decomposition (POD) [16], dynamic mode decomposition (DMD) [185], and Koopman mode decomposition (KMD) [142] are widely used in data science and engineering for model discovery of almost every kind of system. As mentioned in the introduction, we will introduce DMD in Section 3.2. We also make a comparison to POD in Section 3.2.3.

The key feature of reduced-order modeling is the ability to obtain an appropriate basis of functions describing the given dynamical system, $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$. In Dirac notation, $\mathbf{x} \mapsto |\psi\rangle$, and we are asking for a basis $\{|\phi_j\rangle\}_{j=1}^{J}$ (usually defined over a spatial coordinate $\phi_j(x) = \langle x|\phi_j\rangle$) such that $|\psi(t)\rangle \approx \sum_{j=1}^{J} c_j(t)\,|\phi_j\rangle$. In the 19th and early 20th century, it was observed that many of the core physical theories (from electromagnetism to quantum

mechanics) were linear and could be described by a particularly useful set of basis functions using the eigendecomposition of the dynamical operator. These theories all fit under the umbrella of the Sturm-Liouville theory, and finding the eigenfunctions of Sturm-Liouville operators leads to all sorts of special polynomials with names like Bessel, Hermite, or Legendre (these are all versions of $\phi_j(x)$, see Example 9). Critically, applying the Sturm-Liouville theory requires working from a first-principles model of the system. Alternatively, the data-driven paradigm (represented by the reduced-order models of DMD, POD, and KMD) obtains basis functions in an optimal way directly from data–no a priori knowledge of the system is necessary.

In the context of PDEs, the resulting projection of the system dynamics onto the data-driven modes, $|\psi(t)\rangle \approx \sum_{j=1}^{J} c_j(t)\,|\phi_j\rangle$, is often referred to as a *Galerkin expansion* [28]. In a Galerkin expansion, we write the state of the system as a sum of static spatial modes, $\phi_j(x) = \langle x|\phi_j\rangle$, and we approximate the dynamics with that of the coefficients $c_j(t)$ within this reduced space. This is a type of *variational method* for dynamical systems [124, 200]. In a variational method, we suppose the state can be written as a function defined over a set of variational (time-dependent) parameters[2], $|\psi\rangle \approx |\varphi(\mathbf{c}(t))\rangle$. The purpose of a variational method is to exchange the dynamics of the state for the dynamics of the parameters, with the rule being to use as few parameters as possible in order to achieve a reduced-order model. In the context of quantum mechanics, the original variational principle was introduced in the early days of the theory by Paul Dirac to derive the Hartree-Fock equations of motion [46]. In 1934–four years later–Yakov Frenkel elaborated many of the unexplained details of Dirac's work [65]. Therefore, the original variational principle in physics is often referred to as the *Dirac-Frenkel variational principle*. The nuclear and particle physics community reframed this work using the principle of least action, leading to a version called the *time-dependent variational principle*. A separate advance came in 1964 when A.D. McLachlan modified the previous approaches to address a practical limitation due to the unphysical overall phase [139].

[2]Observe that the Galerkin expansion is indeed a special case of this approach, $|\varphi(\boldsymbol{c}(t))\rangle = \sum_{j=1}^{J} c_j(t)\,|\phi_j\rangle$.

This version is now known as the *McLachlan variational principle* or Dirac-Frankel-McLachlan variational principle[3]. The critical step in a variational approach to quantum dynamics is the choice of parameterization for the variational state. In keeping with the data-driven paradigm, modern success stories are tensor networks (high-dimensional matrix factorization similar to the two-dimensional singular value decomposition, see Section 3.2.3) [79], universal function approximations like neural networks [31], and–most recently–variational quantum circuits [63, 117, 200]. Within the reduced-order modeling community encompassing the likes of POD, DMD, and the KMD, there has been similar work to tie in tensor networks [100] and deep neural networks [33, 125].

3.2 Dynamic mode decomposition

The dynamic mode decomposition (DMD) is a regression-based framework for linear system identification. That is to say, suppose we are studying phenomena which can be described by a linear, time-invariant, dynamical system given by $\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t)$. DMD provides a way to translate observations of this system, $\mathbf{x}(t_j)$ for $j = 1, 2, \ldots, J$, into a reduced-order model $\mathcal{A}_r$ with $r = \text{rank}[\mathcal{A}_r] \ll \text{rank}[\mathcal{A}]$. There are interpretations of DMD which lend it to be applicable to nonlinear systems[4]. For our purposes, there is a natural connection to quantum systems because the Schrödinger equation is intrinsically linear, $i\hbar\,|\dot{\psi}\rangle = \mathcal{H}\,|\psi\rangle$.

We introduce the standard presentation of the DMD algorithm in Section 3.2.1. In the next part, Section 3.2.3, we emphasize the model reduction component of DMD. A discussion on error bounds associated with DMD model predictions can be found in Section 3.2.4. In Section 3.2.5, we talk about enforcing linear symmetries within DMD.

---

[3]Interestingly, McLachlan's original paper concluded by pointing out challenges he faced when extending the variational principle to open quantum systems. It was only with the recent references [58, 200] that the goal of extending the various variational principles to open quantum systems seems to have been explicitly realized.

[4]The idea is to lift the original state space to a space of observables, similar to Section 2.4 in Chapter 2

3.2.1 Exact DMD

Consider a linear dynamical system

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \mathcal{A}\mathbf{x}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{3.1}$$

in which the dynamics of a state $\mathbf{x}(t) \in \mathbb{R}^n$ is described by a linear operator $\mathcal{A} \in \mathbb{R}^{n\times n}$. In classical mechanics, perhaps the state $\mathbf{x}$ records the density of some material at all points within a two-dimensional domain, and the linear operator $\mathcal{A}$ describes the diffusion of this density. In quantum mechanics, the components of $\mathbf{x}(t)$ might be observable expectation values, like in the three-dimensional Bloch vector or its generalizations. Suppose we collect data from a numerical simulation or experiment (for more, see Section 3.3). In either case, allow the end result to be a sequential measurement record $\{\mathbf{x}(t_1) = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$, $t_1 < t_2 < \cdots < t_m$ of the state of the system at different times. We call each entry of this sequence a *snapshot*. In the diffusion example, our snapshots might be literal photographs of the two-dimensional density. Once collected, the sequential record can be formatted into a pair of offset *snapshot matrices*:

$$\mathbf{X} \equiv \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix}, \qquad \mathbf{X}' \equiv \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{m} \\ | & | & & | \end{bmatrix}, \tag{3.2}$$

so $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{n\times m-1}$. We can actually relax the offset condition, and consider pairs of snapshot matrices [185], replacing $\mathbf{X}'$ with $\mathbf{Y}$,

$$\mathbf{Y} \equiv \begin{bmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{m-1} \\ | & | & & | \end{bmatrix}. \tag{3.3}$$

Algorithm 1 Dynamic Mode Decomposition

INPUT: Snapshot data $\mathbf{X}$, shifted snapshot data $\mathbf{Y}$, target rank r
OUTPUT: DMD modes $\mathbf{\Phi}$, eigenvalues $\mathbf{\Lambda}$

1: **function** DMD($\mathbf{X}$, $\mathbf{Y}$, r)
2: $\quad\widetilde{\mathbf{U}}, \widetilde{\mathbf{\Sigma}}, \widetilde{\mathbf{V}} \leftarrow \mathrm{SVD}(\mathbf{X}, r)$ ▷ Truncated r-rank SVD of $\mathbf{X}$
3: $\quad\mathbf{A}^{\mathrm{DMD}} \leftarrow \mathbf{Y}\widetilde{\mathbf{V}}\widetilde{\mathbf{\Sigma}}^{-1}\widetilde{\mathbf{U}}^{\dagger}$ ▷ Estimate for $\mathbf{A}$, see Equation (3.6)
4: $\quad\widehat{\mathbf{U}}, \widehat{\mathbf{\Sigma}}, \widehat{\mathbf{V}} \leftarrow \mathrm{SVD}(\mathbf{Y}, r)$ ▷ Truncated r-rank SVD of $\mathbf{Y}$
5: $\quad\widehat{\mathbf{A}} \leftarrow \widehat{\mathbf{U}}^{\dagger}\mathbf{A}^{\mathrm{DMD}}\widehat{\mathbf{U}}$ ▷ Projection of $\mathbf{A}^{\mathrm{DMD}}$
6: $\quad\mathbf{\Lambda}, \mathbf{W} \leftarrow \mathrm{EIG}(\widehat{\mathbf{A}})$ ▷ Eigendecomposition of $\widehat{\mathbf{A}}$
7: $\quad\mathbf{\Phi} \leftarrow \mathbf{A}^{\mathrm{DMD}}\widehat{\mathbf{U}}\mathbf{W}$ ▷ DMD modes for $\mathbf{A}$
8: **end function**

The DMD algorithm returns the optimal one-step propagator connecting the pair of snapshot matrices,

$$\mathbf{A}^{\mathrm{DMD}} \equiv \arg\min_{\tilde{\mathbf{A}}} \left\| \tilde{\mathbf{A}}\mathbf{X} - \mathbf{Y} \right\|_F^2 \tag{3.4}$$

where $\|\cdot\|_F$ is the *Froebenius norm*. The corresponding necessary first-order optimality condition is

$$\begin{aligned} &\nabla_{\tilde{\mathbf{A}}} \left(\left\| \tilde{\mathbf{A}}\mathbf{X} - \mathbf{Y} \right\|_F^2 \right) \stackrel{!}{=} 0 \\ \Rightarrow &\tilde{\mathbf{A}}(\mathbf{X}\mathbf{X}^{\mathrm{T}}) \stackrel{!}{=} \mathbf{Y}\mathbf{X}^{\mathrm{T}}. \end{aligned} \tag{3.5}$$

With the *Moore-Penrose inverse* (denoted +), Equation (3.5) can be written

$$\tilde{\mathbf{A}} \stackrel{!}{=} \mathbf{Y}\mathbf{X}^{+} \tag{3.6}$$

This is the typical presentation for the standard DMD algorithm, $\mathbf{A}^{\mathrm{DMD}} \equiv \mathbf{Y}\mathbf{X}^{+}$. It is referred to as exact DMD to distinguish it from other historical presentations of the algorithm [108]. The DMD algorithm does a bit more than just find the propagator defined by Equation (3.5), as we can see in Algorithm 1. Indeed, a critical part of the DMD algorithm is the determination of DMD modes and eigenvalues. In Step 5., notice that the low-rank operator $\widehat{\mathbf{A}}$ used to

construct the DMD modes and eigenvalues is found by projecting $\mathbf{A}^{\mathrm{DMD}}$ onto the rank-r column space of $\mathbf{Y}$. When working with shifted data in which $\mathbf{Y}$ is replaced by $\mathbf{X}'$, the projection is usually done in the column space defined by the original $\mathbf{X}$, i.e. $\widetilde{\mathbf{A}} \leftarrow \widetilde{\mathbf{U}}^{\dagger}\mathbf{A}^{\mathrm{DMD}}\widetilde{\mathbf{U}}$. The rest of the algorithm proceeds in the same way. In either case, the following remark applies:

Remark 3.2.2. **DMD modes are eigenmodes of $\mathbf{A}^{\mathbf{DMD}}$.** The DMD modes defined by $\mathbf{\Phi} \equiv \mathbf{A}^{\mathrm{DMD}}\widehat{\mathbf{U}}\mathbf{W}$ are in fact eigenmodes of the high-dimensional operator $\mathbf{A}_{DMD} \approx \mathbf{A}$. Indeed, we have that

$$\mathbf{A}^{\mathrm{DMD}}\mathbf{\Phi} = \mathbf{\Phi}\mathbf{\Lambda} \tag{3.7}$$

This fact can be easily checked by following the definitions in Algorithm 1. □

From this remark, we conclude that DMD enables an eigenmode expansion $\mathbf{x}_n \approx \mathbf{\Phi}\mathbf{\Lambda}^{n-1}\mathbf{b}$, where $\mathbf{b} = \mathbf{\Phi}^{\dagger}\mathbf{x}_1$ is the projection of the initial state onto the DMD modes. An example of the DMD algorithm applied to the quantum harmonic oscillator can be found in Example 9.

3.2.3 DMD: Model reduction

Notice that in Algorithm 1, we made use of the singular value decomposition (SVD) as a subroutine. This is because the SVD is the solution to the low-rank matrix approximation problem[5]: given an arbitrary $\mathbf{X}$ with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\dagger}$, the solution to the problem of rank-r approximation in the Froebenius norm

$$\begin{aligned} \mathbf{X}_r \equiv \arg\min_{\widetilde{\mathbf{X}}} & \left\| \mathbf{X} - \widetilde{\mathbf{X}} \right\|_F \\ \text{subject to} \quad & \operatorname{rank}[\widetilde{\mathbf{X}}] \leq r \end{aligned} \tag{3.10}$$

is given by the truncated SVD,

$$\mathbf{X}_r = \mathbf{U}[:, :r]\mathbf{\Sigma}[:r, :r]\mathbf{V}[:, :r]^{\dagger}. \tag{3.11}$$

[5]This is known as the Eckhart-Young theorem.

Example 9: DMD for the quantum harmonic oscillator

Consider the one-dimensional Schrödinger wave equation with a harmonic potential, i.e. $\hat{H} = \hat{T} + \hat{V} = \frac{\hat{p}^2}{2m} + \frac{m\omega^2}{2}\hat{x}^2$. Choosing the position basis, we have $\hat{p} \mapsto i\hbar\frac{\partial}{\partial x}$, and so

$$i\hbar\frac{d}{dt}|\psi\rangle = \hat{H}|\psi\rangle \xrightarrow{\psi(x)=\langle x|\psi\rangle} i\hbar\frac{\partial}{\partial t}\psi(x,t) = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{m\omega^2}{2}\right)\psi(x,t). \tag{3.8}$$

Note that we artificially inflate the natural state space of this system by discretization of the wavefunction. Following the Sturm-Liouville theory, we can compute the eigendecomposition of the dynamical operator $(-\hbar^2\partial_{xx}/2m + m\omega^2/2)\phi_j(x) = E_j\phi_j(x)$ to find a more appropriate discrete eigenbasis for this system. We find that the eigenfunctions $\phi_j(x)$ are related to Hermite polynomials. Alternatively, we can take a d-dimensional discretization of this system and numerically simulate every dt for m steps. From this numerical simulation, we can construct the snapshot matrix

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix} = \begin{bmatrix} x_1(\mathrm{dt}) & x_1(2\mathrm{dt}) & \dots & x_1((m-1)\mathrm{dt}) \\ x_2(\mathrm{dt}) & x_2(2\mathrm{dt}) & \dots & x_2((m-1)\mathrm{dt}) \\ \vdots & \vdots & \ddots & \vdots \\ x_d(\mathrm{dt}) & x_d(2\mathrm{dt}) & \dots & x_d((m-1)\mathrm{dt}) \end{bmatrix} \tag{3.9}$$

and $\mathbf{X}'$ (the offset matrix). We set $m = \hbar = \omega = 1$. Then, starting from an arbitrary Gaussian initial state with mean and standard deviation of $1/2$, we apply the DMD algorithm for $d = 100$ ($x \in [-10, 10]$) with $\mathrm{dt} = 0.12$ and $m = 60$ in Figure 3.1. Notice that the DMD modes obtained from the data coincide with the eigenmodes from Sturm-Liouville theory.

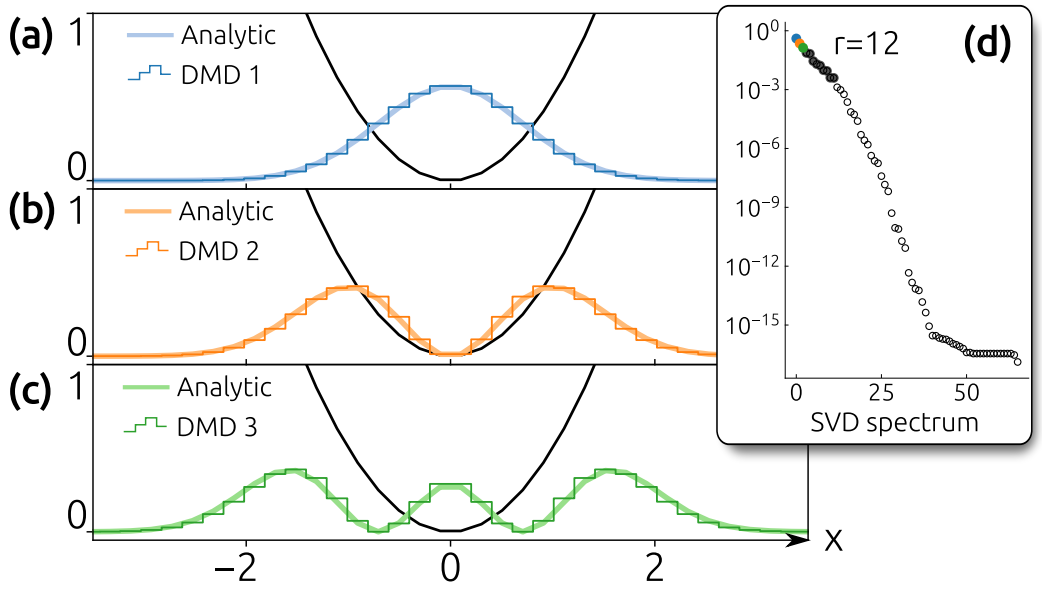


Figure 3.1: Figures **(a)**-**(c)** compare the first three numerical DMD modes computed from Algorithm 1 with the first three analytic eigenmodes of the Schrödinger wave equation for the quantum harmonic oscillator. The quadratic potential is plotted as a black line on each. In Figure **(d)**, the singular values are reported according to their percent contribution to the total variance. The singular values of the selected modes **(a)**-**(c)** have been colored. The total number of singular values is rank-limited by the number of snapshots; in this example, we truncate the DMD singular values at $r = 12$.

where we rely on Python notation to index the matrices.

The proper orthogonal decomposition (POD) [28] is a method for obtaining a reduced-order model which provides a useful comparison to DMD. As a data-driven algorithm, the POD first computes the singular value decomposition of the snapshot matrix from Equation (3.2), so $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger$. Next, the POD modes are defined by the SVD vectors spanning the column space of the snapshot matrix $\mathbf{X}$, i.e. in Python notation, $\mathbf{\Phi}^{\text{POD}} = \mathbf{U}[:, :r]$. Notice that the dynamics have not entered into the calculation except as features in the snapshots; indeed, the POD modes are invariant to permutation of the columns in the snapshot matrix. The property of $\mathbf{\Phi}^{\text{DMD}}$ reported in Remark 3.2.2 does not hold for the POD modes. Example 10 compares the POD and DMD modes for the case of the quantum transverse-field Ising model.

3.2.4 DMD: Error bounds

Suppose that a DMD model is trained with m snapshots–this is the situation in the definition of the snapshot matrices $\mathbf{X}$ and $\mathbf{X}'$ in Equation (3.2). We can ask how accurate the DMD predictions are when using $\mathbf{A}^{\text{DMD}}$ to extrapolate beyond the training set. Consider $n > m$, and let $\mathbf{x}_n^{\text{DMD}} = (\mathbf{A}^{\text{DMD}})^n \mathbf{x}_1$ be the prediction obtained from DMD starting with the initial state. Then [123]

$$\left\|\mathbf{x}_n - \mathbf{x}_n^{\text{DMD}}\right\|_2 \leq \kappa \underbrace{\left\|\mathbf{x}_m - \mathbf{x}_m^{\text{DMD}}\right\|_2}_{\text{Training fit}} + \underbrace{(n-m)\epsilon_m}_{\substack{\text{Accumulation from} \\ \text{numerical integration}}} \tag{3.14}$$

where $\kappa = \kappa(\Phi^{\text{DMD}})$ is the condition number of the DMD modes (the ratio max/min of its singular values), and ϵ_m is a term that improves arbitrarily with a greater density of snapshots as $\mathbf{x}' - \mathbf{A}^{\text{DMD}}\mathbf{x}$ becomes more accurate [104].

3.2.5 DMD: Constraints

Symmetries constrain the kinds of dynamics that a linear operator can induce. Consider a dynamical system, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$. If $\mathbf{A}$ is *skew-symmetric*, i.e. $\mathbf{A}^{\text{T}} = -\mathbf{A}$, then the dynamics are a length-preserving transformation of $\mathbf{x}$. Skew-symmetry is a type of linear constraint on the

Example 10: Spirographs of the transverse-field Ising model

The transverse-field Ising model was first discussed in Example 7. The Hamiltonian

$$\hat{H}_{TFI}(h) = -h\sum_j \hat{\sigma}_x^{(j)} - \sum_j \hat{\sigma}_z^{(j)}\hat{\sigma}_z^{(j+1)} \tag{3.12}$$

acts on a chain of spin-$1/2$ particles (we will assume periodic boundary condition, so technically a ring). The parameter h governs a trade-off between the energy of the nearest-neighbor coupling and the on-site energy. If we consider a chain of 10 spin-$1/2$ particles, then this makes the state space $(\mathbb{C}^2)^{\otimes 10} = \mathbb{C}^{1024}$. We can look for reduced-order models in this large space[a]. The following problem of low-rank approximation for the TFI model was also studied in References [31, 105]: suppose that the ground state of the transverse-field Ising model is obtained at $h = 4$, i.e. $\hat{H}_{TFI}(4)\,|\psi_0\rangle = E_0\,|\psi_0\rangle$. Now, *quench* this system–that is, reduce the parameter to $h = 2$ so that $|\psi_0\rangle$ is no longer in the ground state. Allow this ground state to evolve according to $i\hbar\left|\dot{\psi}_0\right\rangle = \hat{H}_{TFI}(2)\,|\psi_0\rangle$ for 200 steps (setting dt $= 0.01$), and 200 snapshots from this simulation. We can build a reduced order model using the DMD approach (Section 3.2.1) and the POD approach (Section 3.2.3). We truncate the rank of the system to 8, and obtain 8 modes through which we describe the dynamics,

$$|\psi(t)\rangle \approx \sum_{j=1}^{8} c_j^{\text{DMD}}(t)\,|\phi_j^{\text{DMD}}\rangle\,, \qquad |\psi(t)\rangle \approx \sum_{j=1}^{8} c_j^{\text{POD}}(t)\,|\phi_j^{\text{POD}}\rangle\,. \tag{3.13}$$

The 8 modes are sufficient to reproduce 99.9% fidelity with the true simulation in either decomposition. The trajectories of the complex coefficients in Equation (3.13) are shown in Figure 3.2 for 400 steps of the dynamics (i.e. continuing outside the training set). It is evident in Figure 3.2**(a)** that the DMD coefficients have a time-dependence which looks similar to the oscillatory motion associated with eigenvalues–a feature of the DMD algorithm that was discussed in Remark 3.2.2.

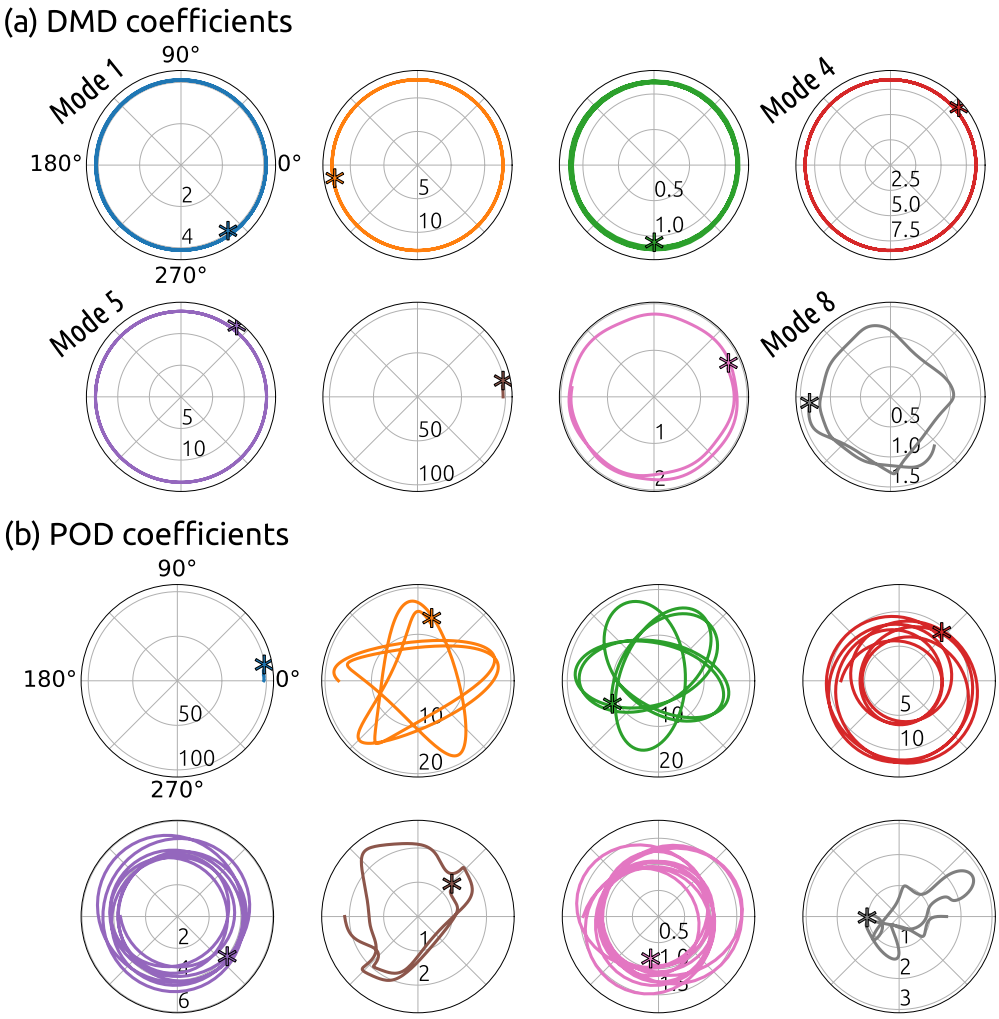


Figure 3.2: The ground state of the transverse-field Ising model at $h = 4$ evolves for 200 steps (dt $= 0.01$) after a quench to $h = 2$. Snapshots are obtained, and a reduced-order model is built with 8 modes using both **(a)** the dynamic mode decomposition (DMD) and **(b)** the proper orthogonal decomposition (POD). The simulation dynamics are expanded using the 8 modes from **(a)** and **(b)**, and plots of the time-dependence of the complex coefficients (now for 400 steps, ending at $*$) are shown. For both **(a)** DMD and **(b)** POD, there is a polar coefficient plotted for each of the 8 modes. The **(a)** DMD coefficients rotate circularly like eigenvalues, while the **(b)** POD coefficients travel sporadically.

---

[a] Intuitively, in finite time with finite energy, the ground state of a system with local interactions can only mix so much [55].

operator **A**. In this section, we look at a way to apply the DMD algorithm so that it respects these kinds of constraints[6]. In order to accomplish our task, we introduce an algorithm that we show is equivalent to exact DMD (Section 3.2.1), but which accommodates these symmetry constraints through the use of *vectorization*. The vectorization of a matrix is a linear transformation that converts a matrix into a vector, $\text{vec} : R^{m\times n} \to R^{mn}$. Vectorization is defined to be column-major or row-major, depending on the stacking priority imposed. For example, Python uses row-major vectorization in NumPy's ndarray.flatten: each row of an input array is sequentially concatenated to form one long row. These operations are interchangeable, as applying a transpose or ndarray.reshape operation to the long row produces the desired column vector. In Dirac notation familiar to physicists, vectorization is the familiar isomorphism for when a bra is flipped to a ket, $|j\rangle\langle k| \cong |j\rangle \otimes |k\rangle$. We restate a well-known property of vectorization (see Lemma 4.3.2 in [86]) here:

Lemma 3.2.1. For any linear transformation, $L : R^{m\times n} \to R^{p\times q}$, there exists a unique matrix $K(L) \in R^{mn\times pq}$ such that $\text{vec}(L(B)) = K(L)\text{vec}(B)$ for any $B \in R^{m\times n}$.

In particular, for the case of left and right matrix multiplication we have

$$\text{vec}(ABC) = \left(A \otimes C^{\text{T}}\right) \text{vec}(B) \tag{3.15}$$

for which we have committed to row-major vectorization to match Python. Observe that $\|B\|_F^2 = \|\text{vec}(B)\|_2^2$ for any $B \in R^{m\times n}$. Therefore, the DMD algorithm defined in Equation (3.4) can be restated as

$$\text{vec}(\mathbf{A}^{\text{DMD}}) \equiv \arg\min_{\text{vec}(\tilde{\mathbf{A}})} \left\| \left(\mathbb{1} \otimes \mathbf{X}^{\text{T}}\right) \text{vec}(\tilde{\mathbf{A}}) - \text{vec}(\mathbf{Y}) \right\|_2^2. \tag{3.16}$$

Remark 3.2.6. **Kronecker commutation matrix.** We digress to remark on another useful consequence of Lemma 3.2.1. The transpose of a matrix is a linear operation, so Lemma 3.2.1

[6]Notable adjacent effort in this direction has been reported in Reference [9] under the distinction of *physics-informed DMD*.

guarantees the existence of a unique matrix $K_{mn} \in R^{mn \times mn}$ such that $\text{vec}(A^{\text{T}}) = K_{mn}\text{vec}(A)$ for any $A \in R^{m \times n}$. Sometimes the matrix K_{mn} is called the *commutation matrix* because it commutes the Kronecker product: if $A \in R^{m \times n}$ and $B \in R^{p \times q}$, then

$$(A \otimes B)\, K_{nq} = K_{mp}\, (B \otimes A)\,. \tag{3.17}$$

□

Returning to the main thread, vectorization can be used to enforce symmetry constraints in DMD through the use of *linear structures* [1]. This approach will also allow for sparsity constraints on the solution without breaking the symmetry. Define *half-vectorization*, $\text{vech} : R^{n \times n} \to R^{n(n+1)/2}$, as vectorization after eliminating the super-diagonal entries from the result. For example, if $A \in R^{2 \times 2}$ then $\text{vech}(A) = [a_{11}, a_{21}, a_{22}]^{\text{T}}$. Concomitant with the half-vectorization operator is the *duplication matrix*: the unique $D_n \in R^{n^2 \times n(n+1)/2}$ such that $D_n\text{vech}(A) = \text{vec}(A)$.

Half-vectorization allows us to insert a symmetric constraint within the vectorized DMD algorithm (Equation (3.16)); the modified optimization is

$$\text{vech}(\mathbf{A}^{\text{DMD}}) \equiv \arg \min_{\text{vech}(\tilde{\mathbf{A}})} \left\| \left(\mathbb{1} \otimes \mathbf{X}^{\text{T}}\right) \mathbf{D}_n \text{vech}(\tilde{\mathbf{A}}) - \text{vec}(\mathbf{Y}) \right\|_2^2. \tag{3.18}$$

Notice that adding regularization to this optimization will not affect the symmetry. Even if regularization is added, the DMD operator remains symmetric by construction.

The idea of half-vectorization and duplication can be generalized. Suppose $A \in R^{m \times n}$ is a generic matrix with $mn - s$ linear relationships among its entries. Suppose further all those relationships are linearly independent. There exists a vector function $\psi : R^{m \times n} \to R^s$ and a matrix $P \in R^{mn \times s}$ such that $P\psi(A) = \text{vec}(A)$. We call the set of all matrices that satisfy the provided constraints P a *linear structure*, denoted $\mathcal{L}(P) \equiv \{A \in R^{m \times n} : \text{vec}(A) \in \text{col}(P)\}$. Any linear structure can be imposed on $\mathbf{A}^{\text{DMD}}$ using the vectorized DMD algorithm like in Equation (3.18).

3.2.6.1 DMD: Unitary constraints

In Section 3.2.5, we introduced a framework which captured a large variety of constraints which we might want to impose on the DMD operator. Missing from this list is a way to make sure that the DMD operator is unitary, i.e. $\|\mathbf{y}\|_2 = \|\mathbf{A}\mathbf{x}\|_2$, or $\mathbf{A}^\dagger\mathbf{A} = \mathbb{1}$. This is a nonlinear constraint, so it does not fit into our previous framework. Instead, we accomplish our unitary goal by projecting the DMD operator to the nearest unitary $\mathbf{P}$ in the operator norm. That is, we seek a solution to the optimization problem

$$\arg\min_{\mathbf{P}} \quad \left\|\mathbf{P} - \mathbf{A}^{\mathrm{DMD}}\right\|_F \tag{3.19}$$

$$\text{subject to} \quad \mathbf{P}^\dagger\mathbf{P} = \mathbb{1}. \tag{3.20}$$

This can be solved using the familiar SVD; if $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger$, then $\mathbf{P} = \mathbf{U}\mathbf{V}^\dagger$ is the nearest unitary matrix. To see this, observe that

$$\|\mathbf{P} - \mathbf{A}\|_F = \left\|\mathbf{U}(\mathbb{1} - \mathbf{\Sigma})\mathbf{V}^\dagger\right\|_F = \|\mathbb{1} - \mathbf{\Sigma}\|_F. \tag{3.21}$$

The idea of applying a constraint by projecting our solution onto the constraint space is a very general idea, but in this particular case, the combination of the unitary projection and the original DMD optimization in Equation (3.4) is a manifestation of the original *Procrustes problem*; such problems are discussed further in Reference [9].

3.3 Collecting data

The essential input to a data-driven algorithm is data from simulation or experiment. For example, consider the snapshot matrices defined in Equation 3.2. In the case of quantum experiments, we have to abide by the measurement rules of quantum mechanics (see Chapter 1, Section 1.3). Recall that in quantum systems, measurement is a dynamical process that changes the state of the system. In particular, a projective measurement of the quantum state necessarily projects the state of the system into an eigenstate of the observable after

the measurement. To obtain sufficient statistics on the expectation value of an observable, many repeat experiments are needed. Then, the expectation values of many observables can be combined to offer a snapshot of the full system[7]. In this way, a quantum experiment is rather unlike a classical one. Indeed, it is possible for a classical snapshot to be an actual photograph of a system in motion–we might take a picture of a swinging pendulum shuffling on a track in our lab [92]. The classical system obtains the entire state vector with one cheap measurement. However, there are other cases where measuring a classical system requires placing a sensor at some specific location, like when a weather service deploys a buoy in the open ocean. The sensor takes measurements of a single local observable, and this information must be combined to offer a glimpse of the full snapshot [133].

3.3.1 Quantum state tomography

In quantum computing, the process of collecting a snapshot of a quantum system has come to be known as *quantum state tomography* [39, 49, 88]. There are a few different approaches, such as maximum likelihood estimation (MLE), Bayesian likelihood estimation (BLE), and linear regression estimation (LRE). In this section, we review the LRE, which has the most efficient numerical algorithm for state reconstruction [49].

In quantum state tomography, the goal is to obtain an estimate of the density matrix, $\rho \in \mathcal{D}(\mathcal{H})$ (see Chapter 1, Section 1.4.4). Naively, this means assembling a set of Hermitian observables in $\mathcal{L}(\mathcal{H})$ (the linear operators acting on the Hilbert space) which are linearly independent and span the space $\mathcal{D}(\mathcal{H}) \subset \mathcal{L}(\mathcal{H})$. For n qubits, $\mathcal{H} = (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$ (see Chapter 1, Section 1.2), and we can identify the linear operators with the matrices $\mathcal{L}(\mathcal{H}) = \mathbb{C}^{2^n \times 2^n}$. That means we are looking for a set of observables $\{\hat{\Omega}_j\}_{j=1}^{2^{2n}}$ that are orthonormal; with Hilbert-Schmidt inner product, this means $\mathrm{Tr}\left[\hat{\Omega}_j \hat{\Omega}_k\right] = \delta_{jk}$. Now, we can decompose our state ρ by expanded it in this basis[8], $\rho = \sum_{j=1}^{2^{2n}} \mathrm{Tr}\left[\hat{\Omega}_j \rho\right] \hat{\Omega}_j$. We can leverage the fact that

[7] Think of the qubit's Bloch vector, which in the open-system case requires specifying the expected value of the three Pauli spin operators.

[8] This generalizes vectorization (see, e.g. Section 3.2.5). To see this, consider making the choice that $\hat{\Omega}_j = |j\rangle\langle j|$. When applying vectorization to the density matrix and other operators in $\mathcal{L}(\mathcal{H})$, the term

$\mathrm{Tr}[\rho] = 1$ to eliminate one degree of freedom; identify $\hat{\Omega}_{2^{2n}} = \mathbb{1}/2^n$ and demanded that the remaining observables are traceless and orthonormal[9], so

$$\rho = \frac{\mathbb{1}}{2^n} + \sum_{j=1}^{2^{2n}-1} \mathrm{Tr}\Big[\hat{\Omega}_j \rho\Big] \hat{\Omega}_j \equiv \frac{\mathbb{1}}{2^n} + \mathbf{x}^{\mathrm{T}} \hat{\mathbf{\Omega}}. \tag{3.22}$$

We have now identified the state $\rho(t)$ with a time-dependent vector of coefficients, $\mathbf{x}(t)$. This is a generalization of the Bloch vector and is known as the *coherence vector* [5, 83]. Of course, a generic Hermitian observable $\hat{\Lambda}_k$ can also be expanded in this basis,

$$\hat{\Lambda}_k = \sum_{j=0}^{2^{2n}-1} \mathrm{Tr}\Big[\hat{\Omega}_j \hat{\Lambda}_k\Big] \hat{\Omega}_j \equiv m_{0,k} \frac{\mathbb{1}}{2^n} + \mathbf{m}_k^{\mathrm{T}} \hat{\mathbf{\Omega}}. \tag{3.23}$$

Now, we can express the expected value of our observable as an inner product between coefficient vectors,

$$y_k = \mathrm{Tr}\Big[\hat{\Lambda}_k \rho\Big] = \frac{m_{0,k}}{2^n} + \mathbf{m}_k^{\mathrm{T}} \mathbf{x}. \tag{3.24}$$

Define $b_k \equiv m_{0,k}/2^n$. Now, the expected outcome of measuring a set of m observables, $\{\Lambda_k\}_{k=1}^m$, can be expressed as a linear equation,

$$\mathbf{y} = \mathbf{b} + \mathbf{M}\mathbf{x}. \tag{3.25}$$

The $\mathbf{y}$ are the known expectation values from measuring our system, and the coherence vector $\mathbf{x}$ is the unknown state. This can be solved with the usual least-squares optimization[10] so

$$\mathbf{x}_{\mathrm{LRE}} \equiv \arg\min_{\tilde{\mathbf{x}}} \|\mathbf{b} + \mathbf{M}\tilde{\mathbf{x}} - \mathbf{y}\|_2. \tag{3.26}$$

---

superoperator is often used by physicists, with the notation $|\rho\rangle\rangle$ for the vectorized density matrix.

[9] The existence of such operators is clear by considering generalizations of the Pauli matrices–indeed, this is simply asking for a basis of the Lie algebra $su(2^n)$, see Chapter 5.

[10] There is no guarantee that the estimate $\mathbf{x}_{LRE}$ in Equation (3.26) actually corresponds to a physical density matrix, $\rho \in \mathcal{D}(\mathcal{H})$ [177]. A physical state is trace-one and Hermitian; both are linear constraints that can be expressed using Lemma 3.2.1. Hence, the closest physical state to the estimate ρ_{LRE} can be

All extensions and adaptations of this least squares problem using sparse sensing, random matrix theory, neural networks, and so on can now be directly applied in this setting.

---

found by linear programming, i.e.

$$\begin{aligned} \arg\min_{\rho} \quad & \|\rho_{\mathrm{LRE}} - \rho\|_F \\ \text{subject to} \quad & \mathrm{Tr}[\rho] = 1 \\ & \rho^\dagger = \rho \end{aligned}$$

or, equivalently,

$$\begin{aligned} \arg\min_{\tilde{\mathbf{x}}} \quad & \|\mathbf{b} + \mathbf{M}\tilde{\mathbf{x}} - \mathbf{y}\|_2 \\ \text{subject to} \quad & \mathbf{C}\tilde{\mathbf{x}} = \mathbf{0} \end{aligned}$$

Chapter 4

BILINEAR DYNAMIC MODE DECOMPOSITION FOR QUANTUM CONTROL

This work is a presentation of Reference [72]. As such, this material is mostly self-contained.

Abstract

Data-driven methods for establishing *quantum optimal control* (QOC) using time-dependent control pulses tailored to specific quantum dynamical systems and desired control objectives are critical for many emerging quantum technologies. We develop a data-driven regression procedure, *bilinear dynamic mode decomposition* (biDMD), that leverages time-series measurements to establish quantum system identification for QOC. The biDMD optimization framework is a physics-informed regression that makes use of the known underlying Hamiltonian structure. Further, the biDMD can be modified to model both fast and slow sampling of control signals, the latter by way of stroboscopic sampling strategies. The biDMD method provides a flexible, interpretable, and adaptive regression framework for real-time, online implementation in quantum systems. Further, the method has strong theoretical connections to Koopman theory, which approximates nonlinear dynamics with linear operators. In comparison with many machine learning paradigms, minimal data is needed to construct a biDMD model, and the model is easily updated as new data is collected. We demonstrate the efficacy and performance of the approach on a number of representative quantum systems, showing that it also matches experimental results.

4.1 Introduction

Quantum optimal control (QOC) is a comprehensive mathematical framework for quantum control in which time-dependent control pulses are tailored to specific quantum dynamical systems and desired experimental objectives [23]. QOC algorithms are critical for emerging

new quantum technologies in scientific and engineering disciplines, including computing, communications, simulation and sensing [127]. For example, algorithm construction in the gate-set model of quantum computation is accomplished by concatenation from a universal set of quantum logic gates; however, QOC can create more accurate implementations of commonly used subroutines by leveraging optimal control [173, 197, 201]. Despite the unique challenges of quantum theory [5, 48], standard model-based control optimization procedures have been developed and refined [23, 51, 54, 97]. Many of the seminal innovations in QOC were developed for applications in *Nuclear Magnetic Resonance* (NMR) [68]; today there are many numerical methods and tools useful in a wide range of QOC application areas [11, 85, 91]. In QOC, practical control design ultimately depends on an experimentally-accurate model of the governing quantum dynamics and the action of controls. In this manuscript, we introduce the data-driven regression framework known as *dynamic mode decomposition* (DMD) to construct control models directly from time-series measurement data. Our DMD method is tailored to the bilinear structure of quantum control dynamics; moreover, it is a completely data-driven approach that can accommodate both fast and slow (stroboscopic) sampling of control signals.

Many data-driven methods for characterizing quantum devices have been developed under a variety of modeling assumptions [56]. The quantum-information community often assumes a quantum-process model of the dynamics. In this case, a fixed quantum process is treated as a *black box* and is experimentally characterized using *quantum process tomography* (QPT) [20, 36, 140, 145, 171]. An alternative to the QPT black box is *Hamiltonian tomography* which attempts to identify the generator of the dynamics by connecting the Hamiltonian parameters to features of the observed dynamics [40, 66, 172]. Our approach is similar to Hamiltonian tomography, but we treat control as an exogenous parameter in order to simultaneously identify separate generators for the drift and control dynamics. Therefore, our approach directly addresses the needs of QOC.

The challenges posed by Hamiltonian tomography are well-known [56, 172]. Statistical difficulties can arise when solving inverse problems involving data. The generated dynamics

are a nonlinear function of the Hamiltonian parameters being sought. There is also the *curse-of-dimensionality*: computational costs grow exponentially with respect to the quantum degrees of freedom. Inspiration can be taken from machine learning methods because they have successfully circumvented many difficulties with high-dimensional and nonlinear data sets through principled construction of low-dimensional embedding spaces. In quantum dynamics, reinforcement learning methods have been used for quantum control design [30, 150, 156], neural networks have been used for finding low-rank embeddings of quantum states and processes [157, 184, 198], and the efficiencies of Bayesian methods [190, 194] have been studied. Our new approach is closely connected to earlier work on the quantum application of linear system identification [203].

Data-driven system identification is a scalable, accurate, and flexible modeling paradigm ubiquitous throughout science and engineering. System identification algorithms use abundant data collection to automate modeling tasks–often in frameworks amenable to model-free control design. Koopman theory for classical system identification and control is one paradigm that even uses the operator-theoretic language developed for quantum dynamics [101, 102, 141]. In Koopman theory, nonlinear classical dynamics are described by the linear action of Koopman operators on an infinite-dimensional Hilbert space of observables. The action on observables makes the operators amenable to general data analysis. The DMD algorithm is a data-driven regression that makes a finite-dimensional approximation to the Koopman operator [185]. DMD avoids the statistical issues associated with inverse problems by finding low-rank, interpretable feature spaces in the data. The algorithm can accommodate physical constraints, delay-embeddings, and multi-scale dynamics [8, 38, 109]. With high-dimensional data DMD can make use of tensor decompositions [71, 100] and compressive sampling [10, 26]. Moreover, the DMD algorithm has been successfully adapted to provide equation-free control design for non-autonomous dynamical systems [75, 126, 159, 163]. In the present work, we show how the DMD system identification framework can be applied to the study of quantum control via a bilinear DMD algorithm (biDMD) [75, 159]. We study the biDMD algorithm as an approach to synthesize numerical optimal control and experimentation. In addition to the

innovation of the biDMD algorithm, we also show how it can be used when sampling is fast or slow relative to the applied control signals–the latter by stroboscopic sampling. The biDMD architecture is applied in several examples in a promising demonstration of the possibility of QOC using time-series data alone.

The first two sections serve as background. Section 4.2 reviews quantum dynamics and recalls the connection with Koopman-von Neumann classical mechanics. Section 4.3 reviews the DMD algorithm. In Section 4.3.3, we introduce biDMD for quantum control. An example is provided for the fast-sample limit where the data record fully resolves the dynamics. Next, Section 4.4 explores biDMD in the opposite limit of stroboscopic sampling. In this context, natural interpretations of DMD using Floquet theory and treatments of biDMD from the perspective of average Hamiltonian theory are discussed with examples.

4.2 Quantum dynamical systems

The first complete formulation of quantum dynamical systems occurred in the 1930s with the operator-theoretic perspective of the Dirac-von Neumann axioms [189]. This description replaced the notion of a possibly nonlinear equation of motion defined over a classical phase space with an infinite dimensional linear operator algebra acting on a Hilbert space. This formulation was not restricted to quantum mechanics. Indeed, an operator-theoretic linearization of classical dynamical systems was contemporaneously made using the Koopman-von Neumann theory [101, 102]. We review the dynamics of quantum states here before transitioning to abstract states of measurement data in Section 4.3.

4.2.1 Autonomous systems

In quantum mechanics, the state of an N-dimensional physical system is represented by a unit vector, or ket, $|\psi\rangle$ in a complex vector space $\mathbb{C}^N$. The dynamics of a quantum state are given by the Schrödinger equation,

$$\frac{\partial}{\partial t}|\psi(t)\rangle = -iH_0|\psi(t)\rangle\,, \qquad |\psi(t_0)\rangle = |\psi_0\rangle\,, \tag{4.1}$$

where the Hamiltonian operator, H_0, has been assumed without loss of generality to be traceless , and Plank's constant $\hbar$ has been set to unity. More generally, an ensemble of pure quantum states can be completely characterized, in the sense of its measurement statistics, by a density matrix $\rho(t)$; that is, a non-negative self-adjoint operator in $\mathbb{C}^{N\times N}$ with trace one. The Liouville-von Neumann, or quantum Liouville equation, describes the time evolution of a density matrix [22]:

$$\frac{\partial}{\partial t}\rho(t) = -i[H_0, \rho(t)] := \mathcal{L}_0\rho(t), \qquad \rho(t_0) = \rho_0\,. \tag{4.2}$$

The quantum Liouville operator $\mathcal{L}_0$ is sometimes known as a *super-operator* because of its linear action on the space of operators. There are a number of ways to *vectorize* the quantum Liouville equation. We will use the language of the familiar Bloch vector (also known as the vector of coherence) to describe passing to a vector of differential equations. Writing $\rho(t) = \mathbb{1}/N + \sum_{j=1}^{N^2-1}\mathrm{Tr}(\rho\sigma_j)\sigma_j$ with $\{\sigma_j\}_{j=1}^{N^2-1}$ as a complete and orthonormal basis for traceless Hermitian operators, we can then define $x_j := \mathrm{Tr}(\rho\sigma_j)$ so that ((4.2)) becomes [5, 107]

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \mathbf{L}_0\mathbf{x}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0\,. \tag{4.3}$$

For a more complete description of this process, see Section 4.2.3. Obtaining an expectation value $x_j(t) := \mathrm{Tr}(\rho(t)\sigma_j)$ requires a collection of identically-prepared quantum states. In this paper, the "measurement" of a quantum state as represented by $\mathbf{x}(t)$ is understood to mean the measurement of such an ensemble.

In terms of the eigenvectors $\mathbf{v}_j$ and eigenvalues λ_j of $\mathbf{L}_0$, the solution to ((4.3)) is

$$\mathbf{x}(t) = \sum_{j=1}^{N^2-1}\mathbf{v}_j e^{\lambda_j(t-t_0)}c_j = \mathbf{V}\exp(\mathbf{\Lambda}(t-t_0))\mathbf{c}\,, \tag{4.4}$$

where the coefficients c_j are the coordinates of $\mathbf{x}_0$ in the eigenvector basis. For continuous dynamics as in ((4.4)), there exists an equivalent discrete time description of the system $\mathbf{x}(t)$

sampled at intervals of Δt.

4.2.2 Non-autonomous systems

Many important classical non-autonomous control systems can be formulated as control-affine dynamical systems. Transforming to the operator theoretic perspective of the Koopman-von Neumann theory, these are bilinear dynamical systems [76]. In this section, we review non-autonomous dynamical systems and establish the bilinear formulation of the quantum control problem.

4.2.2.1 Direct actuation

Direct actuation is a common situation in classical control theory in which an autonomous dynamical system like ((4.3)) undergoes a linear response to control inputs $\mathbf{u}(t) \in \mathbb{R}^{N_c}$ [163]:

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \mathbf{L}_0\mathbf{x}(t) + \mathbf{L}_\mathrm{B}\mathbf{u}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0\,. \tag{4.5}$$

Recall that if the control is input under a zero-order hold across Δt, the discretization $\mathbf{x}_n = \mathbf{x}(t_0 + (n-1)\Delta t)$ and $\mathbf{u}_n = \mathbf{u}(t_0 + (n-1)\Delta t)$ transforms ((4.5)) into the discrete-time dynamical system

$$\mathbf{x}_{n+1} = e^{\mathbf{L}_0\Delta t}\mathbf{x}_n + \left(\int_0^{\Delta t} e^{\mathbf{L}_0(\Delta t - s)}\mathbf{L}_\mathrm{B}\mathrm{d}s\right)\mathbf{u}_n\,. \tag{4.6}$$

Significantly, the dynamics remain control-affine for any span Δt.

4.2.2.2 Bilinear quantum dynamics

The control of a quantum system can be modelled using N_c real-valued control functions, $u_j(t)$, coupled to corresponding time-independent interaction Hamiltonians, H_j, such that the dynamics are described by the bilinear Schrödinger equation,

$$\frac{\partial}{\partial t}\left|\psi(t)\right\rangle = -i\left(H_0 + \sum_{j=1}^{N_c} u_j(t)H_j\right)\left|\psi(t)\right\rangle, \qquad \left|\psi(t_0)\right\rangle = \left|\psi_0\right\rangle\,. \tag{4.7}$$

Following Appendix 4.2.3 like for ((4.3)), the bilinear Schrödinger equation induces the vectorized bilinear quantum Liouville equation [5, 107],

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \left(\mathbf{L}_0 + \sum_{j=1}^{N_c} u_j(t)\mathbf{L}_j\right)\mathbf{x}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0\,, \tag{4.8}$$

by identifying $H_n \mapsto \mathbf{L}_n$ for $n = 0, 1, \ldots, N_c$ and $\mathbf{x} \in \mathbb{R}^{N^2-1}$. This linear differential equation can be integrated in the usual way to obtain

$$\mathbf{x}(t) = e^{\mathbf{L}_0(t-t_0)}\mathbf{x}_0 + \sum_{j=1}^{N_c}\int_{t_0}^{t} e^{\mathbf{L}_0((t-t_0)-s)}u_j(s)\mathbf{L}_j\mathbf{x}(s)\mathrm{d}s. \tag{4.9}$$

If a discretization is taken such that $\mathbf{x}_n = \mathbf{x}(t_0 + (n-1)\Delta t)$ and $u_{j;n} = u_j(t_0 + (n-1)\Delta t)$, then to first order

$$\mathbf{x}_{n+1} = (\mathbb{1} + \mathbf{L}_0\Delta t)\mathbf{x}_n + \sum_{j=1}^{N_c} u_{j;n}\mathbf{L}_j\Delta t\,\mathbf{x}_n + \mathrm{O}(\Delta t^2)\,, \tag{4.10}$$

such that the discrete time dynamical system remains control-affine [159]. Note that in this first-order approximation the computation of derivatives via finite differences and the computation of the discrete time dynamical system are equivalent.

4.2.3 Vectorization of quantum dynamics

In quantum dynamics, a time-independent closed quantum system can be described by a Hermitian operator H_0 (the Hamiltonian) acting on a unit vector $|\psi(t)\rangle$ (the wavefunction) in a complex Hilbert space $\mathcal{H}$ using the Schrödinger equation,

$$\frac{\partial}{\partial t}|\psi(t)\rangle = -\frac{i}{\hbar}H_0|\psi(t)\rangle\,, \quad |\psi(0)\rangle = |\psi_0\rangle\,. \tag{4.11}$$

In this paper we set $\hbar = 1$, choose H_0 to have zero trace, and assume that $\mathcal{H} = \mathbb{C}^N$ has finite dimension. The control of the system can be realized using control functions, $u_k(t)$, coupled

to time-independent interaction Hamiltonians, H_k, leading to a bilinear Schrödinger equation,

$$\frac{\partial}{\partial t}\left|\psi(t)\right\rangle = -i\left(H_0 + \sum_k u_k(t) H_k\right)\left|\psi(t)\right\rangle \ . \tag{4.12}$$

More generally, open quantum dynamics involves mixtures of pure quantum states. The statistics of measurements of this kind of state can be completely described by a density operator, ρ, that is any element in the convex set of all non-negative ($\rho \geq 0$) self-adjoint ($\rho^\dagger = \rho$) linear maps on $\mathcal{H}$ with $\operatorname{Tr}\rho = 1$. The pure state density operator is $\rho(t) = \left|\psi(t)\right\rangle\left\langle\psi(t)\right|$. The case of Markovian dynamics of ρ can be described by the action of completely-positive trace preserving maps generated by the Gorini–Kossakowski–Sudarshan–Lindblad (GKSL) equation [22],

$$\frac{\partial}{\partial t}\rho(t) = \mathcal{L}(t)\rho(t) = -i[H(t), \rho(t)] + \frac{1}{2}\sum_{j,k=1}^{N^2-1} c_{jk}\left([D_j, \rho(t) D_k^\dagger] + [D_j \rho(t), D_k^\dagger]\right) \ , \tag{4.13}$$

where $H(t)$ is a trace-zero Hermitian operator (the system Hamiltonian), $\{D_j\}_{j=1}^{N^2-1}$ is an orthonormal set of complex matrices with trace zero, and $C := (c_{jk})$ is positive semi-definite. For a closed quantum system $C = 0$, and the equation becomes the quantum Liouville equation.

The Lindbladian, $\mathcal{L}$, is sometimes called a super-operator because of its linear action on the operators ρ (in the Liouville limit, $\mathcal{L}$ is instead called the Liouville operator). Several ways to *vectorize* the GKSL equation exist. We will use the physics language of the Bloch vector (also know as the vector of coherence) to describe passing to a vector ODE. Using the trace-one constraint of a density matrix, write

$$\rho(t) = \frac{\mathbb{1}}{N} + \sum_{j=1}^{N^2-1} \operatorname{Tr}(\rho\sigma_j)\sigma_j \tag{4.14}$$

with $\{\sigma_j\}_{j=1}^{N^2-1}$ as a complete and orthonormal basis for traceless Hermitian operators. Take

$x_j := \mathrm{Tr}(\rho\sigma_j)$ so that ((4.13)) becomes

$$\frac{\partial}{\partial t}\mathbf{x}(t) = (L_H + L_D)\mathbf{x}(t) + \mathbf{c} \tag{4.15}$$

by projection onto the basis $\{\sigma_j\}_{j=1}^{N^2-1}$ such that

$$(\mathbf{L}_H)_{j,k} = \sum_{\ell} \mathrm{Tr}(H\,\sigma_\ell) f_{j\ell k} \tag{4.16}$$

$$(\mathbf{L}_D)_{j,k} = -\frac{1}{4}\sum_{\ell,m,n} c_{mn}\left(f_{m\ell j}\left(f_{n\ell k} - i g_{n\ell k}\right) + f_{n\ell j}\left(f_{m\ell k} + i g_{m\ell k}\right)\right) \tag{4.17}$$

$$(\mathbf{c})_j = \frac{i}{N}\sum_{m,n} c_{mn} f_{mnj} \tag{4.18}$$

where $[\sigma_j, \sigma_k] = i\sum f_{jk\ell}\sigma_\ell$ and $\{\sigma_j, \sigma_k\} = \frac{2}{N}\delta_{jk}\mathbb{1} + \sum_\ell g_{jk\ell}\sigma_\ell$ are the structure constants of the basis [5, 107]. Identifying $H(t) = H_0 + \sum_k u_k(t)H_k$ with $\mathbf{L}_H = \mathbf{L}_0 + \sum_k u_k(t)\mathbf{L}_k$ results in a bilinear GKSL equation for the vectorized density matrix,

$$\frac{\partial}{\partial t}\mathbf{x}(t) = (\mathbf{L}_0 + \sum_k u_k(t)\mathbf{L}_k + \mathbf{L}_D)\mathbf{x}(t) + \mathbf{c}\,. \tag{4.19}$$

Similarly, the bilinear quantum Liouville equation is

$$\frac{\partial}{\partial t}\mathbf{x}(t) = (\mathbf{L}_0 + \sum_k u_k(t)\mathbf{L}_k)\mathbf{x}(t) \tag{4.20}$$

4.3 (Bilinear) Dynamic Mode Decomposition

In recent years, a variety of practical computational tools have encouraged more widespread adoption of the Koopman-von Neumann perspective in the dynamical systems community [134, 141]. The Dynamic Mode Decomposition (DMD) broadly refers to a suite of numerical methods originating in the fluid dynamics community for the purpose of studying coherent spatiotemporal structures in complex fluid flows [185]. DMD combines standard dimensionality reduction via the singular value decomposition with Fourier transforms in

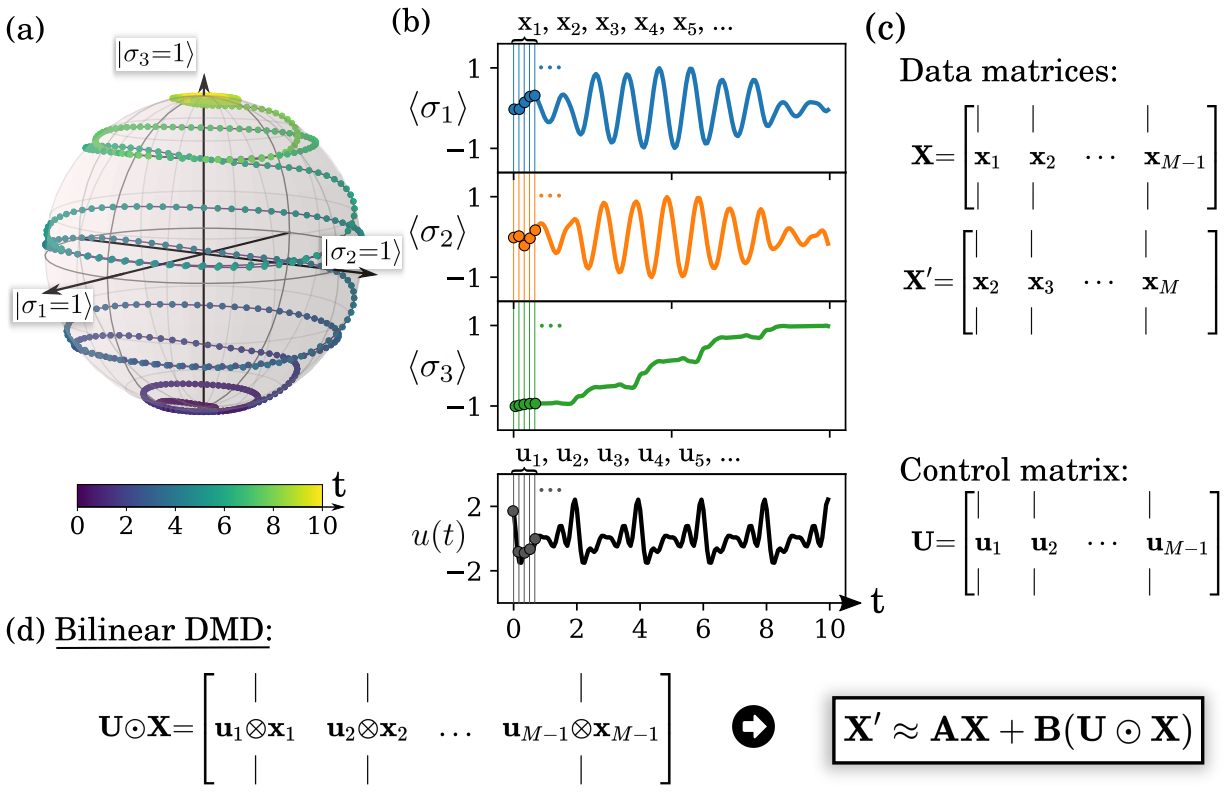


Figure 4.1: The trajectory of a qubit driven by a linearly-polarized semi-classical drive $\mathrm{u}(t)$ (Hamiltonian: $H(t) = \pi\sigma_z + \mathrm{u}(t)\sigma_x$) is shown on the Bloch sphere in Figure 4.1(a). The corresponding Pauli-spin measurements are shown in Figure 4.1(b). Measurements $\mathbf{x}_j$, $j = 1, 2, \ldots$, are taken at discrete time steps and assembled into offset snapshot matrices $\mathbf{X}$ and $\mathbf{X}'$ in Figure 4.1(c). The bilinear Dynamic Mode Decomposition (biDMD, Figure 4.1(d)) is a regression-based algorithm that uses the assembled data matrices and control input from sufficiently-resolved data to learn the intrinsic dynamics, $\mathbf{A}$ and the control, $\mathbf{B}$, for the bilinear dynamics.

time. In application, all variations of the DMD algorithm involve collecting a time series of experimental measurements in order to compute a reduced-order dynamical model based on DMD modes and eigenvalues. The DMD modes identify coherent structures in the measured space and the DMD eigenvalues define the growth, decay, and oscillation frequency of each mode. The following sections will review the DMD framework.

4.3.1 DMD

The standard DMD is a data-driven algorithm defined via the observed trajectories from either experimental data or a numerical simulation of a dynamical system

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \mathbf{L}_0\mathbf{x}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0\,, \tag{4.21}$$

where $\mathbf{x}(t) \in \mathbb{R}^{N^2-1}$ to remain consistent with Section 4.2. (DMD ultimately deals with data and is agnostic to the underling system that produced it.) The first step of the algorithm is to assemble a sequential measurement record $\{\mathbf{x}(t_1) = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$, $t_1 < t_2 < \cdots < t_m$, into *snapshot matrices*:

$$\mathbf{X} := \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_{M-1} \\ | & | & & | \end{bmatrix}, \qquad \mathbf{X}' := \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \ldots & \mathbf{x}_{M} \\ | & | & & | \end{bmatrix}, \tag{4.22}$$

so $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{N^2-1\times M-1}$. In this paper we assume uniform sampling with $t_m := t_0 + (m-1)\Delta t$, but this assumption can be relaxed [8]. In its simplest form, DMD is a regression algorithm that estimates the propagator matrix

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} \tag{4.23}$$

by solving the optimization problem

$$\mathbf{A} = \arg\min_{\hat{\mathbf{A}}} \left\| \hat{\mathbf{A}}\mathbf{X} - \mathbf{X}' \right\|_F \tag{4.24}$$

where $\|\mathbf{M}\|_F = \sqrt{\sum_{j=1}^{J}\sum_{k=1}^{K} m_{jk}}$ is the Frobenius matrix norm defined for any given matrix $\mathbf{M}$. The least-squares solution to ((4.24)) is $\mathbf{A} = \mathbf{X}'\mathbf{X}^{+}$ where $+$ denotes the Moore-Penrose pseudoinverse of a matrix. From the eigendecomposition $\mathbf{A} = \mathbf{W}\mathbf{\Omega}\mathbf{W}^{-1}$, define the DMD modes $\mathbf{w}_j$ as the columns of $\mathbf{W}$. Future states can then be predicted as in a discrete analogue to ((4.4)) using

$$\mathbf{x}_n = \sum_{j=1}^{N^2-1} \mathbf{w}_j \omega_j^n b_j = \mathbf{W}\mathbf{\Omega}^n\mathbf{b} \tag{4.25}$$

where $\mathbf{b}$ are the coefficients of the initial condition in the eigenvector basis, $\mathbf{x}_0 = \mathbf{W}\mathbf{b}$.

4.3.1.1 Scalability

The eigendecomposition of $\mathbf{A} \in \mathbb{R}^{N^2-1\times N^2-1}$ can easily become numerically intractable for large N. However, the big data limit is nothing unusual for the problems appearing in data science and fluid dynamics that DMD has already addressed with success. The DMD algorithm circumvents this problem by projecting the high-dimensional snapshot matrix $\mathbf{X}$ onto a low-rank subspace defined by R modes computed from the singular value decomposition (SVD) of $\mathbf{X}$ (note that because it is computed from the snapshot matrices, the rank of $\mathbf{A}$ is at most $M-1$ when $N^2 > M$). In this way, DMD determines a low-rank approximation $\tilde{\mathbf{A}} \in \mathbb{R}^{R\times R}$ to describe the dynamics of the measured trajectories. In the big data limit, the low-rank approximations in DMD can be computed using randomized algorithms so that the required computations scale with the intrinsic rank of the data instead of the ambient measurement dimension [60].

This paper introduces DMD for quantum control by way of simple low-rank examples. Therefore, there are outstanding and important scalability questions that must be addressed. For instance, low-rank approximations and reduced-order models are not appropriate for

capturing the full expression of a generic quantum dynamical system. However, suitable constraints on the control can ensure the SVD-based DMD is a natural choice for the study of many-body quantum dynamics [105]. As well, DMD accommodates the use of tensor decompositions [71, 100] and is thereby equipped for those quantum states represented by tensor networks [17]. Improvements may also be made by incorporating approximately-known theoretical models; after a straightforward modification of Equation (4.8) our framework becomes one for bilinear discrepancy modeling. This may allow us to introduce low-rank assumptions in the discrepancy space that could not be made in the state space [81]. The DMD framework includes a selection of well-established extensions and improvements beyond the naive algorithms introduced in this paper–many focus on adapting to situations with meager amounts of noisy time series data. We discuss some possible directions of future research in our concluding remarks in Section 4.5.

4.3.2 DMD with Control (DMDc)

Dynamic Mode Decomposition with control (DMDc) [163] is an algorithm developed for modeling the special case of direct actuation:

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \mathbf{L}_0\mathbf{x}(t) + \mathbf{L}_\mathrm{B}\mathbf{u}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0\,, \tag{4.26}$$

with $\mathbf{u}(t) \in \mathbb{R}^{N_c}$. Like DMD in ((4.22)), the size-M measurement record associated with the trajectory of $\mathbf{x}$ is assembled into snapshot matrices $\mathbf{X}$ and $\mathbf{X}'$. This time, the control inputs, $\mathbf{u}_n := \mathbf{u}(t_n)$, are also collected:

$$\boldsymbol{U} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{M-1} \\ | & | & & | \end{bmatrix}. \tag{4.27}$$

Algorithm 2 Bilinear Dynamic Mode Decomposition

INPUT: Snapshot data $\mathbf{X}$, shifted snapshot data $\mathbf{X}'$, control matrix $\mathbf{U}$, and target ranks $\tilde{r}, \hat{r}$

OUTPUT: DMD modes $\mathbf{\Phi}$, eigenvalues $\mathbf{\Lambda}$, and estimates $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{B}}$ for the drift and control matrices

function BIDMD($\mathbf{X}$, $\mathbf{X}'$, $\mathbf{U}$, $\tilde{r}$, $\hat{r}$)

$\quad \mathbf{\Xi} \leftarrow \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \odot \mathbf{X} \end{bmatrix}$

$\quad \widetilde{\mathbf{U}}, \widetilde{\mathbf{\Sigma}}, \widetilde{\mathbf{V}} \leftarrow \mathrm{SVD}(\mathbf{\Xi}, \tilde{r})$ ▷ Truncated $\tilde{r}$-rank SVD of $\mathbf{\Xi}$

$\quad [\widetilde{\mathbf{U}}_A, \widetilde{\mathbf{U}}_B] \leftarrow \widetilde{\mathbf{U}}$ ▷ Decompose $\widetilde{\mathbf{U}}$ according to $\mathbf{A}$ and $\mathbf{B}$

$\quad \widetilde{\mathbf{A}} \leftarrow \mathbf{X}'\widetilde{\mathbf{V}}\widetilde{\mathbf{\Sigma}}^{-1}\widetilde{\mathbf{U}}_A^{\dagger}$ ▷ Estimate for $\mathbf{A}$

$\quad \widetilde{\mathbf{B}} \leftarrow \mathbf{X}'\widetilde{\mathbf{V}}\widetilde{\mathbf{\Sigma}}^{-1}\widetilde{\mathbf{U}}_B^{\dagger}$ ▷ Estimate for $\mathbf{B}$

$\quad \widehat{\mathbf{U}}, \widehat{\mathbf{\Sigma}}, \widehat{\mathbf{V}} \leftarrow \mathrm{SVD}(\mathbf{X}', \hat{r})$ ▷ Truncated $\hat{r}$-rank SVD of $\mathbf{X}'$

$\quad \widehat{\mathbf{A}} \leftarrow \widehat{\mathbf{U}}^{\dagger}\widetilde{\mathbf{A}}\widehat{\mathbf{U}}$ ▷ Low-rank approximation for $\mathbf{A}$

$\quad \mathbf{W}, \mathbf{\Lambda} \leftarrow \mathrm{EIG}(\widehat{\mathbf{A}})$ ▷ Eigendecomposition of $\widehat{\mathbf{A}}$

$\quad \mathbf{\Phi} \leftarrow \widetilde{\mathbf{A}}\widehat{\mathbf{U}}\mathbf{W}$ ▷ DMD modes for $\mathbf{A}$

end function

DMDc is a regression-based approach to system identification that disambiguates the intrinsic dynamics, $\mathbf{A}$, and the effects of control, $\mathbf{B}$:

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix} := \mathbf{G}\mathbf{\Xi}\,. \tag{4.28}$$

The DMDc algorithm achieves this disambiguation by interpreting the best-fit solution according to

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} = \mathbf{X}'\mathbf{\Xi}^{+}\,. \tag{4.29}$$

DMDc Bristowbrings all the benefits of the DMD algorithm to a model-free framework for experimental control optimization.

4.3.3 Bilinear DMD (biDMD)

Bilinear Dynamic Mode Decomposition (biDMD, Algorithm 2) is a data-driven system identification framework for bilinear non-autonomous control systems:

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \left(\mathbf{L}_0 + \sum_{j=1}^{N_c} u_j(t)\mathbf{L}_j\right)\mathbf{x}(t)\,, \quad \mathbf{x}(t_0) = \mathbf{x}_0\,. \tag{4.30}$$

Construct the snapshot matrices $\mathbf{X}$, $\mathbf{X}'$, and $\mathbf{U}$ as in ((4.22)) and ((4.27)) using the size-M measurement record of the trajectory $\mathbf{x}(t)$ and the corresponding input record $\mathbf{u}(t)$. Additionally, construct the bilinear snapshot matrix

$$\mathbf{U} \odot \mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 \otimes \mathbf{x}_1 & \mathbf{u}_2 \otimes \mathbf{x}_2 & \dots & \mathbf{u}_{M-1} \otimes \mathbf{x}_{M-1} \\ | & | & & | \end{bmatrix}. \tag{4.31}$$

where $\mathbf{u} \otimes \mathbf{x} := [u_1x_1, u_1x_2, \dots, u_1x_{N^2-1}, u_2x_1, u_2x_2, \dots, u_{N_c}x_{N^2-1}]^{\mathrm{T}}$ is the Kronecker product and $\mathbf{U} \odot \mathbf{X}$ is the column-wise Kronecker product of two matrices (also known as the Khatri–Rao product). The discrete time dynamics of a bilinear dynamical system are well-approximated (for small enough Δt) [159] by

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\left(\mathbf{U} \odot \mathbf{X}\right). \tag{4.32}$$

where $\mathbf{X} \in \mathbb{R}^{N^2-1 \times M-1}$, $\mathbf{U} \in \mathbb{R}^{N_c \times M-1}$, $\mathbf{U} \odot \mathbf{X} \in \mathbb{R}^{N_c(N^2-1) \times M-1}$, $\mathbf{A} \in \mathbb{R}^{N^2-1 \times N^2-1}$, and $\mathbf{B} \in \mathbb{R}^{N^2-1 \times N_c(N^2-1)}$. In its simplest form, the biDMD algorithm is a regression algorithm in the spirit of DMD ((4.23)) and DMDc ((4.28)) such that

$$\mathbf{X}' \approx \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \odot \mathbf{X} \end{bmatrix} := \mathbf{G}\boldsymbol{\Xi} \tag{4.33}$$

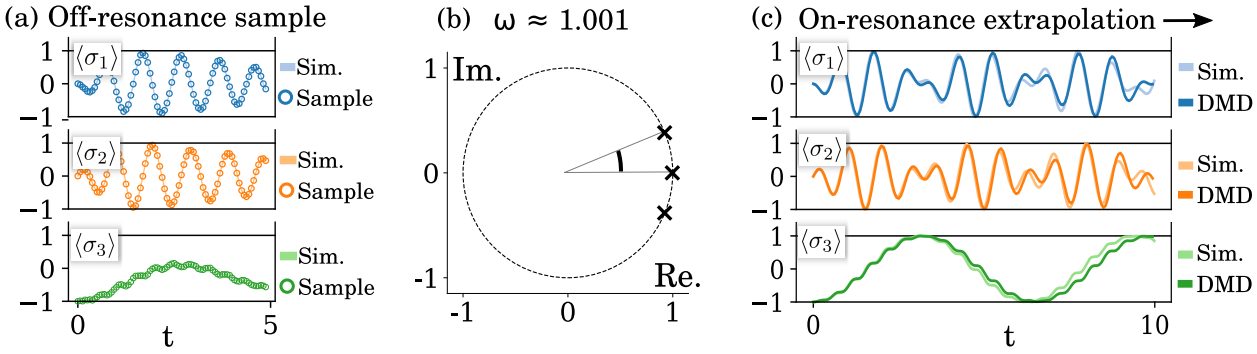


Figure 4.2: In Figure 4.2(a) five periods ($T = 1$) of highly-resolved Pauli-spin data ($\Delta t = T/16$) with additive Gaussian noise ($\sigma = 0.01$) are sampled from simulations of the system $H(t) = \pi\sigma_z + \cos(2\pi\omega_D)\sigma_x$ driven by the slightly off-resonance frequency $\omega_D = 1.1$. The biDMD algorithm disambiguates the drift, $\mathbf{A}$, and control, $\mathbf{B}$, dynamical operators. The biDMD eigenvalues of $\mathbf{A}$ are plotted in Figure 4.2(b). In Figure 4.2(c) a new control is provided using the resonance frequency, $\omega \approx 1.001$, estimated from the DMD eigenvalues in Figure 4.2(b). The biDMD model extrapolates using only an initial state and a desired control. Figure 4.2(c) compares the biDMD extrapolation to the known simulation.

Like DMDc, the biDMD algorithm (Algorithm 2) disambiguates the effect of the intrinsic dynamics, $\mathbf{A}$, from the effect of the control inputs, $\mathbf{B}$, using a factorization of the best-fit solution:

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} = \mathbf{X}'\boldsymbol{\Xi}^{+}. \tag{4.34}$$

4.3.3.1 Example 1

Consider a two-level quantum system, $H(t) = \pi\sigma_3 + u(t)\sigma_1$, where $\{\sigma_j\}_{j=1}^{3}$ are the standard Pauli matrices. This can be realized, to take an example from quantum optics, by applying a linearly polarized semi-classical drive to a dipole-allowed atomic transition $\Delta E = 2\pi$ [22]. Suppose the dynamics have been inaccurately characterized so that the system is driven in an experiment by a slightly off-resonant pulse, $u(t) = \cos(2\pi\omega_D t)$ with $\omega_D = 1.1$. In Figure 4.2, the biDMD algorithm uses samples from such an experiment in order to disambiguate the effect of control. The phase of the DMD eigenvalues ((4.25)) provide an estimate of the resonance frequency, $\omega \approx 1.001$. The biDMD model can then be used to predict the behavior

of the system under the influence of this estimated-resonance drive without a priori obtaining an accurate characterization.

Recall the trajectory samples are measured via the expected value of an ensemble of identically prepared quantum states. Advantageously, the DMD algorithm is based on a least-squares regression that is optimal with respect to Gaussian measurement noise of the finite statistics. The algorithm does not require much data. Moreover, if the algorithm is applied to a system with dissipation, then the model extracts the corresponding modified drift and control operators. An otherwise identical result to Figure 4.2 can be obtained without any change in procedure.

4.4 DMD for Stroboscopic Sampling

Recall that the DMD algorithms in Section 4.3 are applicable only to systems in which the measurement resolution Δt is small enough to accommodate the linear approximation to the desired accuracy. Higher-order approximations to ((4.10)) show that improvements to the biDMD model can be made by adding terms nonlinear in the control [159]. This research direction may be appropriate for cases where the zero-order hold on the control is extended for the entirety of the interval Δt. Instead, in this section we study the case of stroboscopic measurements of the trajectory $\mathbf{x}(t)$ separated by time steps $T \gg \Delta t$ during which the control can change significantly.

The observation of low-frequency dynamics $\sim T$ in a system under the influence of relatively high-frequency actuation $\sim \Delta t$ is a familiar feature of bilinear systems. For example, the classroom experiment of an inverted pendulum stabilized by the high-frequency vertical drive of its pivot can be modelled as a bilinear control system (under an appropriate operator-theoretic transformation of Mathieu's equation). The two-level quantum system from Section 4.3.3.1 provides another example. The textbook analysis of the two-level system proceeds via a rotating frame transformation together with the rotating wave approximation (RWA). Consider $H(t) = \pi\sigma_3 + u(t)\sigma_1$ driven by a pure tone $u(t) = u_0 \cos(2\pi\nu t)$. Changing

to a frame rotating with the drive frequency about the σ_3 axis of the Bloch representation,

$$\tilde{H} = U^\dagger H U - iU^\dagger \frac{\partial U}{\partial t} = \underbrace{\pi\nu\mathbb{1} + \pi(1-\nu)\sigma_3 + \frac{u_0}{2}\sigma_1}_{\text{RWA}} + \frac{u_0}{2}(\cos(4\pi\nu t)\sigma_1 - \cos(4\pi\nu t)\sigma_2) \quad (4.35)$$

with $U(t) = \exp(2\pi\nu i(\mathbb{1} - \sigma_3)t/2)$ where the RWA disregards the contribution of the emergent high-frequency terms to realize a constant Hamiltonian. The coefficient of σ_x in the RWA is u_0 which can be small relative to $2\pi\nu$. In this case, the characteristic Rabi cycle of the two-level system is a low-frequency oscillation divorced from the high-frequency drive. Motivated by these examples, Sections 4.4.1-4.4.2 discuss DMD strategies for the case of stroboscopic measurements.

4.4.1 DMD and Floquet Theory

In this section, we recall how periodically-driven dynamical systems are connected to the case of stroboscopic measurements (and rotating frames) using the framework of Floquet theory. We introduce Floquet theory and show how corresponding re-interpretations of DMD enable an efficient method for studying stroboscopically-measured, periodically-driven quantum systems.

In Floquet theory (known as Bloch theory in condensed matter physics), the discrete-time evolution of a periodic quantum dynamical system, $H(t+T) = H(t)$, is given by the time-independent stroboscopic Floquet Hamiltonian [29],

$$U(t_0 + T, t_0) = \exp(-iH_F[t_0]T)\,. \quad (4.36)$$

The dependence on the initial time, t_0, is a gauge transformation known as the *Floquet gauge*. The Floquet Hamiltonian is a consequence of the observation $U(t_0+nT, t_0) = U(t_0+T, t_0)^n$. A classical version of ((4.36)) also exists; for vectorized quantum systems, exchange $-iH_F[t_0] \mapsto \mathbf{L}_F[t_0]$ in the manner outlined in Appendix 4.2.3.

Take the eigendecomposition of the Floquet operator such that $\mathbf{L}_F[t_0]\boldsymbol{\xi}_j(t_0) = \varepsilon_j\boldsymbol{\xi}_j(t_0)$

for $j = 1, 2, \ldots, N^2 - 1$. Floquet's theorem is the observation that the T-periodic fast-time dynamics can be separated from the slow-time dynamics governed by $\mathbf{L}_F[t_0]$. The theorem asserts that attaching the periodic fast-time propagation to the eigenvectors, $\boldsymbol{\xi}_j(t+T) = \boldsymbol{\xi}_j(t)$, provides a complete set of solutions for the dynamics. In this Floquet basis, the general time-evolution of an initial state $\mathbf{x}(t_0)$ is given by

$$\mathbf{x}(t) = \sum_{j=1}^{N^2-1} \boldsymbol{\xi}_j(t) e^{\varepsilon_j (t-t_0)} c_j \,, \tag{4.37}$$

where the $\boldsymbol{\xi}(t)$ are known as *Floquet modes*, the ε_j are known as *quasi-energies*, and the coefficients c_j are the coordinates of $\mathbf{x}(t_0)$.

Comparing ((4.4)) with ((4.37)), it is evident the DMD algorithm ((4.23)) can provide a framework for the numerical approximation of Floquet modes and quasi-energies using DMD modes and eigenvalues [143]. We refer to this re-interpretation of the DMD algorithm as Floquet DMD. Suppose a size-M measurement record of the trajectory $\mathbf{x}(t)$ is obtained from a non-autonomous system with a known period T. Further suppose that the first s measurements are contained within $[0, T)$, the second within $[T, 2T)$, and so on under the constraint $t_r \equiv t_{ns+r} \mod T$ for $r = 1, 2, \ldots, s$. Consider the reshaped snapshot matrices for this stroboscopically-measured data,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_{s+1} & \cdots & \mathbf{x}_{(\lfloor (M-1)/s \rfloor - 1)s+1} \\ \mathbf{x}_2 & \mathbf{x}_{s+2} & \cdots & \mathbf{x}_{(\lfloor (M-1)/s \rfloor - 1)s+2} \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_s & \mathbf{x}_{2s} & \cdots & \mathbf{x}_{(\lfloor (M-1)/s \rfloor - 1)s+s} \end{bmatrix},$$

$$\mathbf{X}' = \begin{bmatrix} \mathbf{x}_{s+1} & \mathbf{x}_{2s+1} & \cdots & \mathbf{x}_{(\lfloor (M-1)/s \rfloor)s+1} \\ \mathbf{x}_{s+2} & \mathbf{x}_{2s+2} & \cdots & \mathbf{x}_{(\lfloor (M-1)/s \rfloor)s+2} \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_{2s} & \mathbf{x}_{3s} & \cdots & \mathbf{x}_{(\lfloor (M-1)/s \rfloor)s+s} \end{bmatrix} \tag{4.38}$$

where $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{s(N^2-1)\times\lfloor(M-1)/s\rfloor}$. In this way, the columns of $\mathbf{X}$ exist in the span of a discretization of the T-periodic Floquet modes, $\{\boldsymbol{\xi}_j(t)\}_{j=1}^{N^2-1}$. These Floquet modes evolve with a single phase given by the quasi-energy. From the snapshot matrices, the DMD algorithm constructs an optimal propagator in terms of DMD modes and eigenvalues that provide numerical approximations to the Floquet modes and quasi-energies, respectively.

4.4.1.1 Example 2

Consider the two-level quantum system from Section 4.3.3.1 with $H(t) = \pi\sigma_3 + u(t)\sigma_1$ driven by a control $u(t) = \cos(2\pi\omega_D t)$ with $\omega_D = 1.1$. Suppose the system is stroboscopically measured at 4 times per period over a total $4T$ (such that $M = 16$ and $s = 4$). Construct the snapshot matrices ((4.38)) from this measurement record. In Figure 4.3(a,b), the DMD modes and eigenvalues from applying the Floquet DMD algorithm are compared with the exact Floquet modes, exact quasi-energies, the rotating wave approximation (RWA) modes, and the RWA eigenvalues (see Equation (4.35)). Because $\omega_D = 1.1$ is close to resonance, the RWA is valid and able to capture the quasi-energies. However, RWA does so by sacrificing the fast-time dynamics. In contrast, the Floquet DMD is not constrained to the regime of the RWA and resolves both the fast and slow time-scales using the stroboscopic measurement data. The Floquet DMD provides a linear reduced order model that approximates the exact linear dynamics of any periodically-driven quantum dynamical system. This implies the method is well-suited to extrapolation, as shown in Figure 4.3(c).

The Floquet DMD inherits all the model-reduction advantages of the DMD algorithm. However, one drawback of Floquet DMD is that control is internal to the constructed model. As such, the solution can assist control strategies involving switched systems, but cannot generalize to unseen control inputs.

4.4.2 biDMD and Average Hamiltonian Theory

Average Hamiltonian Theory (AHT) [24, 186] is a method with origins in the NMR community that asserts that the dynamics of a quantum dynamical system driven by a time-dependent control can be described by the average effect of the field over a period T of its oscillation.

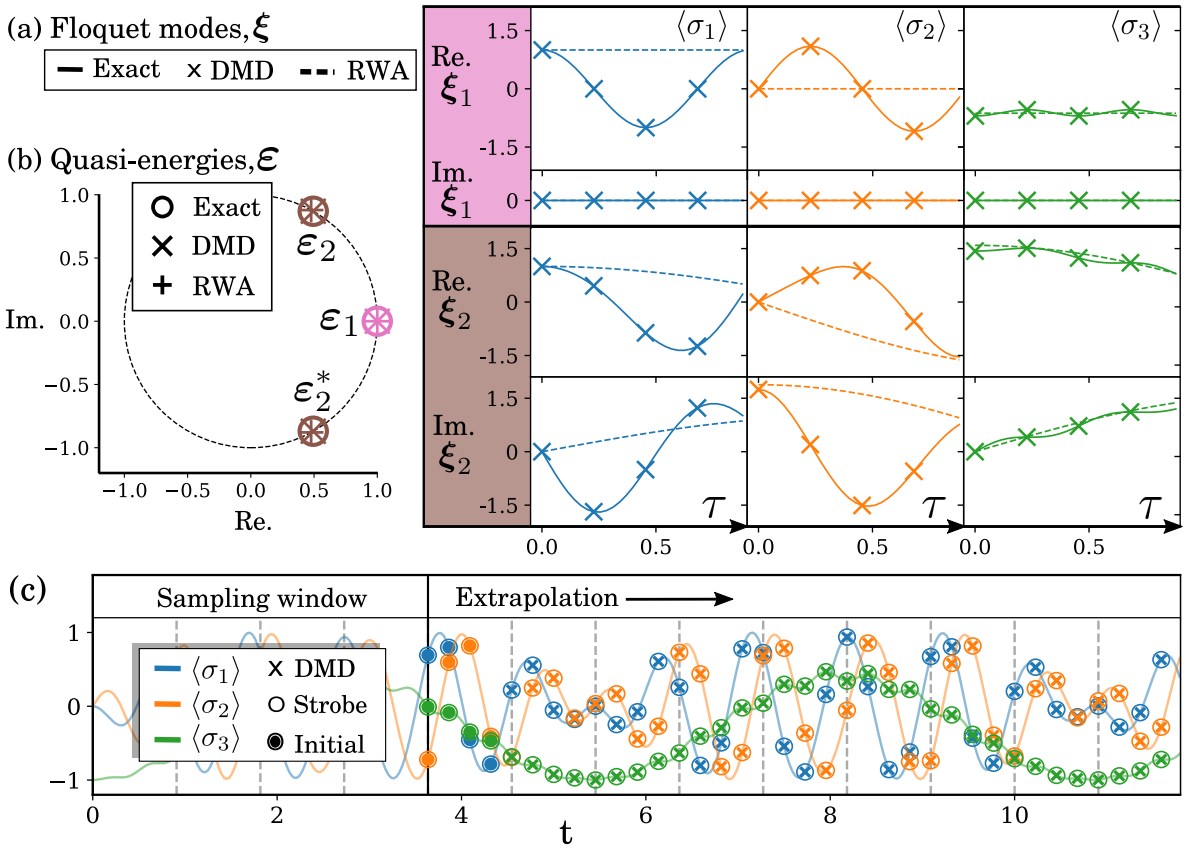


Figure 4.3: 1 A qubit with intrinsic period 1 is driven by a slightly off-resonance frequency with period $T = 1/1.1$. Equi-spaced stroboscopic measurements are sampled at a frequency of $4/T$ during a total sampling window of $4T$ to construct a model using Floquet DMD. Figures 4.3(a-b) compare Floquet modes and quasi-energies to the eigenmodes and eigenvalues from DMD and the rotating wave approximation (RWA). The columns in Figure 4.3(a) are indexed by the coordinate of the state $\mathbf{x}(t) = \mathrm{Tr}(\rho\sigma_j)$. The horizontal axes are labelled by $\tau \equiv t \mod T$. Measurement data from a fifth period is used to provide initial conditions for the model-based extrapolation in Figure 4.3(c).

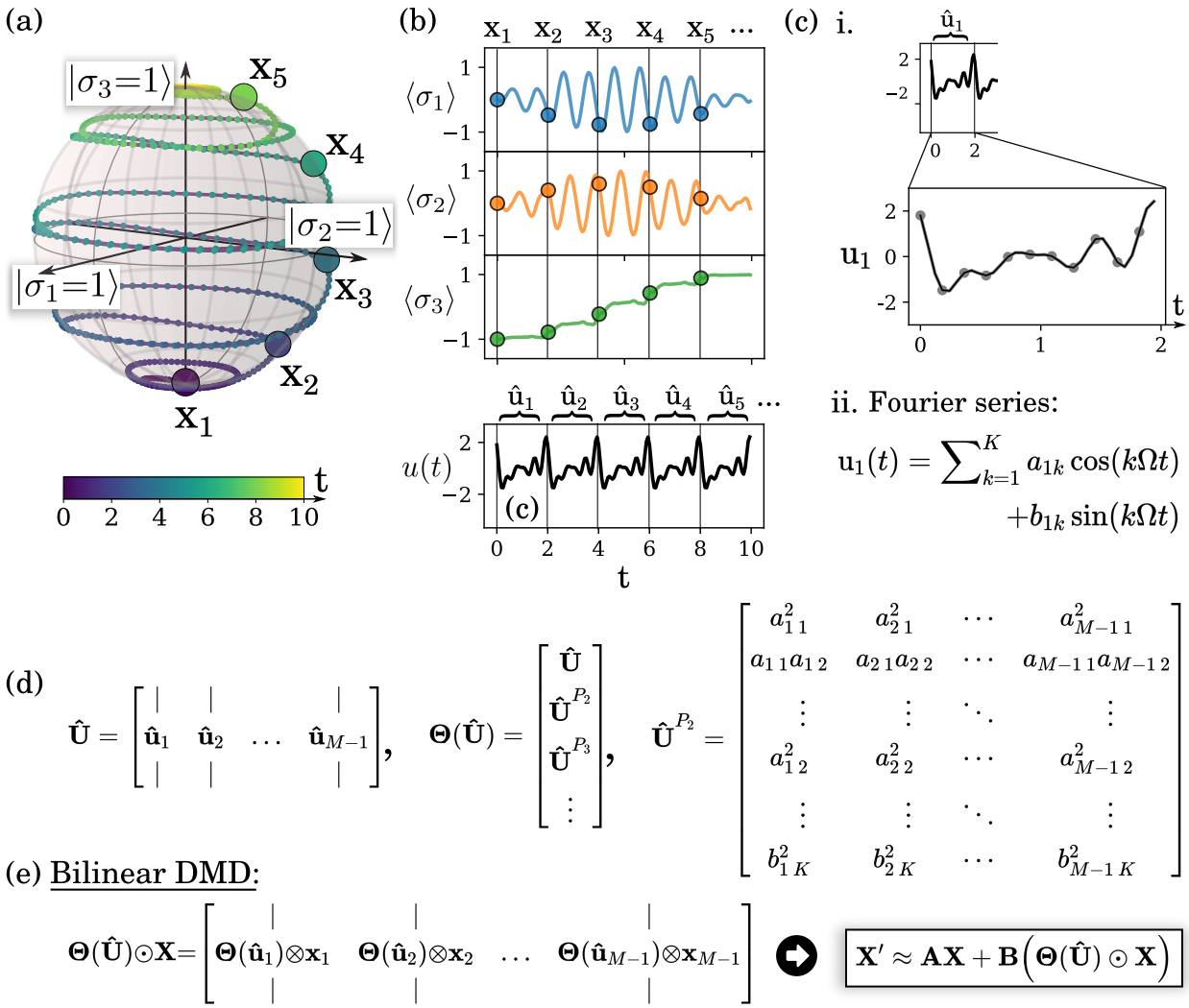


Figure 4.4: Stroboscopic measurements of a qubit driven by a linearly-polarized semi-classical drive $\mathrm{u}(t)$ (Hamiltonian: $H(t) = \pi\sigma_z + \mathrm{u}(t)\sigma_x$) are shown on the Bloch sphere in Figure 4.4(a). The corresponding Pauli-spin measurements are shown in Figure 4.4(b). Control is incorporated by computing Fourier coefficients of each intra-stroboscopic control input (Figure 4.4(c)): $\hat{\mathbf{u}}_n = [a_{n1}, a_{n2}, \dots, a_{nK}, b_{n1}, b_{n2}, \dots, b_{nK}]^{\mathrm{T}}$. Potentially nonlinear combinations of the control coefficients are assembled into a control data matrix in Figure 4.4(d). The stroboscopic bilinear Dynamic Mode Decomposition (biDMD, Figure 4.4(e)) is a regression-based algorithm that uses the assembled data matrices and control input from stroboscopic measurements to learn the intrinsic dynamics, $\mathbf{A}$ and the control, $\mathbf{B}$, for the bilinear dynamics.

In AHT, a known system Hamiltonian is expanded analytically according to the Magnus series [19, 132]. In this section, we show how AHT can be combined with the ideas of Floquet DMD from Section 4.4.1 to provide a framework for biDMD ((4.33)) in the case of stroboscopic measurement data. This allows us to extend model-free optimal control to the limit of stroboscopic measurements. We refer to this interpretation of biDMD as AHT-biDMD.

Consider a periodic bilinear dynamical system

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \mathbf{L}(t)\mathbf{x}(t) = (\mathbf{L}_0 + u(t)\mathbf{L}_u)\,\mathbf{x}(t) \tag{4.39}$$

such that $\mathbf{L}(t+T) = \mathbf{L}(t)$. Define $\Omega := 2\pi/T$. Recall the existence of the constant Floquet operator $\mathbf{L}_F[t_0]\mathbf{x}(t_0) = \mathbf{x}(t_0+T)$. The main idea of this section is to approximate $\mathbf{L}_F[t_0]$ using a high-frequency perturbative expansion which will motivate a way to include control as a parameter in the spirit of biDMD. There are a number of perturbative methods to find the Floquet operator in the limit of a high-frequency drive; we will use the familiar Magnus expansion as used in AHT which allows us to represent the constant Floquet operator $\mathbf{L}_F[t_0]$ as a series of constant operators [29],

$$\mathbf{L}_F[t_0] = \sum_{j=0}^{\infty} \mathbf{L}_F^{(j)}[t_0] \tag{4.40}$$

where $L_F^{(n)}[t_0]$ means that the operator appears in the series expansion with a pre-factor Ω^{-n}. By inserting the Fourier series $u(t) = \sum_{k=1}^{K} a_k \cos(k\Omega t) + b_k \sin(k\Omega t)$ into each $\mathbf{L}_F^{(j)}[t_0]$ in ((4.40)), $j = 0, 1, \ldots, J-1$, we obtain a bilinear model up to terms of order T^{-J} in which controls enter the Floquet operator as multinomials of the constant Fourier series coefficients. An analytic example is shown in Example 11. In the Magnus expansion we presume these coefficients remain constant across a single Floquet period $[(n-1)T, nT]$ but allow for changes between periods. For example, if we permit only the resonance frequency in the Fourier expansion of the control, then our method reduces to an on-resonance carrier with an envelope

under a zero-order hold for the length of the Floquet period. The bilinearization of the propagator through the basis decomposition of the control is motivated by analytic Floquet-Fourier methods [35, 44, 175]. To apply AHT-biDMD, construct the snapshot matrices $\mathbf{X}$ and $\mathbf{X}'$ as in ((4.22)) using the size-M measurement record of the trajectory $\mathbf{x}(t)$ where $\mathbf{x}_n$ are measured stroboscopically, $t_n = t_0 + (n-1)T$. Additional intra-stroboscopic measurements can be included using the methods outlined in Section 4.4.1. Next, suppose controls are restricted to $u_n(t) = \sum_{k=1}^{K} a_{nk}\cos(k\Omega t) + b_{nk}\sin(k\Omega t)$ where n indexes the relevant step $[(n-1)T, nT]$ of the applied control. Define a control snapshot in terms of the coefficients, $\hat{\mathbf{u}}_n = [a_{n1}, a_{n2}, \ldots, a_{nK}, b_{n1}, b_{n2}, \ldots, b_{nK}]^{\mathrm{T}}$, such that the snapshot matrix is now

$$\hat{\mathbf{U}} = \begin{bmatrix} | & | & & | \\ \hat{\mathbf{u}}_1 & \hat{\mathbf{u}}_2 & \cdots & \hat{\mathbf{u}}_{M-1} \\ | & | & & | \end{bmatrix}. \tag{4.41}$$

Furthermore, define the polynomial library matrix to be

$$\boldsymbol{\Theta}(\hat{\mathbf{U}}) = \begin{bmatrix} \hat{\mathbf{U}} \\ \hat{\mathbf{U}}^{P_2} \\ \hat{\mathbf{U}}^{P_3} \\ \vdots \end{bmatrix}, \tag{4.42}$$

where e.g. $\hat{\mathbf{U}}^{P_2}$ denotes all quadratic nonlinearities of the control coefficients,

$$\hat{\mathbf{U}}^{P_2} = \begin{bmatrix} a_{1\,1}^2 & a_{2\,1}^2 & \cdots & a_{M-1\,1}^2 \\ a_{1\,1}a_{1\,2} & a_{2\,1}a_{2\,2} & \cdots & a_{M-1\,1}a_{M-1\,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1\,2}^2 & a_{2\,2}^2 & \cdots & a_{M-1\,2}^2 \\ \vdots & \vdots & \ddots & \vdots \\ b_{1\,K}^2 & b_{2\,K}^2 & \cdots & b_{M-1\,K}^2 \end{bmatrix}. \tag{4.43}$$

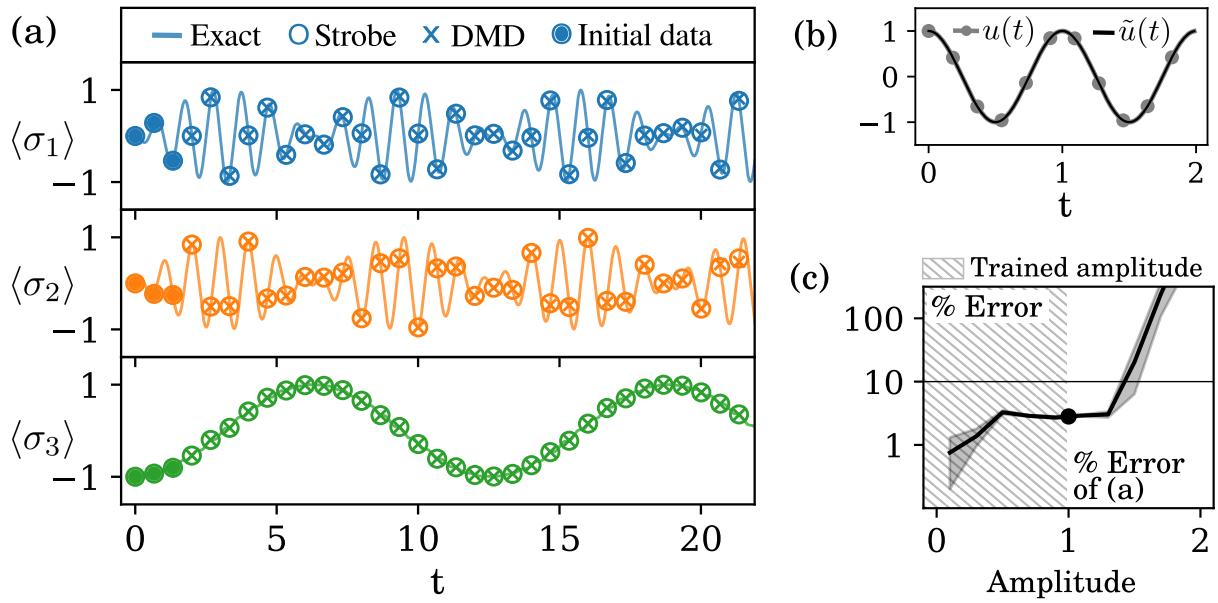


Figure 4.5: Figure 4.5(a) shows the true trajectory compared to the biDMD predictions for a two-level system driven at resonance. Figure 4.5(b) plots the control over a single period and shows an exact match with the truncated Fourier series whose coefficients make up the control input of the biDMD model. Figure 4.5(c) shows the error of (a) and for additional amplitudes of the same control.

Note that in many control systems knowledge of the experimental driving pulse is incomplete as distortions can introduce nonlinear transfer functions of the expected input as $f(u(t))$. Because this is a data-driven approach, the AHT-biDMD model implicitly accounts for the appropriate nonlinear transfer function for the experimental control up to the truncated order of these multinomials. Finally, use the library to construct the bilinear operator as in ((4.31)), $\boldsymbol{\Theta}(\hat{\mathbf{U}}) \odot \mathbf{X}$. Now, AHT-biDMD is the approximation of the discrete time dynamics by the model

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\left(\boldsymbol{\Theta}(\hat{\mathbf{U}}) \odot \mathbf{X}\right). \tag{4.44}$$

The algorithm proceeds as in biDMD in Section 4.3.3.

Example 11: Average Hamiltonian Theory example

Recall the two-level quantum dynamical system (4.3.3.1) with $H(t) = \pi\sigma_3 + u(t)\sigma_1$ where $\{\sigma_j\}_{j=1}^3$ are the standard Pauli matrices. The corresponding vectorized Liouville equation is attained via the Bloch representation discussed in Appendix 4.2.3 where $x_j = \mathrm{Tr}(\rho(t)\sigma_j)$ such that

$$\frac{\partial}{\partial t}\mathbf{x}(t) = \mathbf{L}(t)\mathbf{x}(t) = 2\pi\mathbf{L}_3\mathbf{x}(t) + 2u(t)\mathbf{L}_1\mathbf{x}(t)\,. \tag{4.45}$$

where

$$\mathbf{L}_1 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{L}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{L}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{4.46}$$

are the vectorized Pauli matrices. Consider a fixed-frequency drive $u(t) = u\cos(k\Omega t) + v\sin(k\Omega t)$. The goal is to compute the Floquet operator $\mathbf{L}_F[0]\mathbf{x}(nT) = \mathbf{x}((n+1)T)$, with $T = 2\pi/\Omega$, to second order in Ω using the Magnus expansion. Take $\mathbf{L}_F^{(n)}[t_0]$ to mean that the operator appears in the expansion with a pre-factor Ω^{-n}. The first few terms in the Magnus expansion are:

$$\begin{aligned}
\mathbf{L}_F^{(0)}[t_0] &= \frac{1}{T}\int_{t_0}^{t_0+T} \mathrm{dt}\mathbf{L}(t) \\
\mathbf{L}_F^{(1)}[t_0] &= \frac{1}{2!T}\int_{t_0}^{t_0+T} \mathrm{dt}_1 \int_{t_0}^{t_1} \mathrm{dt}_2[\mathbf{L}(t_1), \mathbf{L}(t_2)] \\
\mathbf{L}_F^{(2)}[t_0] &= \frac{1}{3!T}\int_{t_0}^{t_0+T} \mathrm{dt}_1 \int_{t_0}^{t_1} \mathrm{dt}_2 \int_{t_0}^{t_2} \mathrm{dt}_3[\mathbf{L}(t_1), [\mathbf{L}(t_2), \mathbf{L}(t_3)] + (1 \leftrightarrow 3)\,. \\
& \quad pp
\end{aligned} \tag{4.47}$$

The control appears with the same multiplicity as its attached operator in each non-

vanishing commutator. Note that:

$$\begin{aligned}[\mathbf{L}(t_1), \mathbf{L}(t_2)] &= 4\pi(u(t_2) - u(t_1))[\mathbf{L}_3, \mathbf{L}_1] \\ [\mathbf{L}(t_1), [\mathbf{L}(t_2), \mathbf{L}(t_3)]] &= 8\pi^2(u(t_3) - u(t_2))[\mathbf{L}_3, [\mathbf{L}_3, \mathbf{L}_1]] \\ &\quad + 8\pi u(t_1)(u(t_3) - u(t_2))[\mathbf{L}_1, [\mathbf{L}_3, \mathbf{L}_1]] \,. \end{aligned} \tag{4.48}$$

As such, the desired example expansion is computed to be:

$$\mathbf{L}_F[0] = 2\pi\mathbf{L}_1 + \frac{4\pi v}{k\Omega}\mathbf{L}_2 - \frac{1}{k^2\Omega^2}\left(2\pi(u^2 + 3v^2)\mathbf{L}_1 + 8\pi^2 u\mathbf{L}_3\right) + \mathrm{O}(\Omega^{-3}) \,. \tag{4.49}$$

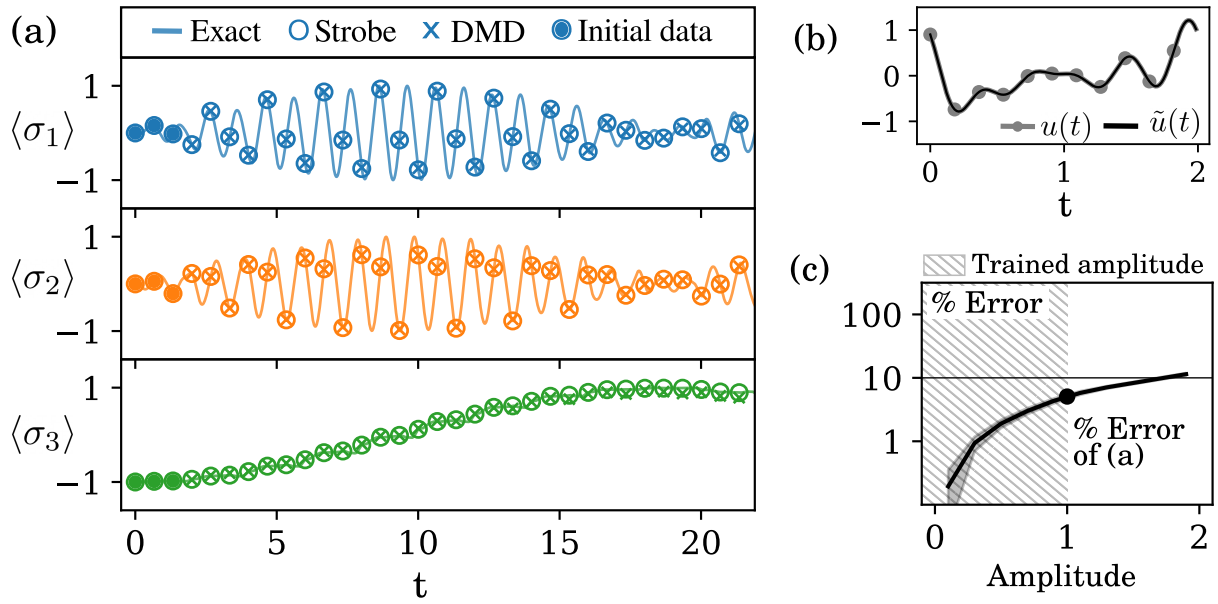


Figure 4.6: Figure 4.6(a) shows the true trajectory compared to the biDMD predictions for a two-level system driven by a random control. Figure 4.6(b) plots the control over a single period and shows an exact match with the truncated Fourier series whose coefficients make up the control input of the biDMD model. Figure 4.6(c) shows the error of (a) and for additional amplitudes of the same control.

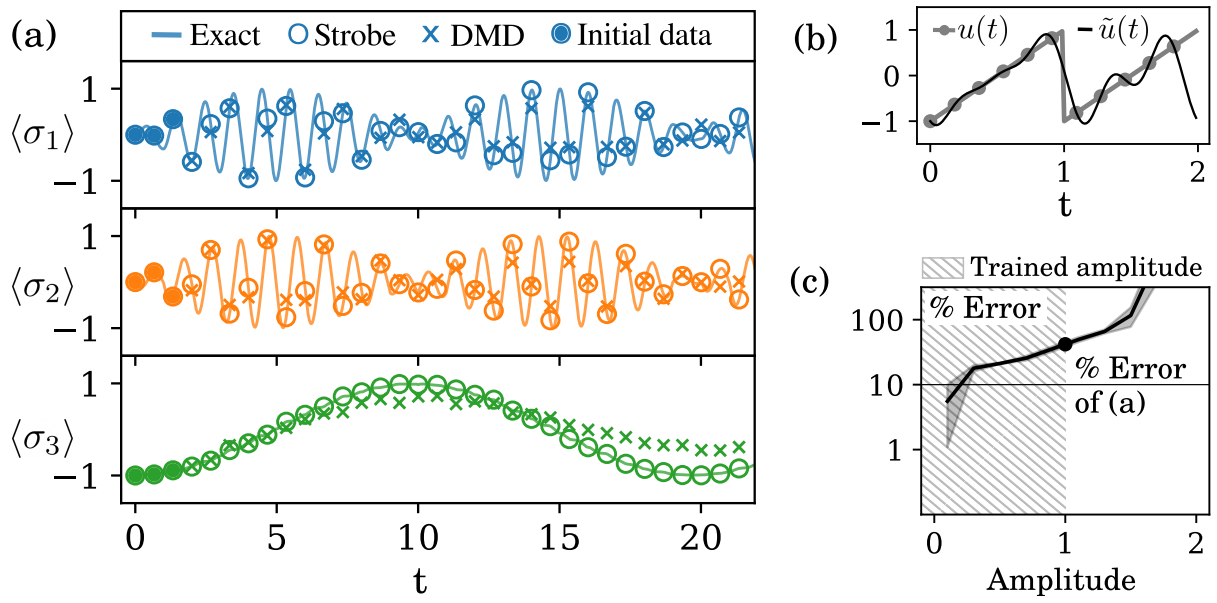


Figure 4.7: Figure 4.7(a) shows the true trajectory compared to the biDMD predictions for a two-level system driven by a sawtooth control. Figure 4.7(b) plots the control over a single period and compares to the approximation via the truncated Fourier series whose coefficients make up the control input of the biDMD model. Figure 4.7(c) shows the error of (a) and for additional amplitudes of the same control.

4.4.2.1 Example 3

Consider again the two-level quantum system with $H(t) = \pi\sigma_3 + u(t)\sigma_1$. Restrict the control to the span

$$u(t) = \sum_{k=1}^{5} a_k \cos(k\Omega t) + b_k \sin(k\Omega t), \quad \Omega = \frac{1}{2}\frac{2\pi}{T} = \pi\,. \tag{4.50}$$

Leveraging the linearity of the biDMD model, assemble the measurement trajectory data as a horizontal stack of numerical experiments driven by pure tones. Suppose the trajectories are measured over 5 periods of applied control in which each of a_k and b_k are taken separately to be $\pm 0, 0.1, \ldots, 1$. Include additive Gaussian noise with $\sigma = 10^{-2}$. From this control data, construct the library snapshot matrix ((4.42)) up to second-order nonlinearities. In the trajectories, we also include additional intra-stroboscopic measurements using the methods of Section 4.4.1. From all of these horizontally-stacked measurement trajectories and the polynomially-extended control coefficients, assemble the snapshot matrices according to ((4.44)), and apply the biDMD algorithm to obtain a model for the system.

In Figures 4.5, 4.6, and 4.7 we look at the predictive capabilities of the biDMD model over 10 control periods (twice the training time) for certain representative control pulses. Part (b) of each figure shows the input control plotted alongside the Fourier-series signal ((4.50)) that best approximates the applied control. Part (c) quantifies the percent error of the measured versus predicted trajectories over a range of amplitudes assuming that the input control pulse remains in a fixed shape.

In particular, Figure 4.5 shows the case of a resonance drive. The model is successful up to a large amplitude limit. This can be understood by recalling that the validity of the Magnus expansion requires a control period that is "small enough"–a condition that exists in an inverse relationship with the amplitude of the applied drive. Figure 4.6 shows a successful prediction for the case of an arbitrary multi-frequency control in the span of the truncated Fourier series (for simplicity, we hold the same coefficients across all periods). Finally, Figure 4.7 shows a sawtooth drive that is only approximately captured by the truncated Fourier series input in the biDMD model; as a result, the model predicts with a reduced accuracy.

4.5 Conclusion

The success of quantum optimal control (QOC) is critically dependent on accurate quantum system identification. In cases where theoretical models are inaccurate or unknown, data-driven system identification is essential for practical QOC. Established ideas from machine learning and regression algorithms like the dynamic mode decomposition have advantages that can be useful to quantum technologies. Like the innovation of the bilinear dynamic mode decomposition (biDMD) in this paper, the ideas must first be adapted to accommodate the underlying mathematics and structure of quantum dynamical systems. Much like its forerunner, the DMDc algorithm [163] for the case of direct actuation in classical control systems, biDMD provides a data-driven and equation-free framework for use with QOC strategies. We have demonstrated the success of the biDMD formulation on a number of quantum systems, showing that it can successfully leverage time-series data alone to enact control.

The use of the DMD framework for quantum systems brings with it a variety of well-established extensions and improvements beyond the naive algorithms introduced in this paper. Studying the ways these improvements fit into the quantum world provides a myriad of future research directions. First, DMD is by construction a method for reduced-order modeling of high-dimensional systems; however, DMD can also accommodate tensor-network representations of high-dimensional data [159]. DMD can also utilize compressive measurements [10, 26]. As such, biDMD should be studied in systems involving multiple qubits under control. Also, time-delays and their connection to random measurements [60, 153, 199] indicate ways DMD can increase system dimensionality in the case of sparse sampling and provide a path for DMD to capture non-Markovian dynamics [53]. Optimization-based DMD algorithms [8, 110] can improve characterization of the DMD system by relaxing data-collection strategies and by incorporating additional physical constraints or modeling assumptions. Such strategies can be used to increase the fidelity of DMD-based models in experimental settings or to adapt the framework to one of discrepancy modeling. Because control is exogenous, feedback

can also be used to increase the DMD fidelity [66]. Ultimately, the goal is to demonstrate experimentally that biDMD provides one path to data-driven and equation-free model-based feedback controllers for engineering future quantum technologies. Moreover, it provides a clear connection to Koopman theory for classical system identification and control, illustrating how nonlinear classical dynamics are linearized via an infinite-dimensional Hilbert space of observables [101, 102, 141].

Code availability

Code supporting the findings of this study is openly available at the following url:

`https://github.com/andgoldschmidt/biDMD-for-quantum`.

Chapter 5

CONTROL THEORY

"A discussion of the nature of any intellectual effort is difficult per se - at any rate, more difficult than the mere exercise of that particular intellectual effort. It is harder to understand the mechanism of an airplane, and the theories of the forces which lift and which propel it, than merely to ride in it, to be elevated and transported by it–or even to steer it."

— John von Neumann, *The Mathematician* [188]

5.1 Controllable systems

A system is controllable if we have complete access to the state space of the system through our choice of inputs. If we conclude that a system is controllable, then the only remaining problem is to find the actual input sequence to realize the intended dynamics (this is the subject of Chapter 6). In this chapter, we review two rank tests for controllability. The first is the *Kalman rank condition* for linear systems, and the second is the *Lie algebra rank condition* for bilinear systems.

We need two concepts in order to discuss controllability of a system. The two concepts address whether we are talking about our ability to control from an initial state or toward a target–equivalently, whether we are talking about the preimage or the image under the controlled dynamics (see Figure 5.1). Suppose we have a generic dynamical system, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ with $\mathbf{x} \in \mathcal{X}$, $\mathbf{u} \in \mathcal{U}$. Also, take a fixed time T. Then:

Definition 5.1.1. Controllable set. Given a target state $\mathbf{x}_{\text{targ.}} \in \mathcal{X}$, an initial state $\mathbf{x}_{\text{init.}} \in \mathcal{X}$ is *T-controllable* to the target if there is an input sequence $\mathbf{u}(\cdot) \in \mathcal{U}[0, T]$ that connects the two points under $\mathbf{f}$. Denote the T-controllable set as the preimage, $\Phi_T^-(\mathbf{x}_{\text{targ.}}, \mathcal{U})$.

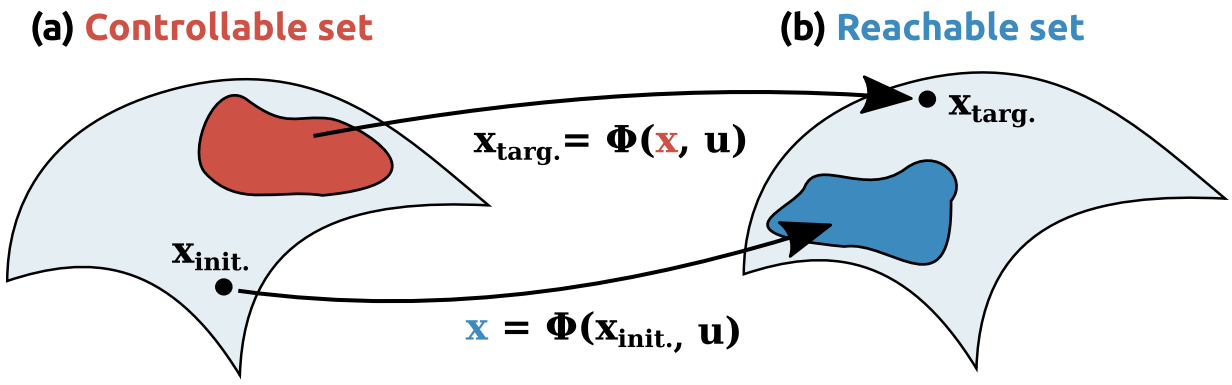


Figure 5.1: **(a)** A given state is controllable to the target if it is in the preimage of the dynamics, and **(b)** a given state is reachable from the initial state if it is in the image of the dynamics.

If this set spans the state space, we say the system has *controllability to* $\mathbf{x}_{\text{targ.}}$.

Definition 5.1.2. Reachable set. Given an initial state $\mathbf{x}_{\text{init.}} \in \mathcal{X}$, a target state $\mathbf{x}_{\text{targ.}} \in \mathcal{X}$ is *T-reachable* from the initial state if there is an input sequence $\mathbf{u}(\cdot) \in \mathcal{U}[0, T]$ that connects the two points under $\mathbf{f}$. Denote the T-reachable set as the image, $\Phi_T^+(\mathbf{x}_{\text{init.}}, \mathcal{U})$.

We can take the union of T-controllable or T-reachable sets over all T; we call these the controllable set, $\Phi^-(\mathbf{x}_{\text{targ.}}, \mathcal{U})$, or reachable set, $\Phi^+(\mathbf{x}_{\text{init.}}, \mathcal{U})$, respectively. If there exists a point for which both the controllable and reachable sets are equal to $\mathcal{X}$, then the system is controllable, in the sense that any two points in $\mathcal{X}$ can be connected by way of this intermediate point. The ideas introduced here can be extended to discrete-time systems in a straightforward way, but we should be cautious about inferring unchecked conclusions about reachable or controllable sets when switching between the two cases[1].

For linear control theory, we use Steve Brunton and Nathan Kutz's expansive *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* [28]. For

[1]For example, in bilinear systems, the controllability can depend on the discretization scheme (see Reference [57], §4.5).

bilinear control theory, we return[2] to Domenico D'Alessandro's *Introduction to Quantum Control and Dynamics* [41]. We also supplement with David Elliott's *Bilinear Control Systems* [57] and Reference [5], an excellent review article on quantum control theory (which also covers the material in Chapter 1). Systems theory often incorporates linear measurements $\mathbf{y} = \mathbf{C}\mathbf{x}$ in addition to the dynamics. This permits a study of the theory of *observability*. We do not focus on observability in this thesis, but this is a natural dual to the idea of controllability, and it is discussed in the references.

5.2 Linear control theory

5.2.1 Discrete time

Suppose we have a discrete, linear, time-invariant control system (D-LTI system),

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t. \tag{5.1}$$

Let $\mathcal{X}$ be the state space with $n = \dim[\mathcal{X}]$. Our goal is to study what makes this a controllable system.

[2] See Chapter 2.

5.2.1.1 Kalman rank condition

Inspect the set reachable from the origin (i.e. $\Phi_T^+(\mathbf{0},\mathcal{U})$). We can iterate through T steps:

$$\mathbf{x}_1 = \mathbf{B}\mathbf{u}_0 \tag{5.2}$$

$$\mathbf{x}_2 = \mathbf{A}\left(\mathbf{B}\mathbf{u}_0\right) + \mathbf{B}\mathbf{u}_1 \tag{5.3}$$

$$\mathbf{x}_3 = \mathbf{A}\left(\mathbf{A}\mathbf{B}\mathbf{u}_0 + \mathbf{B}\mathbf{u}_1\right) + \mathbf{B}\mathbf{u}_2 \tag{5.4}$$

$$\vdots$$

$$\mathbf{x}_T = \begin{bmatrix}\mathbf{A}^{T-1}\mathbf{B} & \mathbf{A}^{T-2}\mathbf{B} & \cdots & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{u}_0\\ \mathbf{u}_1\\ \vdots\\ \mathbf{u}_{T-1}\end{bmatrix} \tag{5.5}$$

Because the control inputs are free parameters, $\Phi_T^+(\mathbf{x})$ is determined by the rank of the matrix

$$\mathcal{C}_T(\mathbf{A},\mathbf{B}) \equiv \begin{bmatrix}\mathbf{A}^{T-1}\mathbf{B} & \mathbf{A}^{T-2}\mathbf{B} & \cdots & \mathbf{B}\end{bmatrix}. \tag{5.6}$$

The reachable set is the image of this matrix acting on $\mathcal{U}^T$, $\Phi_T^+(\mathbf{0},\mathcal{U}) = \mathrm{Im}[\mathcal{C}_T(\mathbf{A},\mathbf{B})]$. Because of the Cayley-Hamilton theorem[3], we know that for finite-dimensional systems we only have to check up to $T = n = \dim[\mathcal{X}]$ to determine the reachable set [28].

Now let's consider the controllable set. Given the origin as a target state, an initial state $\mathbf{x}_0$ is controllable to the origin if there is a control sequence in $\mathcal{U}^n$ such that

$$\mathbf{0} \stackrel{!}{=} \mathbf{x}_n = \mathbf{A}^n\mathbf{x}_0 + \mathcal{C}_n(\mathbf{A},\mathbf{B})\begin{bmatrix}\mathbf{u}_0\\ \mathbf{u}_1\\ \vdots\\ \mathbf{u}_{n-1}\end{bmatrix} \tag{5.7}$$

[3]Every matrix $\mathbf{A}$ satisfies its own characteristic polynomial, i.e. $p(\lambda) = \det[\mathbf{A} - \lambda\mathbb{1}]$ is trivially zero for $\lambda \to \mathbf{A}$. Note $p(\lambda)$ is an order $n = \dim[\mathbf{A}]$ polynomial in λ; using $p(\mathbf{A}) = 0$, we can write $\mathbf{A}^n$ in terms of its lower powers.

This tells us that an initial state $\mathbf{x}_0 \in \Phi_n^-(\mathbf{0})$ only if $\text{Im}[\mathbf{A}^n] \subseteq \text{Im}[\mathcal{C}_n(\mathbf{A}, \mathbf{B})]$. Therefore, if the reachable set from the origin is $\mathcal{X}$, then every point is in the controllable set from the origin:

$$\mathcal{X} \text{ is the } \textit{D-LTI reachable set} \Rightarrow \mathcal{X} \text{ is the } \textit{D-LTI controllable set}.$$

The converse does not hold; for instance, if $\mathbf{A} = 0$, then every state is in the controllable set, unconditional on whether that state is in the reachable set. We can fix this if $\mathbf{A}$ is full-rank. In this case, $\text{Im}[\mathbf{A}^n] = \mathcal{X}$, and we have

$$\mathcal{X} \text{ is the } \textit{D-LTI reachable set} \Leftrightarrow \mathcal{X} \text{ is the } \textit{D-LTI controllable set},$$

where "from/to $\mathbf{0}$" is implied. We now state the main result of this section:

Theorem 5.2.1. Kalman rank condition. If $\mathcal{C}_n(\mathbf{A}, \mathbf{B})$ is full rank, then the D-LTI system in Equation (5.1) is controllable.

Because of its fundamental role in D-LTI analysis, $\mathcal{C}_n(\mathbf{A}, \mathbf{B})$ is known as the *controllability matrix* in the literature. Theorem 5.2.1 can be extended to the case of continuous dynamics with a few small changes [28].

5.3 Bilinear control theory

As discussed in Chapter 2, quantum control systems are bilinear. Recall from Equation (1.24) that the continuous, time-invariant, quantum control dynamics are provided via two versions of the Schrödinger equation. There is one for the state, and one for the operator:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = (\hat{H}_0 + u(t)\hat{H}_1) |\psi(t)\rangle , \quad i\hbar \frac{d}{dt} U(t) = (\hat{H}_0 + u(t)\hat{H}_1)U(t). \tag{5.8}$$

We will use *state-controllable set* and *state-reachable set* for the former and *operator-controllable set*, and *operator-reachable set* for the latter. The system is still called controllable if any two states can be connected by an input sequence. In Equation (5.8), we have assumed that the system is closed. In this case, the matrices $-i\hat{H}_j$ are skew-Hermitian (length-preserving for pure states, $|\psi\rangle$). It's worth noting that if we relax the control dynamics

to the Lindblad case (Equation (1.40)), then the dissipative part can prevent $-i\hat{H}_0$ from being skew-Hermitian, which complicates the analysis[4].

5.3.1 Discrete time

Consider the discrete, bilinear, time-invariant control systems (D-BTI),

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \sum_{j=1}^{M} u_{j,t}\mathbf{B}_j\mathbf{x}_t. \tag{5.9}$$

Suppose we proceed iteratively, like in Section 5.2.1.1. If we naively follow the previous prescription and look for the reachable set from a state at the origin, we find $\mathbf{x}_1 = (\mathbf{A} + u_0\mathbf{B})\,\mathbf{0} = \mathbf{0}$. Clearly, this is not a reasonable starting point for bilinear systems. Moreover, for quantum systems $|\psi\rangle \neq 0$ because the zero state is not normalized. Unlike the linear case, there does not seem to be an obvious initial state to choose. However, there is a natural starting point for the Schrödinger equation for the propagator: the identity operator. Let $\mathbf{X} \in SU(n)$ denote the elements of this unitary space[5]. Note that

$$SU(n) \text{ is the } \textit{operator-reachable set} \Leftrightarrow SU(n) \text{ is the } \textit{operator-controllable set}$$

To illustrate, consider "⇒": suppose we are given an initial $\mathbf{X}_{\text{init.}}$. The state $\mathbf{X}_{\text{init.}}^{\dagger}$ is reachable from $\mathbb{1}$, so we can send $\mathbf{X}_{\text{init.}}$ to $\mathbb{1}$ by applying the controls to reach $\mathbf{X}_{\text{init.}}^{\dagger}$. It's also worth remarking that if $SU(n)$ is the operator-reachable set, then the system is controllable because $SU(n)$ includes the unitary dynamics connecting any two states. Therefore, we are interested in finding a necessary and sufficient condition for the operator-reachable set to be $SU(n)$.

---

[4]The tools change from Lie algebras to Lie wedges, and Lie groups to Lie semigroups, e.g. Reference [114] §27

[5]The unitary matrices are the group $U(n) = \{\mathbf{X} \in \mathbb{C}^{n\times n} : \mathbf{X}^{\dagger}\mathbf{X} = \mathbb{1}$. In the special unitary subgroup $SU(n) \leq U(n)$, the unphysical global phase is removed: $SU(n) = \{\mathbf{X} \in \mathbb{C}^{n\times n} : \mathbf{X}^{\dagger}\mathbf{X} = \mathbb{1}, \det[\mathbf{X}] = 1\}$.

The T-step reachability from $\mathbb{1}$ is

$$\mathbf{X}_1 = (\mathbf{A} + u_0\mathbf{B})\mathbb{1} \tag{5.10}$$

$$\mathbf{X}_2 = (\mathbf{A} + u_1\mathbf{B})(\mathbf{A} + u_0\mathbf{B}) = \mathbf{A}^2 + u_0\mathbf{AB} + u_1\mathbf{BA} + u_1u_0\mathbf{B}^2 \tag{5.11}$$

$$\vdots$$

$$\begin{aligned}\mathbf{X}_T &= (\mathbf{A} + u_{T-1}\mathbf{B})(\mathbf{A} + u_{T-2}\mathbf{B})\cdots(\mathbf{A} + u_0\mathbf{B}) \\ &= \mathbf{A}^T + u_{T-2}\mathbf{ABA}^{n-2} + \cdots + u_0\mathbf{A}^{T-1}\mathbf{B} + \cdots + (\prod_{j=0}^{T-1} u_j)\mathbf{B}^n.\end{aligned} \tag{5.12}$$

How do we analyze the operator-reachable set as specified by Equation (5.12)? It is a complicated multilinear function of the inputs, especially troubled by the presence of the fixed drift term with no control coefficients. The way forward is to suppose that our piecewise-constant controls are an approximation of some continuous control, and then to study the continuous case. We do this in the next section. Working ahead, let's consider such an approximation starting at Equation (5.10): if $\mathbf{A} = \mathbb{1}$, we have an Euler discretization, so $\mathbf{X}_1 = \mathbb{1} + \mathrm{d}t u_0\mathbf{B} = \mathbb{1} + \frac{\mathrm{d}t}{k}(k\,u_0)\mathbf{B}$. In this way, the dynamics can be made to take arbitrarily small timesteps by leveraging the control amplitudes. There is no speed limit to control. Meanwhile, with a nonzero drift, we are simply stuck with its effects. Even if a system is controllable, we are not guaranteed anything about the paths that connect each state, nor whether the state that is reached can be maintained (see Example 5.13).

5.3.2 Continuous time

Consider the continuous, bilinear, time-invariant control system (C-BTI),

$$\frac{d}{dt}\mathbf{x}(t) = (\mathbf{A} + \sum_{j=1}^{J} u_j(t)\mathbf{B}_j)\mathbf{x}(t) \equiv \mathbf{P}(\mathbf{u})\mathbf{x}(t) \tag{5.14}$$

Example 12: Walking on a clock

Non-zero drift in systems can make control more challenging. Imagine walking along the second hand of a giant clock. We can adjust our radial position while the second hand smoothly rotates. This situation can be modeled by a bilinear control system with nonzero drift,

$$\frac{d}{dt}\mathbf{x} = \frac{2\pi}{60}\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\mathbf{x} + u(t)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\mathbf{x}. \tag{5.13}$$

This system is controllable, but a given angle can only be attained once per minute because of the constant drift of the second hand. There is a speed limit to the control. There is a term for what's missing: a system is *small-time locally controllable*[a] if a neighborhood of the state is always reachable, $\forall t > 0$, $\exists \epsilon > 0$: $N_\epsilon(\mathbf{x}_{\text{init.}}) \subset \Phi_t^+(\mathbf{x}_{\text{init.}})$. Equation (5.13) is a controllable system, but not small-time locally controllable. We can see this affect the discretized dynamics, too. If we perform an Euler discretization in polar coordinates, then $\theta_{t+1} = \theta_t + \mathrm{dt}\frac{2\pi}{60}\theta_t$. Setting $\mathrm{dt} = 1$ and starting at midnight, we can only reach the rays of states along each of the second marks.

[a] For instance, see Reference [111] §15. This idea is also connected to time-optimal quantum control.

with $\mathbf{x} \in \mathbb{C}^n$. There is an equivalent system describing the dynamics of the propagator,

$$\frac{d}{dt}\mathbf{X}(t) = \mathbf{P}(\mathbf{u})\mathbf{X}(t) \tag{5.15}$$

The matrices $\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_J$ are assumed to be skew-symmetric; so is their linear combination over real inputs, $\mathbf{P}(\mathbf{u})$. Let $\mathcal{U}$ be the set of real-valued admissible controls.

5.3.2.1 Lie algebra rank condition

The key to this section is the *Lie algebra* generated by the matrices $\mathbf{P}(\mathbf{u})$, denoted by $\mathcal{L} \equiv \{\mathbf{P}(\mathbf{u})\}_{\text{Lie}}$.

Definition 5.3.1. Lie algebra. A *Lie algebra*, $\mathcal{L}$ is a vector space over a field $\mathbb{F}$ with an additional binary operation called the *Lie bracket*. For Lie algebras of matrices, the Lie bracket is given by the familiar commutator, $[\mathbf{X}, \mathbf{Y}] = \mathbf{XY} - \mathbf{YX}$ for all $\mathbf{X}, \mathbf{Y} \in \mathcal{L}$.

Remark 5.3.3. The cross product in $\mathbb{R}^3$ is an example of a Lie bracket. One important fact about the cross product is it produces a vector which points orthogonal to the original two vectors in the cross product. In general, if we iteratively apply the Lie bracket to a given set of generators, then we may produce new elements that increase the span of the set.

We will show that if the dimension of $\{\mathbf{P}(\mathbf{u})\}_{\text{Lie}}$ as a vector space over $\mathbb{C}$ is n^2 or n^2-1, then the system defined in Equation (5.14) is controllable. This is the *Lie algebra rank condition*. Following the same logic as Section 5.3.1, it is sufficient for the system to be controllable if the reachable set from $\mathbb{1}$ contains all unitary matrices, i.e. $U(n)$ or $SU(n) \subseteq \Phi^+(\mathbb{1})$. For piecewise-constant controls, the reachable set is the product of elements of the form $e^{\mathbf{P}(u)t}$, i.e.

$$\Phi^+(\mathbb{1}) = \{e^{\mathbf{P}(u_1)t_1} e^{\mathbf{P}(u_2)t_2} \cdots e^{\mathbf{P}(u_r)t_r} : t_1, \ldots, t_r \in \mathbb{R},\ \mathbf{u}_1, \ldots, \mathbf{u}_r \in \mathcal{U}^J\}. \tag{5.16}$$

Given a Lie algebra $\mathcal{L}$, we can define a similar looking set

$$e^{\mathcal{L}} \equiv \{e^{A_1} e^{A_2} \cdots e^{A_\ell} : A_1, \ldots, A_\ell \in \mathcal{L}\}. \tag{5.17}$$

By inspection of these two sets, it is evident that $\Phi^+(\mathbb{1}) \subseteq e^{\mathcal{L}}$ when $\mathcal{L}$ is the Lie algebra generated by the matrices $\mathbf{P}(\mathbf{u})$.

Remark 5.3.4. By Theorem 3.2.6 in Reference [41], it is in fact the case that $\Phi^+(\mathbb{1})$ is dense in $e^{\mathcal{L}}$. If $e^{\mathcal{L}}$ is a compact set, then the two sets are equal.

It turns out that $e^{\mathcal{L}}$ is what is known as a *Lie group*, and the fact that a Lie algebra $\mathcal{L}$ can generate a Lie group in this way is called the *Lie correspondence*.

Definition 5.3.2. Lie group. A *Lie group* is a group that is also a differentiable manifold. This means that in addition to assigning a manifold to the group elements, the group operations (multiplication, inverse) must be analytic.

Theorem 5.3.3. Lie correspondence. A given Lie algebra $\mathcal{L}$ generates a unique Lie group $e^{\mathcal{L}}$. Moreover, a given Lie group can be associated to a unique Lie algebra[6].

Example 13 illustrates the fact that the Lie algebra of traceless skew-Hermitian matrices, $su(n)$, generates the Lie group of special unitary matrices, $SU(n) = e^{su(n)}$. Now, applying Remark 5.3.4, we have that if $\{\mathbf{P}(\mathbf{u})\}_{\text{Lie}}$ spans $su(n)$, then $\Phi^+(\mathbb{1}) = SU(n) = e^{su(n)}$. We now restate the main result:

Theorem 5.3.4. Lie algebra rank condition. If the *dynamical Lie algebra* for Equation (5.14) given by $\{\mathbf{P}(\mathbf{u})\}_{\text{Lie}}$ has rank $n^2 = \text{rank}(u(n))$ or $n^2 - 1 = \text{rank}(su(n))$ as a vector space, then the operator-reachable set from $\mathbb{1}$ is $U(n)$ or $SU(n)$, respectively. This is sufficient for the system in Equation (5.14) to be controllable.

5.3.4.1 Universal gates

In Section 5.3.1, we motivated spending our efforts on the case where $SU(n)$ is the *operator-reachable set* because this is sufficient for the system to be controllable. However, we ignored the possible case where the system is controllable, but the operator-reachable set is something

[6]The associated Lie algebra is equivalent to the tangent space of the Lie group at the identity. Informally, see Example 13. For additional discussion, see Reference [41] §3.1.

Example 13: The Lie algebra of the special unitary group

The Lie Group

$$SU(N) = \{U \in \mathbb{C}^{N\times N} \,:\, U^\dagger U = \mathbb{1}\,, \det U = 1\}\,. \tag{5.18}$$

is known as the special unitary group. It is a subgroup of $U(n)$, the determinant-one $n \times n$ unitary matrices, and $GL(n)$, the determinant-one matrices. In this example, we show how this group naturally induces the constraints of the Lie algebra of skew-Hermitian matrices,

$$su(N) = \{U \in \mathbb{R}^{N\times N} \,:\, U^\dagger + U = 0\,, \mathrm{Tr}[U] = 0\}\,. \tag{5.19}$$

using the fact that *the Lie algebra is the tangent space of the Lie group at the identity.* There is a neighborhood around the identity contained in the Lie group which can be expressed $1 + \mathrm{d}A \in SU(N)$. The key step is to note the correspondence of elements of the Lie algebra with the tangent vectors, $\mathrm{d}A$, and to use that these tangent vectors are infinitesimal, $\mathrm{d}A^2 = 0$.

First, consider the unitary condition,

$$\mathbb{1} = U^\dagger U = (\mathbb{1} + \mathrm{d}A^\dagger)(\mathbb{1} + \mathrm{d}A) = \mathbb{1} + \mathrm{d}A^\dagger + \mathrm{d}A \tag{5.20}$$

so $\mathrm{d}A$ is skew-Hermitian.

Next, consider the determinant condition, so

$$1 \stackrel{!}{=} \det \begin{pmatrix} 1 + \mathrm{d}A_{11} & \mathrm{d}A_{12} & \cdots & \mathrm{d}A_{1N} \\ \mathrm{d}A_{21} & 1 + \mathrm{d}A_{12} & \cdots & \mathrm{d}A_{2N} \\ \vdots & & & \vdots \\ \mathrm{d}A_{N1} & \mathrm{d}A_{N2} & \cdots & 1 + \mathrm{d}A_{NN} \end{pmatrix} \tag{5.21}$$

$$= 1 + \mathrm{d}A_{11} + \mathrm{d}A_{22} + \cdots + \mathrm{d}A_{NN} \tag{5.22}$$

hence $\mathrm{d}A$ is traceless. In the last step, any term that isn't on the main diagonal must contain some factor $\mathrm{d}A_{ij}\mathrm{d}A_{k\ell} = 0$, and so the only non-vanishing terms arise from $\prod_j (1 + \mathrm{d}A_{jj})$.

less than $SU(n)$ [7]. Another possible case comes from the field of quantum computation, because–with its finite gate sets–we only need a few specific operations to belong to the reachable set. It turns out that, for the Schrödinger equation, $SU(n)$ is almost always the operator-reachable set.

Theorem 5.3.5. A random pair of gates is almost surely universal.

Proof. For a discussion of this result, see Reference [41] §3. □

[7] See Reference [41] §3.3 for an in-depth discussion of this case.

Chapter 6

OPTIMAL CONTROL

"Numerical solution was considered the last resort of an incompetent mathematician. The opposite, of course, is true. Once working in the area, it is very quickly realized that far more ability and sophistication is required to obtain a numerical solution than to establish the usual existence and uniqueness theorems. It is far more difficult to obtain an effective algorithm than one that stops with a demonstration of validity."

— Richard Bellman, *Eye of the Hurricane: An Autobiography*

Optimal control is a practical way to realize quantum control for experiments. In optimal control, the goal is to find actions which minimize the score of some user-defined task. The actions are called *open-loop* if they rely on the current system state and future predictions, or *closed-loop* if they rely on only the current state. In open-loop control, a model of the system is used to look ahead and devise a plan of action. In closed-loop control, the best action can be decided entirely from the current information. Closed-loop control usually involves measuring and modifying the system online, although it can also be used in concert with the equations of motion to design an open-loop control. In quantum control, the complete state information used to apply traditional closed-loop control is rendered unavailable because any measurement will disturb the system [1]. Therefore, most quantum control strategies are open-loop. We review methods for open-loop control in Section 6.1.

The success of open-loop control depends entirely on the accuracy of the model being used for planning. With a perfect model, there is little need to involve ourselves with closed-loop

[1] Quantum feedback control can still be implemented with suitable modifications. One way is to use couple a second quantum system to the original system, and have the dynamics of the second system regulate the original. Another way is by taking a weak measurement where the system is minimally disturbed. The weakly-measured system evolves as a stochastic process, and the information from the weak measurement is immediately used to close the loop [195].)

control outside of perhaps computational efficiency. The perfect model can make perfect predictions about system behavior, and hence derive perfect control laws before running a single experiment. In the grit of the real world, a mischaracterization or disturbance can result in the failure of the open-loop plan. We can attempt to dig our way out of this problem by being even more careful about characterization and robustness properties of the controller. Alternatively, we can incorporate feedback into open-loop control by running the experiment in iterations. We discuss this idea in Section 6.2

6.1 Approaches to open-loop optimal control

Historically, optimal control problems were solved using the calculus of variations. The two main directions, the *Hamilton-Jacobi-Bellman equation* and *Pontryagin's principle*, are concerned with finding the minimum of a functional[2]: that is, a function from a Hilbert space of functions to its underlying field. The functional that both methods seek to minimize is called the cost [3], and it scores the success of some task. The cost varies over the space of control functions used to accomplish the task.

In this thesis, we use measurement data and computational simulations, so we consider a discrete formulation of this optimal control problem. Quantum optimal control as a discrete-time problem begins with a standard cost function,

$$J(\mathbf{x}_{1:T+1}, u_{1:T}) = \underbrace{\sum_{t=1}^{T} \ell(\mathbf{x}_t, \mathbf{u}_t, t)}_{\text{Running cost}} + \underbrace{m(\mathbf{x}_{T+1})}_{\text{Final cost}}, \tag{6.1}$$

with $\mathbf{x}_{1:T} \equiv \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T+1}\}$ and $\mathbf{u}_{1:T} \equiv \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_T\}$, which is constrained by the

[2]The solution of the Hamilton-Jacobi-Bellman equation is closed-loop in the sense that the control law can be computed by using a global *value function* defined over all states. Pontryagin's principle is often more tractable; the solution is open-loop control because it depends on the dynamics and relies on an initial and final boundary condition. For a review, see e.g. Reference [119].

[3]Or reward, for the optimist.

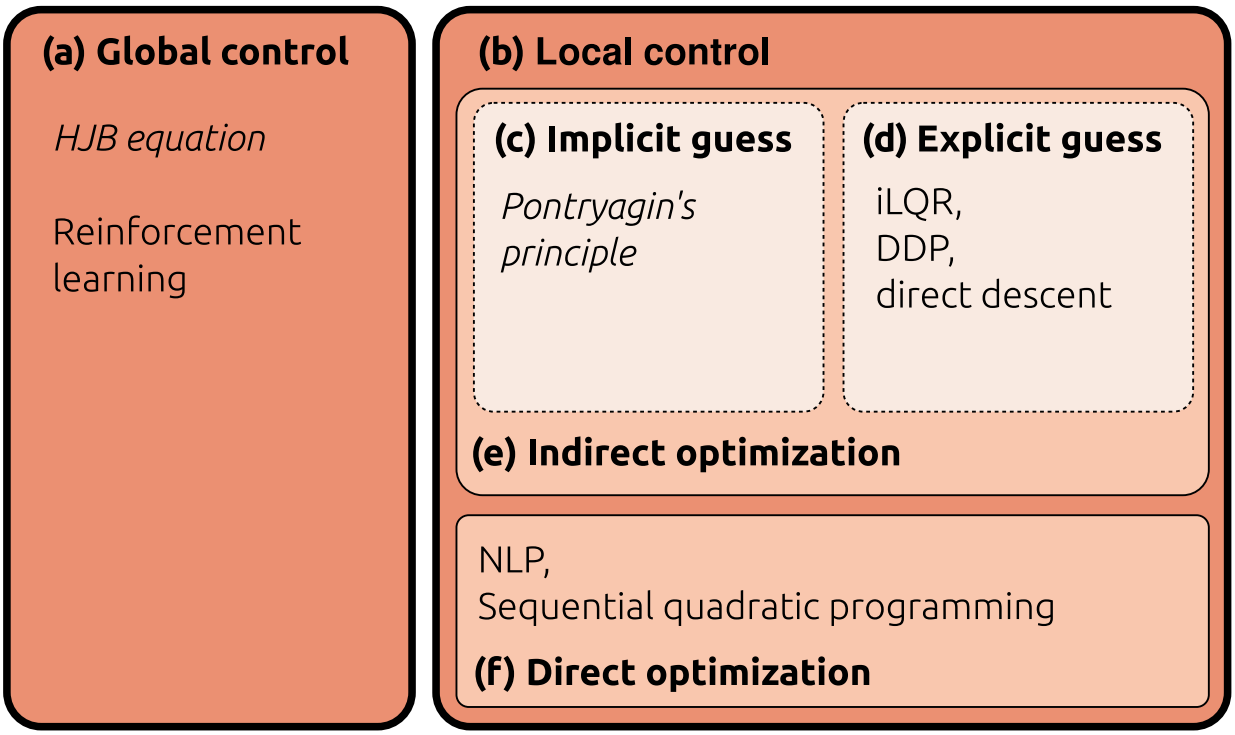


Figure 6.1: Historically, the continuous-time optimal control problem was solved either by the Hamilton-Jacobi-Bellman (HJB) equation or Pontryagin's principle [119]. Solving the HJB equation provides a *global* solution based on the value function, while using Pontryagin's principle involves constructing necessary conditions found by *locally* expanding about the optimal state and control trajectories. This demonstrates how optimal control can be divided into **(a)** *global* methods, and **(b)** *local* methods. Local methods start from a nominal (or guess) trajectory for the state or control and use successive approximations to find the solution. In Pontryagin's principle, these nominals are **(c)** *implicitly* the optimal trajectories, and necessary equations are derived under these conditions. Alternatively, the user can be asked to **(d)** *explicitly* supply the nominal trajectories. Some examples of this latter case include iterative linear-quadratic methods (iLQR) [183], differential dynamic programming (DDP) [135, 181], and direct descent [43, 97]. Whether explicit or implicit, these are all examples of **(e)** *indirect* methods in which the dynamics have been used to eliminate the optimization over the state variables. In **(f)** *direct* methods, we still explicitly supply nominals to localize the solution, but we treat both the state and control as optimization variables. A direct problem is solved using standard nonlinear programming (NLP) with the dynamics as a constraint [77].

bilinear quantum control dynamics,

$$\mathbf{x}_{n+1} = \left(\mathbf{A} + \sum_{j=1}^{J} u_{n,j} \mathbf{B}_j \right) \mathbf{x}_n. \tag{6.2}$$

We may also have additional state or input constraints, e.g. $\mathbf{g}_n(\mathbf{x}_n, \mathbf{u}_n) \leq \mathbf{0}$ and $\mathbf{h}_n(\mathbf{x}_n, \mathbf{u}_n) = \mathbf{0}$, $\forall n$. As shown in Figure 6.1, there are many ways to solve this kind of open-loop optimal control problem. At the top level, these divide into global methods like the HJB equation and local methods like Pontryagin's principle. In this thesis, we focus on *local* methods[4], also known as trajectory optimization methods. These are open-loop solutions to the optimal control problem that start with a nominal (or guess) trajectory for the state or control and use successive approximations about these nominals to find the control which yields the optimal cost. Local methods provide a navigation protocol through large state spaces; they are not exposed to the curse-of-dimensionality in the sense that afflicts global methods.

6.1.1 Indirect optimization

The most common algorithms for quantum optimal control are based on a type of indirect optimization algorithm called direct descent [68]. These methods (e.g. GRAPE [43, 97], Krotov [155], DYNAMO [129]) iteratively update the control inputs by computing descent directions for the cost function. In direct descent, the dynamics are used to eliminate the state trajectory as a variable, so that the cost is a function of only the control trajectory, i.e. $J(\mathbf{x}, \mathbf{u}_{1:T})$ for initial state $\mathbf{x}$. Direct descent is just one kind of indirect optimization, as illustrated by Figure 6.1. To make this point more concrete, Example 15 compares direct descent and iterative LQR for a one-dimensional system with a single control input.

6.1.2 Direct optimization

Direct optimization is implemented using nonlinear programming (NLP) directly in the joint $(\mathbf{x}, \mathbf{u})$ space of states and controls. In pseudocode, NLP means solving the following

[4]Global methods based on deep reinforcement learning can require a great deal of data for training, but have been successfully applied to the quantum control problem, e.g. [13, 30, 42, 150]

Example 14: The Bellman principle

The *value function* at time n reports the optimal cost for Equation (6.1) under the dynamics[a] in Equation (6.2),

$$V^*(\mathbf{x}_n, n) \equiv \min_{\mathbf{u}_{n:T}} J(\mathbf{x}_{n:T+1}, \mathbf{u}_{n:T}, n) \equiv \min_{\mathbf{u}_{n:T}} \left[\sum_{t=n}^{T} \ell(\mathbf{x}_t, \mathbf{u}_t, t) + m(\mathbf{x}_{T+1}) \right]. \tag{6.3}$$

The boundary condition at the final time is $V^*(\mathbf{x}, T+1) = m(\mathbf{x})$. In order to purchase brevity, we have abused notation somewhat by defining the cost from time n as $J(\cdot, \cdot, n)$. We can write the value function in a recursive way using the *Bellman principle*[b],

$$V^*(\mathbf{x}, n) = \min_{\mathbf{u}} \underbrace{\ell(\mathbf{x}, \mathbf{u}, n)}_{\text{current cost}} + \underbrace{V^*(\mathbf{x}', n{+}1)}_{\text{cost-to-go}} \tag{6.4}$$

with $\mathbf{x}'$ as a shorthand for the next step of the dynamics.

---

[a] Value functions are always defined with respect to a dynamical process, $\mathbf{f} : \mathbf{x}, \mathbf{u} \mapsto \mathbf{x}'$, and a policy for action selection, $\pi : \mathbf{x} \mapsto \mathbf{u}$; as such, write $V^{\pi}_{\mathbf{f}}(\mathbf{x}, n)$. Constrained by this process and policy, the value function only depends on an initial state. If the policy is ignored for just the initial selection, write $Q^{\pi}_{\mathbf{f}}(\mathbf{x}, \mathbf{u}, n)$. This is called the quality function, or state-action value function. In either case, the reference to the dynamical process is usually suppressed, and the optimal policy is denoted by $*$, as in Equation (6.3). Following this discussion, we see that the generic cost, J, is the value function without either constraint. As such, it is defined over both a state and control trajectory, as in Equation (6.1). For more on value and quality functions, see Reference [179].

[b] The original statement by Richard Bellman was: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." [15]

Example 15: Indirect optimization for a one-dimensional system

Consider the following one-dimensional control problem:

$$J(x_{1:T+1}, u_{1:T}) = \sum_{t=1}^{T} \ell(x_t, u_t) + m(x_{T+1})$$
$$\text{subject to} \quad x_{t+1} = f(x_t, u_t). \tag{6.5}$$

In this example, we explore two indirect methods for optimal control: direct descent and iterative LQR. Because these are indirect methods, we begin with a nominal input, $\bar{u}_{1:T}$, and initial condition, x. After substituting in the nominal dynamics, the cost becomes a function of the control trajectory alone, i.e. $J(x, u_{1:T})$.

Option 1: Direct descent. We compute the gradient of the control trajectory with respect to the cost, $J(x_1, \bar{u}_{1:T})$, for an arbitrary time slice:

$$\frac{d}{du_j} J(x, \bar{u}_{1:T}) = \frac{d}{du_j} \left(\sum_{t=1}^{T} \ell(\bar{x}_t, \bar{u}_t) + m(\bar{x}_{T+1}) \right) \tag{6.6}$$
$$= \ell_u(\bar{x}_j, \bar{u}_j) + \sum_{t=j+1}^{T} \ell_x(\bar{x}_t, \bar{u}_t) \frac{\partial \bar{x}_t}{\partial u_j} + m_x(\bar{x}_{T+1}) \frac{\partial \bar{x}_{T+1}}{\partial u_j} \tag{6.7}$$

where

$$\frac{\partial \bar{x}_t}{\partial u_j} = f_x(\bar{x}_{t-1}, \bar{u}_{t-1}) f_x(\bar{x}_{t-2}, \bar{u}_{t-2}) \cdots f_x(\bar{x}_{j+1}, \bar{u}_{j+1}) f_u(\bar{x}_j, \bar{u}_j). \tag{6.8}$$

Now, we update each control time slice using gradient descent, and find new marginals according to

$$u_j = \bar{u}_j - \epsilon \frac{d}{du_j} J(x, \bar{u}_{1:T}), \quad j = 1, 2, \ldots, T \tag{6.9}$$
$$x_{t+1} = f(x_t, u_t) \tag{6.10}$$

where $\epsilon > 0$ is the step size and $x_1 = x$. This process is iterated until convergence.

Option 2: iLQR. In iterative LQR [180], we start by recalling Bellman's principle (see Example 14) for the optimal value function (suppress the $*$ for brevity), i.e.

$$V(x,n) = \min_u \ell(x,u) + V(f(x,u),n{+}1) \tag{6.11}$$

with $V(x,T+1) = m(x)$. Next, we expand the argument, $\ell(x,u) + V(f(x,u),n{+}1)$, about the nominal trajectories. To simplify the notation, define $\bar{\ell} \equiv \ell(\bar{x},\bar{u})$ and $\bar{V}' \equiv V(f(\bar{x},\bar{u}),n{+}1)$. Now, we keep all second order terms that are linear in the dynamics,

$$\begin{aligned}&\ell(\bar{x}+\delta x,\bar{u}+\delta u) + V(f(\bar{x}+\delta x,\bar{u}+\delta u),n{+}1)\\ &\quad\approx \bar{\ell} + \bar{V}' + Q_x\delta x + Q_u\delta u + Q_{xx}(\delta x)^2 + Q_{xx}\delta x\delta u + Q_{uu}(\delta u)^2\end{aligned} \tag{6.12}$$

where

$$\begin{aligned}&Q_x = \bar{\ell}_x + \bar{V}_x'\bar{f}_x, \qquad Q_u = \bar{\ell}_u + \bar{V}_u'\bar{f}_u,\\ &Q_{xx} = \bar{\ell}_{xx} + \bar{V}_{xx}'\bar{f}_x^2, \quad Q_{ux} = \bar{\ell}_{xu} + \bar{V}_{xu}'\bar{f}_x\bar{f}_u, \quad Q_{uu} = \bar{\ell}_{uu} + \bar{V}_{uu}'\bar{f}_u^2.\end{aligned}$$

We can use this expression to compute the first-order necessary condition for the control differential to be at a minimum, $\delta u^* = (Q_u + Q_{ux}\delta x)/Q_{uu}$. The Q's must be computed on a backward pass from $T+1$ to 1 due to the recursive $\bar{V}'$ terms. Then, a forward pass can be used to update the marginals starting from $x_1 = x$,

$$u_{t+1} = \bar{u}_t + (Q_u + Q_{ux}(x_t - \bar{x}_t))/Q_{uu} \tag{6.13}$$

$$x_{t+1} = f(x_t,u_t). \tag{6.14}$$

This process is iterated until convergence.

It can be shown from this starting point that gradient descent and iLQR share similar convergence properties. For an analysis showing the shared convergence properties of Newton's method and differential dynamic programming, consult Reference [147].

optimization problem:

$$\begin{aligned} \text{minimize} \quad & J(\mathbf{x}_{1:T+1}, \mathbf{u}_{1:T}) \\ \text{subject to} \quad & \mathbf{g}_t(\mathbf{x}_t, \mathbf{u}_t) \leq \mathbf{0} \\ & \mathbf{h}_t(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{0} \end{aligned} \tag{6.15}$$

Here, J is a possibly nonlinear cost, and $\mathbf{g}$ and $\mathbf{h}$ are possibly nonlinear constraints. This is quite general, so existing NLP software can be grabbed off-the-shelf and immediately brought to bear on our quantum optimal control problem. Moreover, constraints are built into the problem statement of this framework. The catch is that we inherit any downsides which emerge from solving challenging NLP problems.

Suppose we are studying a quantum control system, $i\hbar\frac{d}{dt}|\psi\rangle = H(u(t))|\psi\rangle$. Consider optimizing for the preparation of a pure quantum state, $|\psi_{\text{targ.}}\rangle$. A reasonable cost function is the overlap between these two vectors,

$$|\langle\psi|\psi_{\text{targ.}}\rangle|^2 = \langle\psi|\left(|\psi_{\text{targ.}}\rangle\langle\psi_{\text{targ.}}|\right)|\psi\rangle \equiv \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} \tag{6.16}$$

where $|\psi\rangle \mapsto \mathbf{x}$ is the d-dimensional vector of coefficients in $\mathbb{C}^d$ or, under the usual transformation, in $\mathbb{R}^{2d}$. If we also penalize the energy of the control, we have the familiar quadratic cost function,

$$J(\mathbf{x}, \mathbf{u}) \equiv \sum_{t=1}^{T} \mathbf{x}_t^{\mathrm{T}}\mathbf{Q}\mathbf{x}_t + \mathbf{u}_t^{\mathrm{T}}\mathbf{R}\mathbf{u}_t, \tag{6.17}$$

for trajectories $\mathbf{x}_{1:T+1}$ and $\mathbf{u}_{1:T}$. The matrices $\mathbf{Q}$ and $\mathbf{R}$ are positive definite. If this system is also subject to linear constraints on the trajectories, then this is known as a *quadratic program*, a type of convex optimization problem which can be solved using standard algorithms. CVXPY [45] is a Python embedding of a research-friendly and open source convex-programming library for solving such problems. As we have discussed (Chapter 2, or Equation (6.2)), quantum control dynamics are bilinear, so the constraints imposed on

Equation (6.17) are nonlinear. Thus, the QP distinction no longer applies. However, we can make a two-step modification: 1. Linearize the bilinear quantum control dynamics about nominal trajectories for the state and control, 2. Solve the resulting quadratic program to determine the appropriate update direction, but avoid taking the full step because our linear dynamics may not hold true far away from the linearization. By following these two steps over successive approximations, we transform our problem into a sequential quadratic program (SQP). We solve SQP problems for quantum control in Chapter 7. In particular, pseudocode for the SQP algorithm is presented in Section 7.2.3.

6.2 Feedback in open-loop optimal control

Quantum control models are not prefect. Open-loop control schemes that are directly applied to a system with an imperfect model will result in failed outcomes. Quantum characterization is challenging [56]. Perfect characterization is not a realistic expectation for engineered systems like superconducting qubits. To make optimal control practical, effort must be made to either repair the mischaracterized model or adjust the faulty pulse sequence. This can be done by combining feedback with many iterations of the open-loop control policy.

6.2.1 Looking backward

There are many ways to measure how success an open-loop policy was when actually implemented on an experimental system (e.g., fidelity, operator norms, or randomized benchmarking [99]). Whatever measure we choose, the success or failure of our attempt is given by how accurately the model (and thereby the open-loop control) reflects the experiment. Critically, we can use feedback to attempt to improve the score by adjusting the model parameters or the values of the control sequence itself. In quantum control, this process is often referred to as a *calibration* to the experiment [196]. Because we assume the model is inaccurate, we must calibrate using a *gradient-free search*. That is, model inaccuracies prevent us from knowing how the objective function depends on the model parameters; therefore, we cannot compute its gradients. However, each score is a sample of the objective function, so we can improve our score by a direct search derived from these samples. The Nelder-Mead algorithm [148] is an

efficient example of exactly this type of gradient-free optimization, and it has been used for calibration in protocols like Ad-HOC [54] and ORBIT [96] (the difference in the two comes from how the open-loop control is scored). Broadly speaking, in calibration, we are looking backward and modifying past inputs or parameters to improve the success of our control.

6.2.2 Looking forward

An alternative way to incorporate feedback into open-loop control is model predictive control (MPC) [136, 166]. In the standard approach to MPC, a sequence of open-loop controllers are implemented *online* as a function of the current state. In the current state, a model prediction is made out to some finite prediction horizon and used to synthesize an open-loop controller. Then, the control is applied to the experiment, but–critically–this occurs for a control horizon that is much less than the prediction horizon. Often the control horizon is just a single time step. After application of the control, the resulting state is measured, and the prediction begins again from the new state. MPC is often referred to as a *receding-horizon* strategy because of this structure. The concatenated controls synthesized by MPC are robust, as MPC inherits a natural degree of disturbance rejection due to the receding-horizon feedback.

Notice that instead of a final score for the policy like in the previous Section 6.2.1, MPC relies on a receding sequence of quantum state tomography outcomes. The forward-looking and backward-looking approaches both take advantage of iterations of open-loop control experiments to improve outcomes. However, the forward-looking MPC does not rely on a gradient-free optimization; this can be advantageous because calibrating many parameters with gradient-free search is challenging. Instead, MPC uses efficient open-loop control schemes to design its control policies. To calibrate, it leverages extra online control time and its natural robustness properties to realize convergence. More about model predictive control and its application in the context of quantum optimal control can be found in Chapter 7.

Chapter 7

MODEL PREDICTIVE CONTROL FOR ROBUST QUANTUM STATE PREPARATION

This work is a presentation of Reference [73]. As such, this material is mostly self-contained.

Abstract

A critical engineering challenge in quantum technology is the accurate control of quantum dynamics. Model-based methods for optimal control have been shown to be highly effective when theory and experiment closely match. Consequently, realizing high-fidelity quantum processes with model-based control requires careful device characterization. In quantum processors based on cold atoms, the Hamiltonian can be well-characterized. For superconducting qubits operating at millikelvin temperatures, the Hamiltonian is not as well-characterized. Unaccounted for physics (i.e., mode discrepancy), coherent disturbances, and increased noise compromise traditional model-based control. This work introduces *model predictive control* (MPC) for quantum control applications. MPC is a closed-loop optimization framework that (i) inherits a natural degree of disturbance rejection by incorporating measurement feedback, (ii) utilizes finite-horizon model-based optimizations to control complex multi-input, multi-output dynamical systems under state and input constraints, and (iii) is flexible enough to develop synergistically alongside other modern control strategies. We show how MPC can be used to generate practical, optimized control sequences in representative examples of quantum state preparation. Specifically, we demonstrate for a qubit, a weakly-anharmonic qubit, and a system undergoing crosstalk, that MPC can realize successful model-based control even when the model is inadequate. These examples showcase why MPC is an important addition to the quantum engineering control suite.

7.1 Introduction

Quantum computation can be viewed as an assembly of analogue control pulses driving quantum states toward desired targets [11, 115, 138, 176, 197]. An accurate dynamical model is important for designing optimal control pulses sufficient for each application [41]. In particular, open-loop control strategies using model-based numerical methods have proven to be highly effective in some practical settings, such as for processors based on cold atoms, where models are well-known [68]. Sufficiently accurate models are obtained through careful device characterization [56]. However, descriptions of superconducting qubits [106] can fall short due to unknown Hamiltonian terms [131], instrument noise, control transfer functions [82], and/or unwanted coupling to unmodeled modes [193] or other systems [169]. To address these challenges, new frameworks must incorporate measurement feedback in addition to device characterization to optimize pulses for the desired targets. For example, an insufficient pulse from an inaccurate model can be updated retroactively using a model-free optimization derived from experiment feedback [54, 96]. In this work, we introduce the model predictive control (MPC) [136, 166] framework since it possess a number of structural advantages for closed-loop control of systems with imperfectly known dynamics: (i) MPC inherits a natural degree of disturbance rejection due to measurement feedback, (ii) receding finite-horizon model predictions allow MPC to control complex multi-input multi-output systems under state and control constraints, and (iii) MPC is flexible enough to develop synergistically alongside control innovations like reinforcement learning [74, 204]. We detail the success of MPC on a number of practical superconducting qubit applications.

MPC [136, 166] (see Figure 7.1) is a closed-loop optimization strategy that compliments existing approaches – that is, MPC is not a new type of controller on its own. Instead, standard MPC implements a sequence of open-loop controllers *online* as a function of the current state. First, a model prediction is made out to some finite horizon and used to inform the present control decision. Second, the decision is implemented, and the prediction begins again from the new measured state. For this reason, MPC is often referred to as

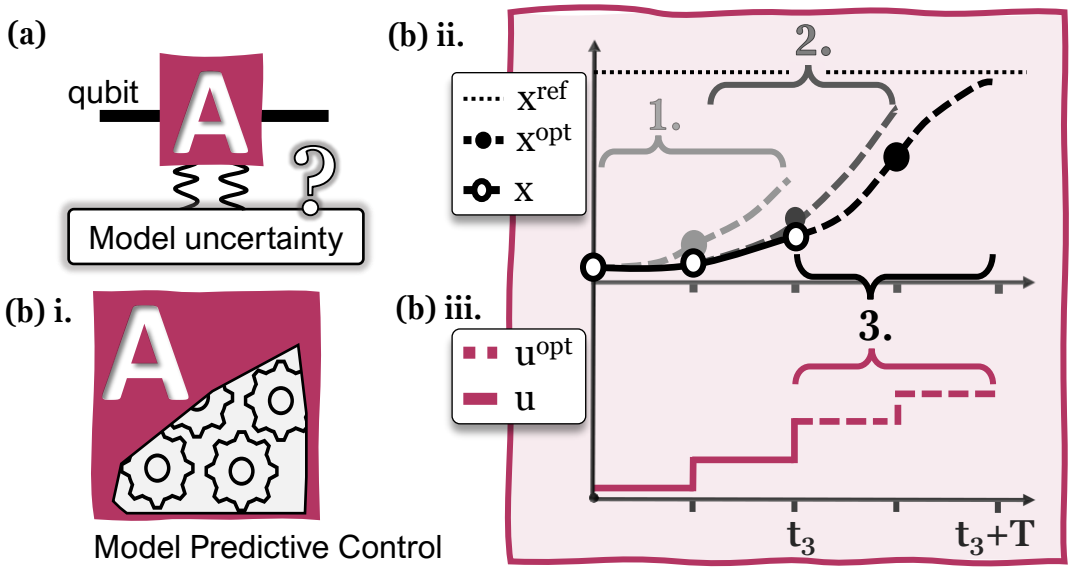


Figure 7.1: **(a)** In the circuit model a qubit is represented by a line segment, and a box is draw to indicate a quantum process applied to the qubit. Here, we represent a quantum state preparation abstractly by **A** and posit uncharacterized modifications that make **A**'s model unreliable for open-loop control. **(b)** Under the hood of the **A** operation, robust control pulses are designed using model predictive control (MPC). In **(b) ii-iii**, MPC synthesizes a robust control input by incorporating state feedback into a sequence of receding-horizon control laws (in **(b) ii.**, the first three MPC iterations are labeled 1., 2., and 3., and colored with increasing grayscale value). At each MPC iteration, an open-loop control is solved over the current prediction horizon T using **A**'s unreliable system model to yield $\mathbf{x}^{\mathrm{opt}}(t)$ and $\mathbf{u}^{\mathrm{opt}}(t)$. The first entry of the open-loop solution $\mathbf{u}^{\mathrm{opt}}(t)$ is applied as $\mathbf{u}(t)$ in **(b) iii.**, and the resulting state $\mathbf{x}(t+1)$ is recorded in **(b) ii.** The next MPC iteration begins from this recorded state.

a receding-horizon strategy. MPC has a history of practical successes ranging from the chemical process industry [165] to autonomous vehicles [61] and reusable rocket landings [59]. Notably, MPC for quantum dynamics is closely connected to MPC for classical control using Koopman-von Neumann theory, a description of classical mechanics using a Hilbert space of observables defined on phase-space. Koopman-von Neumann theory shares strong historical connections with quantum mechanics and has wide applicability to modern data-driven analysis [27, 101, 102, 141, 142, 189]. Connections between Koopman-von Neumann theory and quantum control dynamics were studied in Reference [72]. Recent successes using MPC within the classical Koopman-von Neumann theory [3, 25, 64, 103, 158, 159] can motivate synergistic development of MPC for quantum dynamics. In this work, we consider planning for control using fixed model discrepancies within MPC. In practice, model discrepancies can be improved by integrating data-driven modelling together with MPC. The integration of planning over a model and learning of a model requires making choices about how to best utilize the available data. There are many trade-offs to consider such as accuracy, interpretability, and data efficiency: for a review, see Reference [144]. One approach to the integration of planning and learning is data-driven model synthesis based on Koopman theory, using streaming data [67, 160, 202] to build or improve operator-theoretic models of the dynamics in an *online* way as the MPC horizon recedes. In the language of Reference [144], this approach is known as *planning over a learned model*, and is exemplified by approaches like Embed to Control (E2C) [192]. Error bounds for Koopman models of control systems were recently studied in [152, 170], enabling the rigorous analysis of data-driven MPC schemes.

There is a distinction to make between the classical and quantum settings. When MPC is used in robotics, the implementation of a control decision results in the actual motion of the robot to a new position. Unlike the robot, a quantum state is a statistical outcome from the measurement of many identically-prepared quantum experiments. After running successive or parallel experiments to realize quantum state tomography [56], the next quantum experiment is reset at the initial (or perhaps projected) state. There is a possibility of using the state information to look backward and improve upon the previous model or pulse. Backward-

looking, model-free algorithms iteratively search for parameter changes that improve upon desired control objectives. Such methods have been highly successful in calibrating controls to quantum experiments [54, 96, 196]. However, many iterations may be necessary to successfully search the parameter space; hence, many evaluations of the control objective may be required. Moreover, free parameters are often limited in number to maintain the feasibility of the optimization. The alternative MPC perspective is to solve the quantum problem as if it is an *online* optimization. The forward-looking MPC trades iterative improvements for extra online control time by accepting and proceeding from the current quantum state tomography outcome (we return to this comparison in detail in Section 7.3.1.1 after establishing an explicit context via our numerical examples). The resulting MPC control scheme is robust, as MPC inherits a natural degree of disturbance rejection due to the feedback. Moreover, MPC can handle the many parameters of multi-input, multi-output systems. Finally, MPC has an expansive literature base from which to build. In this paper, we demonstrate the utility of the MPC perspective in the context of quantum control engineering. First, we introduce our MPC implementation. Then, we proceed with examples of robust quantum state preparation for ideal qubits, weakly-anharmonic qubits, and qubits coupled by undesired crosstalk.

7.2 MPC for quantum control

In what follows, Section 7.2.1 describes the theoretical framework for standard MPC, restricted for simplicity to linear models with quadratic cost functions. Section 7.2.2 describes the nonlinear equations of motion for quantum control dynamics and Section 7.2.3 shows how to modify standard MPC for the nonlinear quantum control dynamics by way of nonlinear MPC.

7.2.1 Background: MPC

The standard linear-quadratic MPC involves a discrete-time linear model, a quadratic cost function, and possibly linear constraints on the state and control. The optimization problem generated under these assumptions is a quadratic program (QP). With an initial state $\mathbf{x}_0$ and reference trajectories $\mathbf{X}^{\text{ref}} = \left[\mathbf{x}^{\text{ref}}(0), \mathbf{x}^{\text{ref}}(1), \ldots, \mathbf{x}^{\text{ref}}(T)\right]$ and $\mathbf{U}^{\text{ref}} =$

$\left[\mathbf{u}^{\mathrm{ref}}(0), \mathbf{u}^{\mathrm{ref}}(1), \ldots, \mathbf{u}^{\mathrm{ref}}(T-1)\right]$, the QP is given by

$$\mathrm{QP}_{\mathrm{MPC}}(\mathbf{x}_0, \mathbf{X}^{\mathrm{ref}}, \mathbf{U}^{\mathrm{ref}}) = \tag{7.1a}$$

$$\arg\min_{\mathbf{X},\mathbf{U}} \sum_{t=0}^{T-1} \left(\left\|\mathbf{x}(t) - \mathbf{x}^{\mathrm{ref}}(t)\right\|_{\mathbf{Q}}^2 + \left\|\mathbf{u}(t) - \mathbf{u}^{\mathrm{ref}}(t)\right\|_{\mathbf{R}}^2 \right) + \left\|\mathbf{x}(T) - \mathbf{x}^{\mathrm{ref}}(T)\right\|_{\mathbf{Q}_f}^2 \tag{7.1b}$$

$$\text{s.t.} \quad \mathbf{x}(t+1) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad t = 0, 1, \ldots, T-1 \tag{7.1c}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{7.1d}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}(t) \leq \mathbf{x}_{\max}, \quad t = 1, 2, \ldots, T \tag{7.1e}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max}, \quad t = 0, 1, \ldots, T-1 \tag{7.1f}$$

where $\|\mathbf{x}\|_{\mathbf{Q}}^2 := \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}$. The quadratic costs $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{Q}_f$, dynamics $\mathbf{A}(t)$, $\mathbf{B}(t)$, and prediction horizon T are parameters. The QP returns an optimal state and control, $\mathbf{X}^{\mathrm{opt}}$ and $\mathbf{U}^{\mathrm{opt}}$, for the entire prediction horizon. If no state or input constraints are introduced, then this problem could be solved using familiar linear quadratic methods [6]. Alternatively, more general constraints may be considered if the resulting optimization can be reformulated as a second-order cone program (SOCP), of which a QP is a special case [21, 87]. Typical MPC applications involve real-time control of complex systems. This motivates work on faster and more efficient MPC algorithms [191]. Many powerful open-source and commercial MPC solvers exist today for a variety of purposes. In our work, we used CVXPY [45] (a research-friendly Python-embedded convex-programming library) together with the QP solver OSQP [178].

7.2.2 Quantum control dynamics

Markovian quantum control dynamics [5] are bilinear with respect to the state vector $|\psi(t)\rangle \in \mathbb{C}^N$ and coherent control amplitudes $u_j(t)$ such that

$$\frac{\partial}{\partial t}|\psi(t)\rangle = -i(H_0 + \sum_j u_j(t)H_j)\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle\,. \tag{7.2}$$

For a qubit, $N = 2$, while for a qudit, $N = d$. A sequence of n qudits forms a quantum register which can be used for quantum computations; in this case, the state grows exponentially with respect to the length of the register, $N = d^n$. An ensemble of pure quantum states or registers can be completely characterized, in the sense of their measurement statistics, by a density matrix $\rho(t)$; that is, a non-negative self-adjoint operator in $\mathbb{C}^{N\times N}$ with trace one. The density matrix enables a simple characterization of Markovian interactions by the environment on the state in terms of dissipative dynamical operators; this is the Markovian master equation description of an open quantum system,

$$\frac{\partial}{\partial t}\rho(t) = -i[H(t), \rho(t)] + \frac{1}{2}\sum_{j,k=1}^{N^2-1} c_{jk}\left([D_j, \rho(t)D_k^\dagger] + [D_j\rho(t), D_k^\dagger]\right), \tag{7.3}$$

where $H(t)$ is a trace-zero Hermitian operator (corresponding with the system Hamiltonian), $\{D_j\}_{j=1}^{N^2-1}$ is an orthonormal set of complex matrices with trace zero, and $C := (c_{jk})$ is positive semi-definite. For a closed quantum system $C = 0$, the equation simplifies to the quantum Liouville equation. In this paper, we use the density matrix to represent the quantum state for the purpose of describing control experiments for quantum state preparation. In order to continue to represent our density matrix as a state vector in line with our MPC intentions, we apply the vectorization operation (see Section 7.2.2.1)

$$\begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix} \overset{\text{vec}}{\mapsto} \begin{pmatrix} \rho_{00} \\ \rho_{01} \\ \rho_{10} \\ \rho_{11} \end{pmatrix}. \tag{7.4}$$

In addition, where it is necessary to work with completely real state vectors we rely on the isomorphism between complex numbers and 2×2 matrices for the operators,

$$a + bi \overset{\text{real}}{\mapsto} \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \tag{7.5}$$

To apply MPC for quantum control dynamics, the constraint in Equation (7.1c) is replaced by the discretization of the bilinear dynamics in Equation (7.3) [72]. To first order it is given by

$$\mathbf{x}(t+1) = (\mathbf{A} + \sum_j u_j(t)\mathbf{N}_j)\mathbf{x}(t). \tag{7.6}$$

Our intended task in this paper is quantum state preparation, in which a quantum state represented by ρ is driven to realize a reference state ρ^{ref}. The success of quantum state preparation is typically scored using the (squared) fidelity $F(\rho, \rho^{\text{ref}}) = \text{Tr}\{\sqrt{\sqrt{\rho}\rho^{\text{ref}}\sqrt{\rho}}\}^2$ [99]. In the common case, where ρ^{ref} is a pure state, $F(\rho, \rho^{\text{ref}}) = \text{Tr}\{\rho\rho^{\text{ref}}\}$. Fidelity approximates the distance measures of quantum states given by the Schatten p-norms, $\left\|\rho - \rho^{\text{ref}}\right\|_p = \left\|\sigma(\rho - \rho^{\text{ref}})\right\|_p = (\sum_j \sigma_j^p)^{1/p}$ where $\sigma(\cdot)$ returns the vector of singular values σ_j of a given matrix. In Equation (7.1b), we have formulated the MPC objective in the standard way as a least-squares cost, which penalizes deviations from a reference trajectory. In the case of the vectorized density matrix, this choice is consistent with overlap fidelity in the following way: if $\mathbf{x}$ is the corresponding vectorization of the density matrices, their norms are connected under the identity $\|\rho\|_2 = \|\mathbf{x}\|_2$ (Section 7.2.2.2). Quantum state preparation can be understood as a subroutine to design the control pulses necessary to realize a reference quantum process. In Section 7.2.2.3, we discuss how MPC for quantum state preparation can be naturally extended to the pursuit of quantum gate synthesis, where the goal is to realize a finite set of unitary processes which can be combined to yield arbitrary quantum computations.

7.2.2.1 Vectorization

The vectorization of a matrix is a linear transformation that converts a matrix into a column vector, vec : $R^{m\times n} \to R^{mn}$. Vectorization is defined to be column-major or row-major, depending on the stacking priority imposed. For example, Python uses row-major vectorization in NumPy's ndarray.flatten: each row of an input array is sequentially concatenated to form one long row. In Dirac notation, vectorization is the isomorphism for when a bra is flipped to a ket, $|j\rangle\langle k| \cong |j\rangle \otimes |k\rangle$. We restate a well-known property of vectorization (c.f. Lemma 4.3.2 in [86]) here:

Lemma 7.2.1. For any linear transformation, $L : R^{m\times n} \to R^{p\times q}$, there exists a unique matrix $K(L) \in R^{mn\times pq}$ such that $\text{vec}(L(B)) = K(L)\text{vec}(B)$ for any $B \in R^{m\times n}$.

In particular, for the case of left and right matrix multiplication under row-major vectorization we have

$$\text{vec}(ABC) = \left(A \otimes C^{\text{T}}\right) \text{vec}(B) \tag{7.7}$$

7.2.2.2 Trajectory-based optimization

In Equation (7.1b) we perform trajectory-based optimization for state preparation via a cost $\left\|\mathbf{x} - \mathbf{x}^{\text{ref}}\right\|_2$ defined in terms of the ℓ_2-norm. The state is $\mathbf{x} \equiv \text{vec}(\rho)$ as discussed in Section 7.2.2 and Section 7.2.2.1. Commensurate with this trajectory-based optimization, we noted in Section 7.2.2 that state preparation is often described by quantum information theorists in terms of the state fidelity $F(\rho, \rho^{\text{ref}}) = \text{Tr}\{\sqrt{\sqrt{\rho}\rho^{\text{ref}}\sqrt{\rho}}\}^2$. Fidelity approximates (by the Fuchs-van de Graaf inequalities) the distance measures of quantum states given by the Schatten p-norms, $\left\|\rho - \rho^{\text{ref}}\right\|_p = \left\|\sigma(\rho - \rho^{\text{ref}})\right\|_p = (\sum_j \sigma_j^p)^{1/p}$ where $\sigma(\cdot)$ returns the vector of singular values σ_j of a given matrix [99]. If both ρ and ρ^{ref} are pure states, then these two perspectives are equivalent, and

$$\frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^{\text{ref}}\right\|_2^2 = \frac{1}{2}\left\|\rho - \rho^{\text{ref}}\right\|_2^2 = 1 - F(\rho, \rho^{\text{ref}}). \tag{7.8}$$

7.2.2.3 Gate synthesis

In this paper, we have focused on quantum state preparation to illustrate the application of MPC for quantum dynamics. In the case of quantum control for the purposes of quantum computation, the more desirable comparison is to quantum gate synthesis. It is known that quantum state preparation can be thought of as a subroutine for gate synthesis. Indeed, gate synthesis can also be defined as the realization of a reference state under the action of control, as we have done for quantum state preparation. In gate synthesis, the reference state is no longer a density matrix, but is instead a unitary quantum process. For a closed quantum system, a unitary quantum process is equivalent to time evolution, $\rho(t) = U(t)\rho(0)U^\dagger(t)$. From

this, observe that the reference state which is prepared by a fixed gate depends on the initial state $\rho(0)$. This means that the synthesized control pulses for a gate must simultaneously realize the state-dependent reference process for all initial states. It has been shown that for gate synthesis, it is sufficient to consider the simultaneous state preparation of just three specific initial states [70]. Therefore, to realize gate synthesis, we can directly apply our MPC algorithm to these concatenated states under dynamics given by the appropriate direct sum of the individual models. Alternatively, consider the analogy that connects the density matrix associated with a pure state to the quantum process associated with a unitary matrix. First, vectorize the unitary matrix (assume row-major vectorization), $\text{vec}(U) \equiv |U\rangle\rangle$. Now the equivalence

$$\rho \equiv |\psi\rangle \langle\psi| \leftrightarrow P \equiv |U\rangle\rangle\langle\langle U| \tag{7.9}$$

$$\dot{\rho} = -i[H, \rho] \leftrightarrow \dot{P} = -i[H \otimes \mathbb{1}, P] \tag{7.10}$$

can be used to directly map our previous work to the new problem. For example, the reference is now a process matrix, P^{ref}. Note that there are different fidelity measures to score the success of quantum gate synthesis, so the ℓ_2-norm used in the trajectory-based optimization must be interpreted accordingly [99].

7.2.3 Nonlinear MPC

Continuous-time quantum control dynamics are bilinear. By introducing a general nonlinear dynamics constraint $\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ in place of the linear dynamics found in Equation (7.1c), we enter the realm of nonlinear MPC. Here, the necessary optimization problems are typically non-convex. Nonlinear MPC permits the use of more complex models, so applications are abundant. In recent years, practical realizations have improved alongside advances in numerical methods and non-convex optimization [166]. In this paper, we treat the nonlinear MPC by following the sequential quadratic program (SQP) approach to solve the open-loop optimization between MPC steps. The SQP approach is outlined in

Algorithm 3 Sequential Quadratic Programming

INPUT: Initial state $\mathbf{x}_0$, initial guesses $(\mathbf{X}^{\text{guess}}, \mathbf{U}^{\text{guess}})$, and references $(\mathbf{X}^{\text{ref}}, \mathbf{U}^{\text{ref}})$.
OUTPUT: Optimal trajectories $(\mathbf{X}^{\text{opt}}, \mathbf{U}^{\text{opt}})$.

function SQP($\mathbf{x}_0$, $\mathbf{X}^{\text{ref}}$, $\mathbf{U}^{\text{ref}}$, $\mathbf{X}^{\text{guess}}$, $\mathbf{U}^{\text{guess}}$)
  while Not converged **do**
    Compute the linearizations $\mathbf{A}^{\text{guess}}(t)$, $\mathbf{B}^{\text{guess}}(t)$ ▷ Eq. (7.11)
    $(\mathbf{X}^{\text{opt}}, \mathbf{U}^{\text{opt}}) \leftarrow \text{QP}_{\text{NMPC}}(\mathbf{x}_0, \mathbf{X}^{\text{ref}}, \mathbf{U}^{\text{ref}}, \mathbf{X}^{\text{guess}}, \mathbf{U}^{\text{guess}})$ ▷ Eq. (7.12)
    Compute $\alpha \in [0, 1]$ using line search
    $(\mathbf{X}^{\text{guess}}, \mathbf{U}^{\text{guess}}) \leftarrow (\mathbf{X}^{\text{guess}}, \mathbf{U}^{\text{guess}}) + \alpha(\mathbf{X}^{\text{opt}} - \mathbf{X}^{\text{guess}}, \mathbf{U}^{\text{opt}} - \mathbf{U}^{\text{guess}})$
  end while
  $(\mathbf{X}^{\text{opt}}, \mathbf{U}^{\text{opt}}) \leftarrow (\mathbf{X}^{\text{guess}}, \mathbf{U}^{\text{guess}})$
end function

Algorithm 3 [77]. SQP is a type of direct optimization: the state and control are explicitly treated as optimization variables which are constrained by the dynamics [164]. This is in contrast to indirect approaches (in quantum control, familiar examples include GRAPE [97], GOAT [128], or Krotov's method [69]), where the dynamics are used to eliminate the optimization over the state variables. SQP iteratively solves local QP approximations of the nonlinear optimization problem until convergence is achieved. To account for the approximate character of the problem, steps are taken by following a line search [151]. Each local QP approximation is taken about guess trajectories $\mathbf{X}^{\text{guess}} = [\mathbf{x}^{\text{guess}}(0), \mathbf{x}^{\text{guess}}(1), \ldots, \mathbf{x}^{\text{guess}}(T)]$ and $\mathbf{U}^{\text{guess}} = [\mathbf{u}^{\text{guess}}(0), \mathbf{u}^{\text{guess}}(1), \ldots, \mathbf{u}^{\text{guess}}(T-1)]$). Denote this modified version of the QP as $\text{QP}_{\text{NMPC}}(\mathbf{x}_0, \mathbf{X}^{\text{ref}}, \mathbf{U}^{\text{ref}}, \mathbf{X}^{\text{guess}}, \mathbf{U}^{\text{guess}})$. To run this modified optimization, we first linearize the dynamics $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ such that

$$\begin{aligned} \mathbf{A}^{\text{guess}}(t) &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^{\text{guess}}(t),\, \mathbf{u}^{\text{guess}}(t)} \\ \mathbf{B}^{\text{guess}}(t) &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{x}^{\text{guess}}(t),\, \mathbf{u}^{\text{guess}}(t)}. \end{aligned} \tag{7.11}$$

Then, we replace Equation (7.1c) in $\text{QP}_{\text{MPC}}(\mathbf{x}_0, \mathbf{X}^{\text{ref}}, \mathbf{U}^{\text{ref}})$ with

$$\mathbf{x}(t+1) = \mathbf{x}^{\text{guess}}(t+1) + \mathbf{A}^{\text{guess}}(t)\Delta\mathbf{x}(t) + \mathbf{B}^{\text{guess}}(t)\Delta\mathbf{u}(t) + \mathbf{r}(t+1) \tag{7.12}$$

where

$$\Delta\mathbf{x}(t) := \mathbf{x}(t) - \mathbf{x}^{\text{guess}}(t)$$
$$\Delta\mathbf{u}(t) := \mathbf{u}(t) - \mathbf{u}^{\text{guess}}(t)$$
$$\mathbf{r}(t+1) := \mathbf{f}(\mathbf{x}^{\text{guess}}(t), \mathbf{u}^{\text{guess}}(t), t) - \mathbf{x}^{\text{guess}}(t+1).$$

The discretization of the continuous time quantum control dynamics can be taken before or after the linearization in Equation (7.11) [77]. We pursue the case where discretization is taken first. For quantum control dynamics, we saw in Equation (7.6) that the discretized model contains a bilinear nonlinearity in the state and control. A bilinear nonlinearity is relatively simple, and it has been shown that successful and efficient nonlinear MPC controllers can be implemented even under guesses that account only for the initial state (i.e. $\mathbf{x}^{\text{guess}}(t) = \mathbf{x}_0$ for $t = 0, 1, \ldots T-1$) [25, 37, 64]. The known bilinear structure can also be used to increase optimization efficiency [159]. Additional accuracy can be sought by using better discretizations or even data-driven numerical integrators [72, 77]. Automatic differentiation tools can also prove impactful in these cases.

SQP can be warm-started after the implementation of the first control decision. At the initial timestep $t = 0$, suppose SQP was run in full to find an optimal state and control over the prediction horizon. As a by-product, good guess trajectories $\mathbf{X}_0^{\text{guess}}$ and $\mathbf{U}_0^{\text{guess}}$ for this prediction horizon have been obtained through SQP. All future guess trajectories can then be found via a shifting procedure applied to these good guess trajectories. A common practical implementation is as follows: obtain the warm-start guess trajectories for the initial timestep t from the previous $\mathbf{X}_{t-1}^{\text{guess}}$ and $\mathbf{U}_{t-1}^{\text{guess}}$ by eliminating the first value and duplicating the final value so that each sequence retains its length. Heuristically, the shifted guesses are close to the optimal value of the nonlinear program. Therefore, for all initial timesteps $t > 0$ the SQP can be terminated after just one iteration with $\alpha = 1$ (see Algorithm 3) [77].

7.3 MPC for robust quantum state preparation

MPC can be utilized for many control strategies, such as setpoint stabilization (realizing a static reference state), tracking (following a time-dependent reference trajectory), and path following (staying close to any part of a time-independent reference trajectory at all times) [166]. Each strategy has different objectives and criteria for convergence and stability. Selecting a suitable control strategy is important for obtaining the best performance. In our examples, we consider MPC for setpoint stabilization to perform robust quantum state preparation in the presence of coherent noise and modeling inaccuracies.

Section 7.3.1 introduces MPC for robust state preparation of a qubit, and discusses the implications of the measurement feedback period. Section 7.3.2 uses MPC to control a weakly anharmonic qubit assuming no model knowledge of the anharmonicity and compares the MPC treatment with an analytic pulse design. Finally, Section 7.3.3 looks at simultaneous state preparation of coupled qubits in the presence of an unmodeled coupling (i.e. crosstalk). Working in the individual (reduced) qubit spaces, MPC is used to overcome the crosstalk effect. For each of our ground truth simulations, we use the QuTiP Python package [90, 91]. The following examples report the Hamiltonians of each toy system, with the understanding that the relevant modifications are made following Section 7.2.2 to obtain the dynamical model used to constrain the optimal control problem.

7.3.1 Qubit

Our first example is a qubit treated in a rotating frame within the rotating wave approximation. The Hamiltonian is

$$H(t) = \frac{\Delta}{2}\sigma_z + \frac{u(t)}{2}\sigma_x \tag{7.13}$$

where σ_z and σ_x are the usual Pauli matrices [106]. The prefactor $\Delta = \omega_Q - \omega_R$ is the discrepancy between the qubit resonance frequency ω_Q and the chosen rotating frame ω_R. The control $u(t)$ is the envelope of a drive pulse with a carrier frequency of ω_R. Our goal is to apply an area-π pulse or swap of the occupation probability of the ground and excited states. To do this, we set a fixed ρ^{ref} such that $\rho^{\text{ref}}_{11} = 1$ else zero. In all examples, we only

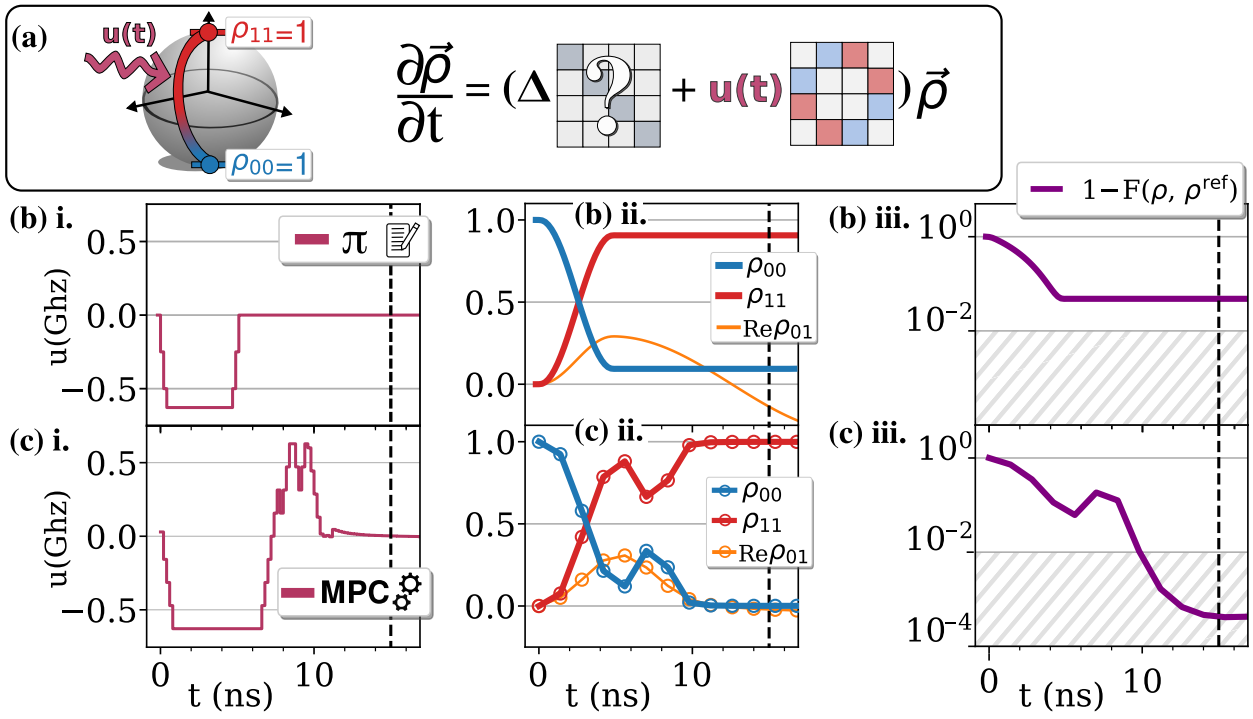


Figure 7.2: **(a)** A qubit trajectory for the desired state preparation is represented on the Bloch sphere. The goal is to transfer the ground state probability to the excited state, i.e. $\rho_{00} \rightarrow \rho_{11}$. There is a model discrepancy of $|\Delta| = 200$ MHz relative to the simulation. **(b) i.** Without a discrepancy, an area-π pulse would perform the desired state preparation. Constraints have been enforced on the control's maximum amplitude and first derivative. **(b) ii.** The realized trajectories of the density matrix components under the pulse in **(b) i.** are plotted. In **(b) iii.** the infidelity is shown to fail to reach the hatched success region due to the unreliable model. **(c) i.** The pulse designed by model predictive control (MPC) is shown. The same constraints used in **(b) i.** are imposed here. **(c) ii.** Every 1.4 ns, the density matrix was recorded (open circles) and used to initialize an open-loop optimization over the MPC prediction horizon. **(c) iii.** The operation infidelity is shown to reach the hatched success region due to the robust MPC pulse. The dotted vertical lines throughout **(b)**-**(c)** indicates the time used for the infidelity in Section 7.3.1.2 Figure 7.4.

enforce on-axis terms (e.g. ρ_{00}, ρ_{11}) in our cost function by way of the appropriate $\mathbf{Q}$. We set $\mathbf{Q}_f = \mathbf{Q}$, and $\mathbf{R} = 10^{-2}\ \mathbb{1}$.

Mischaracterization of the qubit frequency ω_Q results in an offset rotating frame and a nonzero drift term Δ in the qubit Hamiltonian (visualized in Figure 7.2(a)). Open-loop controllers rely on highly-accurate models to find good pulse designs. In contrast, we will demonstrate that MPC does not require optimal open-loop solutions over its prediction horizon to realize good pulse designs; instead, MPC iterations are best understood as planning exercises. In Figure 7.2, MPC is used to design a control pulse swapping the occupation probability of the ground and excited states. The simulated qubit is assumed to be mischaracterized. Specifically, we force MPC to rely on an inaccurate model with $\Delta = 0$ while the simulation actually has been set to $\Delta/2\pi = -0.2/2\pi \approx -30$ MHz. For a base comparison, in Figure 7.2(b) we report the effect of an analytic area-π pulse. This is the naive design that would be realized by an open-loop controller if the $\Delta = 0$ model accurately reflected the simulation; we see that the presence of the nonzero Δ in the simulation renders this analytic pulse insufficient.

We apply MPC in Figure 7.2(c). We set our prediction horizon to $T = 10$ ns. Feedback is provided via the simulated density matrix every 1.4 ns. In experiment, this feedback would necessitate the use of quantum state tomography [56]. We explore the effect of decreasing or increasing the feedback period in Section 7.3.1.2. Recall that the MPC controller naturally accommodates constraints like those emerging from hardware limitations. To demonstrate, in Figure 7.2 notice that we have set constraints on both the maximum control amplitude ($|u|/2\pi = 0.1$) and the allowed initial change in control relative to the previous value ($\Delta u/2\pi = 0.04$). We enforce the same constraints for the analytic pulse in Figure 7.2(b). In general, these hard constraints can be turned off or softened (by allowing for some mild infractions of the desired inequalities) to increase the space of feasible controls and state trajectories within MPC. Upon comparing Figure 7.2(c) with Figures 7.2(b), we see how the MPC solution uses feedback to adjust and robustly counteract the model deficiencies observed during its receding-horizon iterations. In the introduction, a conceptual

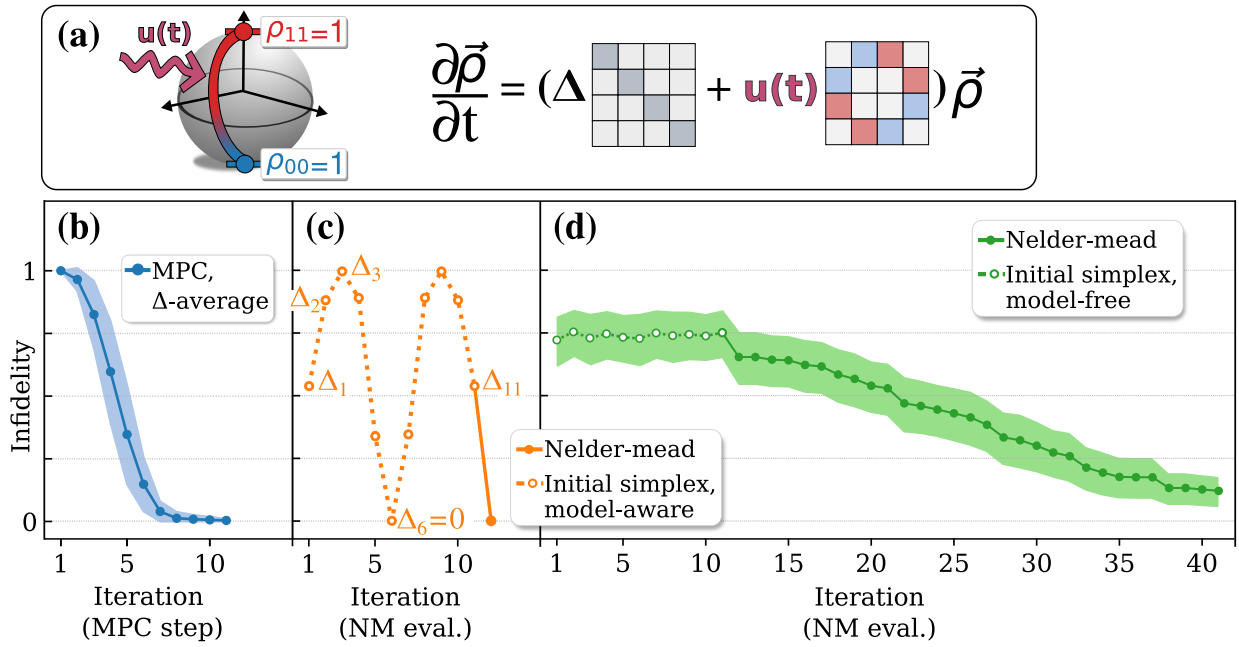


Figure 7.3: **(a)** The system from Figure 7.2 is repeated here, with 1 ns timesteps yielding 10 ns control pulses (thus, 10 free parameters). The true model is $\Delta = 0$, while a uniform sample of 11 discrepancies is considered: $\Delta \in [-0.36, 0.36]$. **(b)** MPC is applied to synthesize a robust 10 ns control pulse for each of the 11 models. **(c)** Nelder-Mead is used for descent in the 10-dimensional parameter space of the controls. The 11 coordinates for the initial Nelder-Mead simplex are the open-loop optimal controls determined from the models specified by the 11 values for Δ (thus, the simplex is model-aware). Notice the range of infidelities depending on how well the model reflects the simulation. **(d)** Nelder-Mead is applied using a random initial simplex (thus, the simplex is model-free). Each iteration is an evaluation of the final infidelity of a 10 ns control pulse.

comparison between the forward-looking MPC and backward-looking, model-free algorithms (e.g. Reference [54, 96, 196]) was offered. To further elucidate this point, Section 7.3.1.1 illustrates this comparison by way of the toy qubit system introduced here in this section.

7.3.1.1 Comparison with model-free descent

In this section, we compare MPC for pulse synthesis to model-free descent in the parameter space of the control pulse. This latter perspective is a simplification of the approach suggested in, e.g. References [54, 96, 196]. We compare the two approaches by tracking the iterations

of quantum state tomography (QST) required to implement a control pulse that satisfies the state preparation objective from Section 7.3.1 (Figure 7.2). In MPC, each QST instance is used to evaluate the current state for the purpose of determining the forward-looking open-loop control optimization. Each MPC step requires one QST iteration to assess the current state. In model-free descent, QST is used to determine the final pulse infidelity. Each QST iteration is an evaluation of this infidelity, which occurs after the application of a complete control sequence. It should be noted that one advantage of model-free descent, as advocated in [54, 96, 196], is the use of methods like randomized benchmarking in lieu of QST [56] for the purpose of evaluating a proxy for the direct infidelity.

We use the Nelder-Mead algorithm [54] to calibrate the control pulses via model free descent. The Nelder-Mead algorithm starts with a simplex of one dimension larger than the number of parameters, and chooses new search directions based on the objective values of this simplex. In this section, we slightly modify the example from Section 7.3.1: we take 1 ns timesteps and consider a total control window of 10 ns, meaning that we have 10 independent control parameters determining our desired pulse. Notice that this means a direct application of the Nelder-Mead algorithm will require an 11-point simplex. We allow for discrepancies, $\Delta \in [-0.36, 0.36]$ GHz, with the true model given by $\Delta = 0$ (for more on these choices, see Figure 7.4 in Section 7.3.1.2). In particular, we consider 11 uniformly-sampled $\Delta_1, \Delta_2, \ldots, \Delta_{11}$. In Figure 7.3 (b) we show the infidelity that results from applying MPC with horizon 5 ns from step 1 to step 10 of the pulse synthesis. The reported infidelity has been averaged over the 11 uniformly-sampled values Δ, with error given by the standard deviation. For the full range of models, MPC is able to prepare the state after 10 iterations of QST (equivalently, 10 steps of the MPC algorithm). In Figure 7.3 (c) and (d) we instead consider model-free descent for two different initial simplexes. The first, in Figure 7.3 (c), we refer to as model-aware. We take as the coordinates of our simplex the 10 ns pulses determined by solving the open-loop optimal control problem for each $\Delta_1, \Delta_2, \ldots, \Delta_{11}$. Notice that the infidelity of the open-loop control varies depending on how closely the model Δ reflects the true $\Delta = 0$ of the simulation. We call this approach model-aware because we are in effect

reducing the model-free calibration to a trial-and-error search over the discrepancy value. Again, each iteration is an evaluation of the infidelity and therefore treated as a QST instance equivalent to an MPC step: just setting up the simplex demands 11 QST iterations. In Figure 7.3 (d), we seed the construction of the initial simplex using a single random 10 ns pulse within the allowed range of control amplitudes. The remaining coordinates of the simplex are chosen by separately incrementing each component by the maximum Δu (a default approach for implementations of Nelder-Mead like in SciPy [187]). In the example in this section, we have set the control saturation at $|u|/2\pi = 0.1$ and $|\Delta u|/2\pi = 0.05$, to match the example in the main text. A total of 10 random initial simplexes are considered, providing the mean and variance of the sample mean reported in Figure 7.3 (d).

Observe that in the fully model-free application of the Nelder-Mead descent seen in Figure 7.3 (d), relatively many iterations were necessary to successfully search the parameter space. Indeed, if we extrapolate to a case where there are more than 10 free parameters, even the data required for the initial simplex becomes costly–this without even beginning the descent. We see in Figure 7.3 (b) that the alternative MPC perspective solved the quantum problem as if it was an *online* optimization. The forward-looking MPC was able to trade iterative improvements for the possibility of extra online control time by accepting and proceeding from the current quantum state tomography outcome. We see that the resulting MPC control scheme was robust across a range of modeling inaccuracies, as MPC inherits a natural degree of disturbance rejection due to the feedback. Moreover, MPC can handle the many parameters of multi-input, multi-output systems because it is only solving a forward-looking open-loop optimization at each step.

7.3.1.2 MPC limits: Feedback period and discrepancy

A control strategy based on MPC will eventually break down if the model discrepancy is too great or the feedback is too infrequent. In this appendix, we explore these limits within the context of the example in Section 7.3.1. We consider fixed model discrepancies within MPC. In practice, model discrepancies can be improved by integrating data-driven modelling (e.g.,

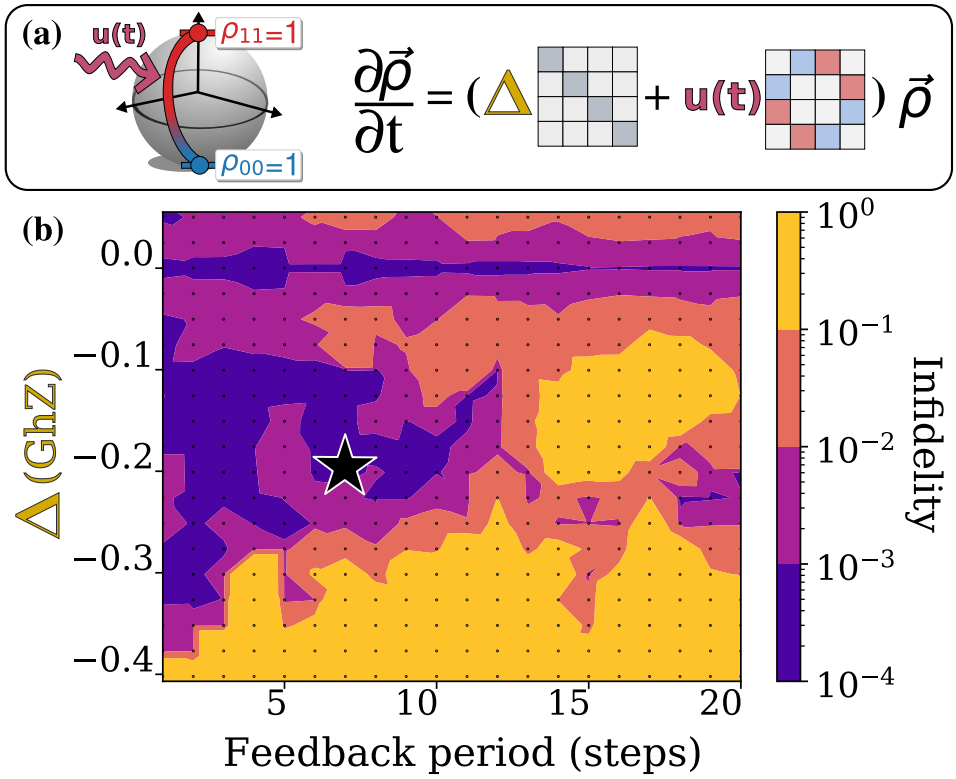


Figure 7.4: **(a)** The study from Figure 7.2 is repeated here for various Δ (discrepancies between the model and simulation). **(b)** Level curves are plotted which show qubit infidelity $F(\rho, \rho^{\text{ref}})$ after 12.5 ns of the state preparation experiment exemplified by Figure 7.2. The vertical axis, Δ, is the discrepancy between the simulation and model qubit frequencies. The horizontal axis measures the number of model predictive control (MPC) iterations undertaken before receiving the next state feedback from the qubit simulation. For any intermediate iterations, MPC was closed using state feedback from the possibly-inaccurate model predictions. For example, at $\Delta = 0$ MHz, the model coincided with the simulation so an optimal infidelity (with respect to our numerics) was obtained for all feedback periods. Each dot represents one MPC computation using full sequential quadratic programming (SQP); the coloring was performed via linear interpolation between dots. The results in Figure 7.2(c) were obtained from the parameters labelled by the star.

Reference [72]) together with MPC: indeed, the error bounds for Koopman models of control systems were recently studied in [152, 170], allowing for a more careful study of data-driven MPC schemes.

In addition to the fixed model discrepancy, the other source of error is the frequency with which the MPC controller incorporates measurement feedback. In applications of MPC for robotics, it is possible that there is a timing mismatch between feedback frequency and MPC runtime. A common challenge is the case where the robot dynamics and corresponding feedback occur faster than the MPC algorithm can return a control decision. Fortunately, the ensemble nature of the quantum experiments used to compute the quantum state tomography limits this concern. On the other hand, in robotics there is also the case where measurements occur slower than the timescales demanded by the model dynamics–a similar situation emerges when trying to limit the number of times the quantum state tomography is performed. In this latter case, the MPC algorithm can be closed using the model of the dynamics until the feedback time is reached, at which point a new quantum state tomography can be performed. Reliance on model predictions can be effective for limiting the number of measurements, but introduces additional dependence on the quality of the model.

In Figure 7.4, we looked at the infidelity of the control as a function of the discrepancy Δ and how often we rely on the model to close the MPC loop. The infidelity of the state preparation was measured after 15 ns (see Figure 7.2) and is colored by order of magnitude. The horizontal axis reports measurement feedback period in steps (the MPC timestep is 0.2 ns). This period is the number of MPC iterations for which a model prediction was used to close the loop before a sample from the qubit simulation was used. That is, a long measurement feedback period relies more heavily on a possibly-inaccurate model. Observe that at $\Delta = 0$ MHz in Figure 7.4 the model prediction coincided with the qubit simulation, so for any measurement feedback period we attain an optimal infidelity (for our numerics).

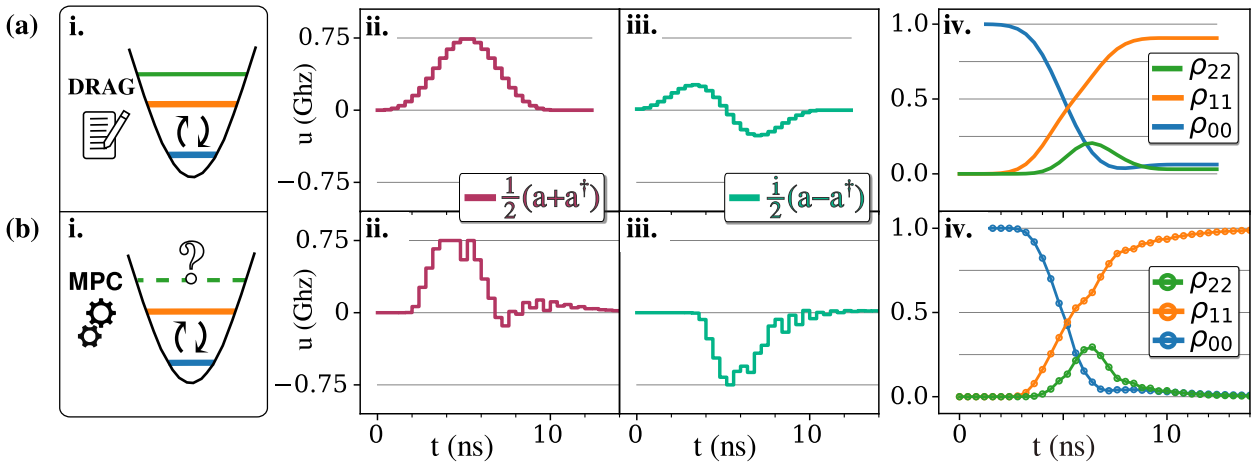


Figure 7.5: **(a) i.** In a weakly-anharmonic oscillator with unequal level spacing quantified by an anharmonicity of -100 Mhz, the intended outcome is a swap of the ground and excited state probabilities $\rho_{00} \leftrightarrow \rho_{11}$. Here ρ_{jj} is the occupation probability of the j-th state, $|j\rangle$.The DRAG procedure is an analytic suppression of leakage out of the qubit subspace defined by levels $|0\rangle$ and $|1\rangle$. It is implemented by a Gaussian π pulse on the first control axis in **(a) ii.** and a Gaussian derivative on the second control axis in **(a) iii.** in order to cancel the spectral overlap of the qubit π-pulse at the energy splitting of the $|1\rangle \leftrightarrow |2\rangle$ transition. In **(a) iv.**, the ρ_{22} population is suppressed and the ρ_{11} is enhanced by the DRAG scheme. In **(b) i.**, it is assumed that the anharmonicity is unmodeled and only a qubit model is available. In **(b) ii-iii.**, MPC is used to design a pair of robust control pulses by relying on measurement feedback to reduce the leakage from the qubit subspace. **(b) iv.** Without modeling the anharmonicity, MPC improves on the state preparation of the analytic DRAG result. Open circles are drawn every 0.4 ns to indicate the quantum state feedback. Consult Section 7.3.2.1 and Figure 7.6 for a discussion of additional cases.

7.3.2 Weakly-anharmonic qubit

A qubit is implemented by restricting a system to a pair of accessible states, for example the ground eigenstate $|0\rangle$ and the first excited eigenstate $|1\rangle$ of a bare system Hamiltonian. If there existed higher level spacing equal to the $|0\rangle \leftrightarrow |1\rangle$ transition, then driving the qubit at the $|0\rangle \leftrightarrow |1\rangle$ frequency would lead to unwanted dynamics involving higher energy eigenstates. For this reason, the energy level spacing of a candidate qubit Hamiltonian must be nonlinear. In transmon qubits, the anharmonicity is the defined to be the difference between the $|1\rangle \leftrightarrow |2\rangle$ level spacing and the $|0\rangle \leftrightarrow |1\rangle$ level spacing of the system's bare Hamiltonian. The limit of infinite absolute anharmonicity results in a perfect qubit; in practice, transmon anharmonicities cannot be made arbitrarily large and are usually between -100 to -300 MHz [106]. Naively, the transmon anharmonicity isolates the qubit transition from exchange with higher energy levels. In practice, fast changes of the drive-pulse envelope widen the spectrum of the qubit control pulse in Fourier space. The wide spectrum can lead to a nontrivial overlap with the frequency of the $|1\rangle \leftrightarrow |2\rangle$ transition and results in an undesired interaction involving the $|2\rangle$ state. The DRAG procedure (Derivative Reduction by Adiabatic Gate) is an analytic suppression of such leakage errors by coordinating control on both the σ_x and σ_y axes [106, 146]. In DRAG, if the first control envelope is fixed to a Gaussian, then the second pulse is proportional to a Gaussian derivative term that eliminates the spectral weight at the $|1\rangle \leftrightarrow |2\rangle$ transition. Numerical optimal control experiments have been shown to reproduce pulse designs similar to the DRAG scheme [146, 193].

In Figure 7.5, we consider a weakly-anharmonic transmon qubit. Our simulation Hamiltonian is defined within the $|0\rangle, |1\rangle, |2\rangle$ space. We use the rotating frame of the $|0\rangle \leftrightarrow |1\rangle$ transition and take the rotating wave approximation to realize

$$H(t) = \alpha |2\rangle\langle 2| + \frac{u_x(t)}{2}\left(a + a^\dagger\right) + \frac{iu_y(t)}{2}\left(a - a^\dagger\right) \tag{7.14}$$

where $u_x(t)$ and $u_y(t)$ are the envelopes of pulses driven at the $|0\rangle \leftrightarrow |1\rangle$ transition. In the $|0\rangle, |1\rangle, |2\rangle$ space, the transmon control operations are expressed in terms of truncated raising

and lowering operators as $\sigma_x \mapsto (a+a^\dagger)/2$ and $\sigma_y \mapsto i(a-a^\dagger)/2$. The transmon anharmonicity is $\alpha = -0.6$ or $\alpha/2\pi \approx -100$ MHz. To demonstrate the robustness of MPC, we suppose that the transmon anharmonicity is unmodeled. That is, our MPC framework is forced to use a model with $\alpha = 0$. This is similar to the situation demonstrated in Section 7.3.1, but now there are two control pulses that must operate in tandem. With only a qubit model available to our controller, we rely on MPC to design robust pulses for swapping the ground and excited state probabilities. The prediction horizon is set to 10 timesteps (4 ns) and the feedback timestep is 0.4 ns. Like in Section 7.3.1, we limit the maximum control amplitude ($|u| \leq 0.75$) and constrain the allowed initial change in control relative to its previous value ($|\Delta u| \leq 0.2$). Our constraints are chosen so that the MPC pulses are similar to the 10 ns analytic DRAG pulses shown in Figure 7.5(a). To implement our DRAG scheme, we set $u_y(t) = 0.6\dot{u}_x(t)/|\alpha|$. The constant of proportionality is a dimensionless scaling parameter, which we set to the value in $[0, 1]$ that maximizes the fidelity of the state preparation [106]. In comparing Figure 7.5(a) to Figure 7.5(b), we see that MPC is able to coordinate the two control pulses and design a trajectory for the occupation probabilities similar to that of the analytic scheme; significantly, MPC accomplishes this without knowledge of the simulation anharmonicity. In fact, we observe the MPC pulse outperforms our simple version of the analytic DRAG scheme despite this hindrance. In Section 7.3.2.1 we add additional context by successfully applying MPC to two more cases: (i) further reducing and (ii) increasing model awareness of the underlying simulation.

7.3.2.1 DRAG

Additional exploration of the weakly anharmonic qubit is pursued in this section. Figure 7.6 is an expanded version of Figure 7.5 in the main text. New examples appear as new rows, while columns i.-iv. remain the same as the corresponding columns in Figure 7.5 of the main text. An additional analytic example in Figure 7.6(a) shows the result of failing to apply the DRAG correction to the weakly anharmonic qubit. Figure 7.6(b) is the same as Figure 7.6(a) from the main text, showing the result of applying the DRAG scheme. As well, there are two

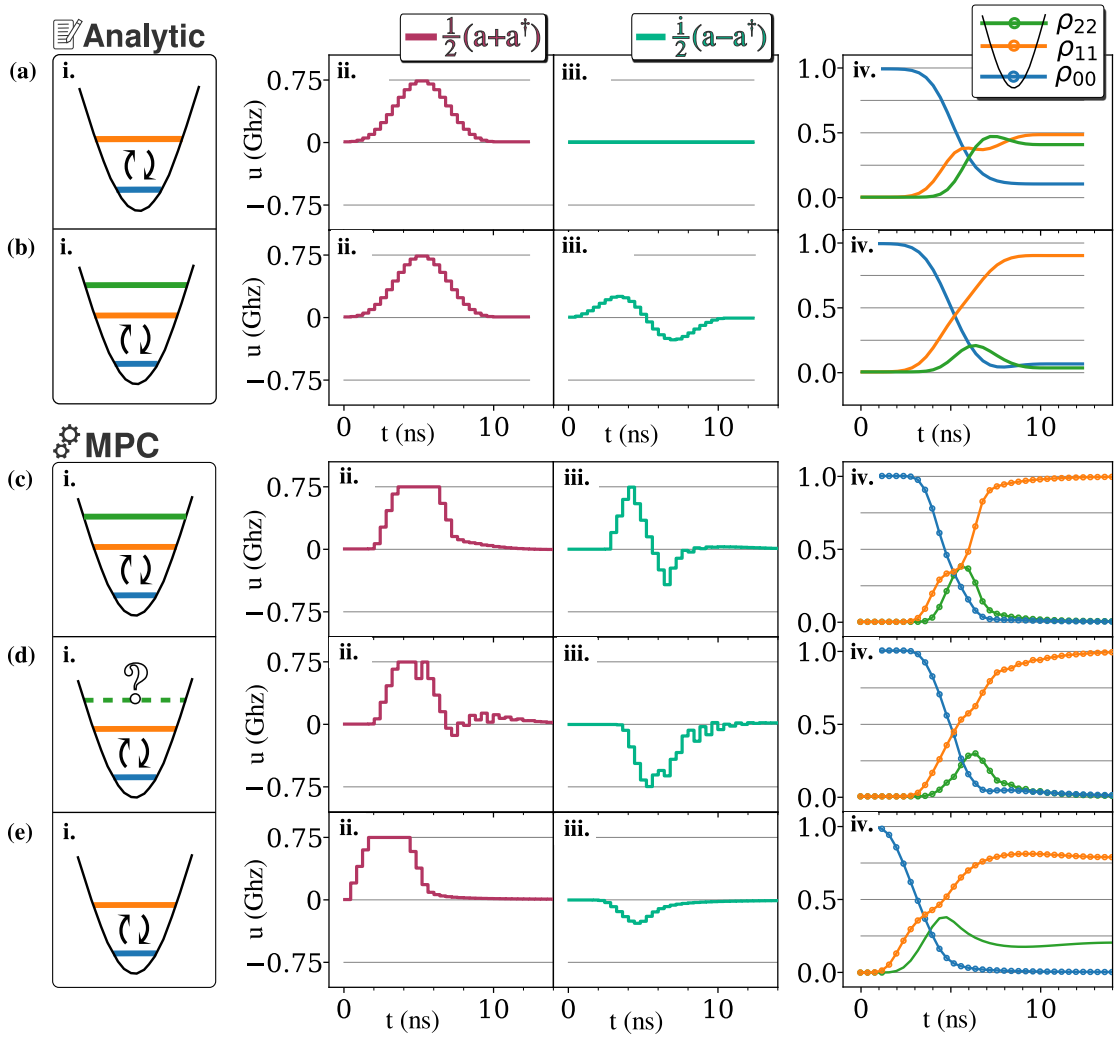


Figure 7.6: (Expanded version of Figure 7.5). In column **iv.**, recall ρ_{jj} is the occupation probability of the j-th state. In **(a)**, we show the result of failing to include the analytic DRAG correction on the second control term. **(b)** is the same as Figure 7.5(a). Parts **(c)**-**(e)** rely on MPC under differing assumptions. In the MPC section, column **iv.** has open circles on the trajectories to indicate the quantum state feedback used by MPC. In **(c)**, the model is assumed to match the simulation. Notice that in this case the control design closely resembles the analytic DRAG design in **(b)**. Part **(d)** is the same as Figure 7.5(b). In **(e)**, the model is restricted to the qubit subspace and MPC is implemented, taking only measurements in this subspace. This is illustrated in **(e) iv.** by the absence of open circles over the ρ_{22} trajectory. Even in this limited case, MPC is able to enhance the population of the ρ_{11} state significantly, in contrast to the analytic Gaussian π-pulse in **(a)**.

additional examples for MPC. The first is Figure 7.6(c) showing the result of applying MPC in the case where the model used by MPC matches the underlying simulation. A design similar to the DRAG scheme emerges from the numerical optimization. This was also observed in, e.g. References [146, 193]. Figure 7.6(d) is the same as the main text Figure 7.5(b) showing MPC in the case where the model does not include the term quantifying the anharmonicity used for the simulation. In Figure 7.6(e), we further reduce the model knowledge by forcing MPC to rely entirely on the qubit subspace. That is, unlike Figure 7.6(d), the feedback state is given only in the reduced qubit subspace and not in the full $|0\rangle, |1\rangle, |2\rangle$ space. Observe in Figure 7.6(e) that a significant enhancement of the ρ_{11} population is still achieved. In such a case as this, where the qubit model is presumed to be the best estimate of the simulation, the resulting analytic pulse is something like the Gaussian π pulse shown in Figure 7.6(a). By contrasting Figure 7.6(a) with Figure 7.6(e), we see that MPC provides a robust framework for control design in the presence of model uncertainty.

7.3.3 Crosstalk

When calibrating gates and algorithms for quantum processors at scale, additional terms in the subspace dynamics may appear due to unintended crosstalk between otherwise independent parts of the full quantum system [169]. We suggest simultaneously applying MPC to the parts that would, without crosstalk, have known models and independent control objectives. For this purpose, we recall the reduced density matrix $\mathrm{Tr}_E\{\rho_{SE}\} = \rho_S$ for a quantum state $\rho \in \mathcal{H}_S \otimes \mathcal{H}_E$ in the joint Hilbert space of a system S and environment E. Here, Tr_E is the partial trace over $\mathcal{H}_E$. In the following discussion, the feedback state supplied to MPC is the concatenation of the reduced density matrices for each of the otherwise independent parts of the full quantum system (i.e. for two parts A and B of a joint system, the feedback state is the concatenation of $\rho_A = \mathrm{Tr}_B\{\rho_{AB}\}$ and $\rho_B = \mathrm{Tr}_A\{\rho_{AB}\}$ as will be described below). As the MPC control horizon recedes, control decisions are made using just the information in these parts to correct for the unintended effects of the crosstalk. At a given time step in the control design, we assume that any observed crosstalk effects were coherent and fixed by the past

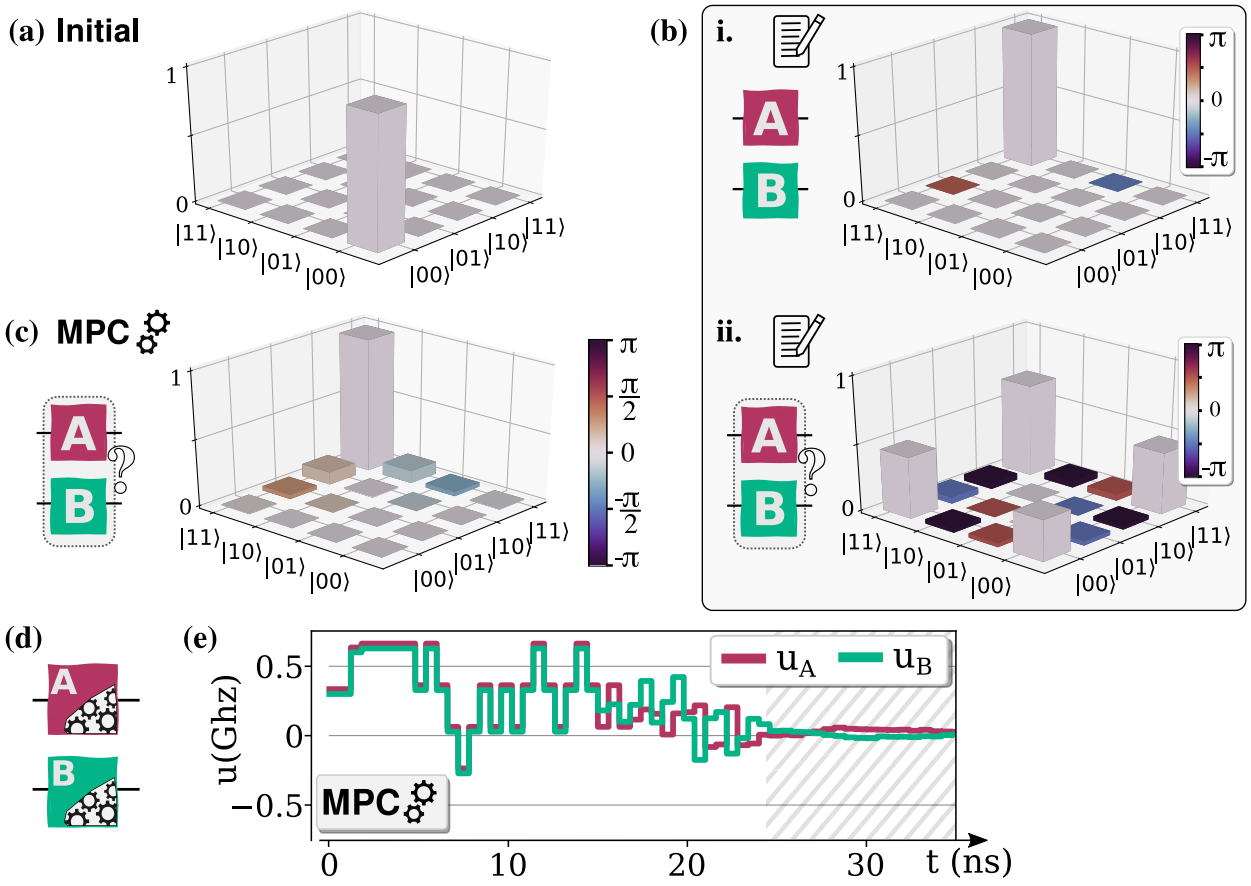


Figure 7.7: A simulation of two qubit state preparations denoted by **A** (Hamiltonian: $H_A(t) = u_A(t)\sigma_x^A/2$) and **B** ($H_B(t) = u_B(t)\sigma_y^B/2$) is modified by a strong crosstalk term proportional to $\sigma_z^A \otimes \sigma_z^B$. **(a)** The initial density matrix is in the ground state $\rho(0) = |00\rangle\langle 00|$ where $|00\rangle \equiv |0\rangle_A \otimes |0\rangle_B$. The height of the bar indicates the magnitude of that entry in the density matrix, and the color is the phase. The goal is to send the ground state to the excited state for both qubits. This means the target or reference state is set to $\rho^{\text{ref}} = |11\rangle\langle 11|$. **(b)** Suppose the crosstalk term is not present so u_A, u_B are analytically two π-pulses realizing the independent qubit state preparations. **(b) i.** In a simulation without crosstalk, the density matrix of the coupled system (shown at 25 ns) is successfully prepared in the reference state. **(b) ii.** In a simulation with crosstalk, the same pulses fail to correctly prepare the density matrix of the coupled system (shown at 25 ns). **(c)** We force model predictive control (MPC) to use a model with zero crosstalk to design the controls for a simulation where crosstalk is present. The state of the coupled system (shown at 25 ns) is successfully prepared by MPC. **(d)** The MPC framework is able to robustly prepare the state using only feedback from the concatenation of the reduced density matrices ρ_A, ρ_B in order to overcome the crosstalk modeling discrepancy. **(e)** The settling time of the successful MPC control is > 20 ns (shaded region); without crosstalk, the settling time was < 10 ns.

control decisions. That is, repeat experiments experience the same crosstalk. Corrections based on MPC are made by way of the future evolution. This feature is important because any changes made to past controls modify the form of the past crosstalk in new and uncertain ways.

In Figure 7.7, we use MPC to implement state preparation for two qubits coupled by unintended crosstalk. The qubit spaces are indexed by A and B, and the desired state preparations are denoted redundantly as $\mathbf{A}$ and $\mathbf{B}$. The total system Hamiltonian (each qubit in the rotating frame after the rotating wave approximation) is

$$\begin{aligned} H(t) &= H_{AB} + H_A(t) \otimes \mathbb{1}_B + \mathbb{1}_A \otimes H_B(t) \\ &= \frac{\xi}{2} \underbrace{\sigma_z^A \otimes \sigma_z^B}_{\text{crosstalk}} + \frac{u_A(t)}{2}\, \sigma_x^A \otimes \mathbb{1}_B + \frac{u_B(t)}{2}\, \mathbb{1}_A \otimes \sigma_y^B \end{aligned} \tag{7.15}$$

with $u_A(t)$ and $u_B(t)$ the control envelopes of the on-resonance carrier pulses. Here, we choose $\xi = 0.5$ ($\xi/2\pi \approx 80$ MHz). We use separate axes to implement the $\mathbf{A}$ and $\mathbf{B}$ operations: respectively, $H_A(t) = u_A(t)\sigma_x^A/2$ and $H_B(t) = u_B(t)\sigma_y^B/2$. This allows the trajectories of the reduced states to be somewhat distinct, while simultaneously retaining for each operation the intuition developed during Section 7.3.1. We assume that the model used by MPC is unaware of the crosstalk in the simulation Hamiltonian, and that the feedback state for MPC is the concatenation of the reduced density matrices ρ_A and ρ_B with otherwise independent objectives $\mathbf{A}$ and $\mathbf{B}$. The MPC prediction horizon is set to 10 timesteps (6 ns) with a feedback timestep of 0.6 ns and constraints are added identical to Section 7.3.1. In Figure 7.7(a), we show the initial density matrix which is in the ground state of the joint system, $\rho_{AB}(0) = |00\rangle \equiv |0\rangle_A \otimes |0\rangle_B$. Expressed in the joint system, the target or reference state is $\rho^{\text{ref}} = |11\rangle \equiv |1\rangle_A \otimes |1\rangle_B$. In Figures 7.7(b)-(c) we report the result of different state preparation experiments after 25 ns. For context, recall that the single qubit state preparations performed in Sections 7.3.1-7.3.2 were realized in < 10 ns. In Figure 7.7(b), we apply MPC to design a pair of control pulses for the case of a simulation where no crosstalk

is present. In this situation, the model used by MPC matches the simulation. The MPC solution is comparable to two analytic π pulses applied to both qubits. This is because the two systems were uncoupled, so the feedback state defined by the concatenation of the reduced density matrices ρ_A and ρ_B provides complete information about the independent system dynamics. Indeed, we see in Figure 7.7(b) i. that this pair of π-pulse proxies prepares the appropriate joint reference state. For comparison, Figure 7.7(b) ii. shows the result of using the same pulses to control a simulation where a crosstalk term (Equation (7.15)) couples the two systems. Observe that this case (which corresponds to using an analytic or open-loop solution based on an incorrect model) results in a failed state preparation.

Contrast Figure 7.7(b) with Figure 7.7(c)–in the latter, we apply MPC to realize our reference state in a simulation with crosstalk using a model that has no crosstalk term. The control pulses designed by MPC are able to use simulation feedback to overcome the model discrepancy from the unknown crosstalk. Figure 7.7(c) shows that the joint density matrix ρ_{AB} of the coupled system after 25 ns is successfully prepared in the reference state. Figure 7.7(d) is a cartoon to emphasize that the control pulses are designed using a feedback state defined by the concatenation of just the reduced density matrices ρ_A and ρ_B. In Figure 7.7(e) we see the control pulses designed by MPC. Observe that the forward-looking MPC addresses the unknown crosstalk by using feedback to compensate for unplanned dynamics. This results in longer settling times > 20 ns when compared with the single qubit state preparations of the uncoupled system which were realized in < 10 ns.

7.4 Conclusion

Model predictive control (MPC) allows for robust quantum state preparation in the presence of model uncertainty. MPC gains a natural degree of disturbance rejection by relying on feedback to update its receding horizon control plan. In Sections 7.3.1-7.3.3 we demonstrated this using prototypical superconducting qubit control simulations. In our examples, we showed MPC is able to design successful control pulses when forced to rely on a model that is mismatched from the underlying simulation. Our examples focused on setpoint stabilization (realizing a static reference state), but MPC can be utilized for other control strategies like

tracking (following a time-dependent reference trajectory) and path following (staying close to any part of a time-independent reference trajectory at all times) [166].

MPC is able to integrate state-of-the-art trajectory-based optimization approaches within its receding horizon architecture to improve outcomes and computational tractability at scale. Well-designed optimization constraints can increase robustness and can be used to seek time-optimal controls [164]. Tube MPC can be used in the presence of bounded external disturbances to keep the actual state within an invariant *tube* around the nominal trajectory [62, 122, 137]. A state observer included in trajectory-based optimization can help limit the number of measurements needed to estimate the quantum state [113]. For many-qubit systems, trajectory-based controllers have been developed to assist with the exponential increase of the quantum state space [2, 78, 174]. These ideas will be incorporated in future work.

MPC is also simple and flexible enough to complement other existing control strategies. It is often useful to rely on data-driven models in MPC [14, 18, 93]; for quantum dynamics, one direction is through bilinear dynamic mode decomposition [72]. MPC and reinforcement learning (RL) have different benefits and costs, and the two can benefit from each other [74]. MPC can provide the *expert* demonstration needed to initialize the learning process in RL approaches [161]. RL approaches are successful at improving quantum optimal control [13, 150] but there are significant data costs when training this global model-free approach. The MPC architecture introduced here is robust, interpretable, and requires less data than RL-based strategies. Furthermore, model-based strategies like MPC provide greater potential for extrapolation and generalization during control planning.

Code Availability

Code supporting the findings of this study is openly available at the following URL:

https://github.com/andgoldschmidt/MPC4quantum.

BIBLIOGRAPHY

[1] Karim M. Abadir and Jan R. Magnus. Patterned matrices: commutation and duplication matrix. In *Matrix Algebra*, pages 299–320. Cambridge University Press, August 2005.

[2] Mohamed Abdelhafez, David I Schuster, and Jens Koch. Gradient-based optimal control of open quantum systems using quantum trajectories and automatic differentiation. *Physical Review A*, 99(5):052327, 2019.

[3] Ian Abraham, Gerardo de la Torre, and Todd Murphey. Model-based control using Koopman operators. In *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, jul 2017.

[4] Yuri Alexeev, Dave Bacon, Kenneth R. Brown, Robert Calderbank, Lincoln D. Carr, Frederic T. Chong, Brian DeMarco, Dirk Englund, Edward Farhi, Bill Fefferman, Alexey V. Gorshkov, Andrew Houck, Jungsang Kim, Shelby Kimmel, Michael Lange, Seth Lloyd, Mikhail D. Lukin, Dmitri Maslov, Peter Maunz, Christopher Monroe, John Preskill, Martin Roetteler, Martin Savage, and Jeff Thompson. Quantum computer systems for scientific discovery, 2019.

[5] Claudio Altafini and Francesco Ticozzi. Modeling and control of quantum systems: An introduction. *IEEE Transactions on Automatic Control*, 57(8):1898–1917, 2012.

[6] Brian DO Anderson and John B Moore. *Optimal control: Linear quadratic methods*. Courier Corporation, 2007.

[7] Philip W Anderson. More is different. *Science*, 177(4047):393–396, 1972.

[8] Travis Askham and J. Nathan Kutz. Variable projection methods for an optimized

dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, January 2018.

[9] Peter J Baddoo, Benjamin Herrmann, Beverley J McKeon, J Nathan Kutz, and Steven L Brunton. Physics-informed dynamic mode decomposition (piDMD). *arXiv preprint arXiv:2112.04307*, 2021.

[10] Zhe Bai, Eurika Kaiser, Joshua L. Proctor, J. Nathan Kutz, and Steven L. Brunton. Dynamic mode decomposition for compressive system identification. *AIAA Journal*, 58(2):561–574, February 2020.

[11] Harrison Ball, Michael Biercuk, Andre Carvalho, Jiayin Chen, Michael Robert Hush, Leonardo A De Castro, Li Li, Per J Liebermann, Harry Slatyer, Claire Edmunds, et al. Software tools for quantum control: Improving quantum computer performance through noise and error suppression. *Quantum Science and Technology*, 2021.

[12] Leonardo Banchi, Daniel Burgarth, and Michael J Kastoryano. Driven quantum dynamics: will it blend? *Physical Review X*, 7(4):041015, 2017.

[13] Yuval Baum, Mirko Amico, Sean Howell, Michael Hush, Maggie Liuzzi, Pranav Mundada, Thomas Merkh, Andre R.R. Carvalho, and Michael J. Biercuk. Experimental deep reinforcement learning for error-robust gate-set design on a superconducting quantum computer. *PRX Quantum*, 2(4), nov 2021.

[14] Thomas Baumeister, Steven L Brunton, and J Nathan Kutz. Deep learning and model predictive control for self-tuning mode-locked lasers. *JOSA B*, 35(3):617–626, 2018.

[15] Richard Bellman. Dynamic Programming. *Science*, 153(3731):34–37, 1966.

[16] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

[17] Jacob Biamonte. Lectures on quantum tensor networks, 2019.

[18] Katharina Bieker, Sebastian Peitz, Steven L Brunton, J Nathan Kutz, and Michael Dellnitz. Deep model predictive flow control with limited sensor data and online learning. *Theoretical and Computational Fluid Dynamics*, pages 1–15, 2020.

[19] S. Blanes, F. Casas, J.A. Oteo, and J. Ros. The magnus expansion and some of its applications. *Physics Reports*, 470(5-6):151–238, January 2009.

[20] Robin Blume-Kohout, John King Gamble, Erik Nielsen, Kenneth Rudinger, Jonathan Mizrahi, Kevin Fortier, and Peter Maunz. Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography. *Nature Communications*, 8(1), February 2017.

[21] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[22] Heinz-Peter Breuer and Francesco Petruccione. *The theory of open quantum systems*. Clarendon Press, 2010.

[23] Constantin Brif, Raj Chakrabarti, and Herschel Rabitz. Control of quantum phenomena: past, present and future. *New Journal of Physics*, 12(7):075008, July 2010.

[24] Andreas Brinkmann. Introduction to average hamiltonian theory. i. basics. *Concepts in Magnetic Resonance Part A*, 45A(6), November 2016.

[25] Daniel Bruder, Xun Fu, and Ram Vasudevan. Advantages of bilinear Koopman realizations for the modeling and control of systems with unknown dynamics. *IEEE Robotics and Automation Letters*, 6(3):4369–4376, 2021.

[26] Steven L. Brunton, , Joshua L. Proctor, Jonathan H. Tu, J. Nathan Kutz, and and. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2(2):165–191, 2015.

[27] Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern Koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.

[28] Steven L Brunton and J Nathan Kutz. *Data-driven Science and Engineering: Machine learning, Dynamical systems, and Control*. Cambridge University Press, 2022.

[29] Marin Bukov, Luca DÁlessio, and Anatoli Polkovnikov. Universal high-frequency behavior of periodically driven systems: from dynamical stabilization to floquet engineering. *Advances in Physics*, 64(2):139–226, March 2015.

[30] Marin Bukov, Alexandre G. R. Day, Dries Sels, Phillip Weinberg, Anatoli Polkovnikov, and Pankaj Mehta. Reinforcement learning in different phases of quantum control. *Phys. Rev. X*, 8:031086, Sep 2018.

[31] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[32] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, August 1995.

[33] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[34] Hui Chen, Hanhan Li, Felix Motzoi, Leigh Martin, K Birgitta Whaley, and Mohan Sarovar. Quantum proportional-integral (PI) control. *New Journal of Physics*, 22(11):113014, 2020.

[35] Shih-I Chu and Dmitry A. Telnov. Beyond the floquet theorem: generalized floquet formalisms and quasienergy methods for atomic and molecular multiphoton processes in intense laser fields. *Physics Reports*, 390(1-2):1–131, February 2004.

[36] Isaac L. Chuang and M. A. Nielsen. Prescription for experimental determination of the dynamics of a quantum black box. *Journal of Modern Optics*, 44(11-12):2455–2467, November 1997.

[37] Tayfun Çimen. State-dependent Riccati equation (SDRE) control: A survey. *IFAC Proceedings Volumes*, 41(2):3761–3775, 2008.

[38] Soledad Le Clainche and José M. Vega. Higher order dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 16(2):882–925, January 2017.

[39] Marcus Cramer, Martin B Plenio, Steven T Flammia, Rolando Somma, David Gross, Stephen D Bartlett, Olivier Landon-Cardinal, David Poulin, and Yi-Kai Liu. Efficient quantum state tomography. *Nature communications*, 1(1):1–7, 2010.

[40] Marcus P. da Silva, Olivier Landon-Cardinal, and David Poulin. Practical Characterization of Quantum Devices without Tomography. *Physical Review Letters*, 107(21):210404, November 2011. Publisher: American Physical Society.

[41] Domenico d'Alessandro. *Introduction to quantum control and dynamics*. Chapman and Hall/CRC, 2021.

[42] Mogens Dalgaard, Felix Motzoi, Jens Jakob Sørensen, and Jacob Sherson. Global optimization of quantum dynamics with alphazero deep exploration. *npj Quantum Information*, 6(1):1–9, 2020.

[43] P De Fouquieres, SG Schirmer, SJ Glaser, and Ilya Kuprov. Second order gradient ascent pulse engineering. *Journal of Magnetic Resonance*, 212(2):412–417, 2011.

[44] Bernard Deconinck and J. Nathan Kutz. Computing spectra of linear operators using the Floquet–Fourier–Hill method. *Journal of Computational Physics*, 219(1):296–321, November 2006.

[45] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(1):2909–2913, jan 2016.

[46] Paul AM Dirac. Note on exchange phenomena in the Thomas atom. In *Mathematical proceedings of the Cambridge philosophical society*, volume 26, pages 376–385. Cambridge University Press, 1930.

[47] D.P. DiVincenzo and D. Loss. Quantum information is physical. *Superlattices and Microstructures*, 23(3-4):419–432, March 1998.

[48] Daoyi Dong and Ian R. Petersen. Quantum control theory and applications: A survey. *IET Control Theory & Applications*, 4(12):2651–2671, December 2010.

[49] Daoyi Dong and Ian R Petersen. Quantum estimation, control and learning: opportunities and challenges. *arXiv preprint arXiv:2201.05835*, 2022.

[50] David L Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

[51] Patrick Doria, Tommaso Calarco, and Simone Montangero. Optimal control technique for many-body quantum dynamics. *Phys. Rev. Lett.*, 106:190501, May 2011.

[52] Jonathan P Dowling and Gerard J Milburn. Quantum Technology: The Second Quantum Revolution. *arXiv preprint quant-ph/0206091*, 2002.

[53] Daniel Dylewsky, Eurika Kaiser, Steven L Brunton, and J Nathan Kutz. Principal component trajectories for modeling spectrally continuous dynamics as forced linear systems. *Physical Review E*, 105(1):015312, 2022.

[54] Daniel J Egger and Frank K Wilhelm. Adaptive hybrid optimal quantum control for imprecisely characterized systems. *Physical Review Letters*, 112(24):240503, 2014.

[55] Jens Eisert, Marcus Cramer, and Martin B Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of modern physics*, 82(1):277, 2010.

[56] Jens Eisert, Dominik Hangleiter, Nathan Walk, Ingo Roth, Damian Markham, Rhea Parekh, Ulysse Chabaud, and Elham Kashefi. Quantum certification and benchmarking. *Nature Reviews Physics*, 2(7):382–390, 2020.

[57] David L. Elliott. *Bilinear control systems: matrices in action*, volume 169. Springer, 2009.

[58] Suguru Endo, Jinzhao Sun, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational quantum simulation of general processes. *Physical Review Letters*, 125(1):010501, 2020.

[59] Utku Eren, Anna Prach, Başaran Bahadır Koçer, Saša V Raković, Erdal Kayacan, and Behçet Açıkmeşe. Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7):1541–1566, 2017.

[60] N. Benjamin Erichson, Lionel Mathelin, J. Nathan Kutz, and Steven L. Brunton. Randomized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 18(4):1867–1891, January 2019.

[61] Paolo Falcone, Francesco Borrelli, Jahan Asgari, Hongtei Eric Tseng, and Davor Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on control systems technology*, 15(3):566–580, 2007.

[62] David D Fan, Ali-akbar Agha-mohammadi, and Evangelos A Theodorou. Deep learning tubes for tube MPC. In *Robotics: Science and Systems XVI (2020)*, 2020.

[63] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, June 1982.

[64] Carl Folkestad and Joel W Burdick. Koopman NMPC: Koopman-based learning and

nonlinear model predictive control of control-affine systems. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7350–7356. IEEE, 2021.

[65] Jacov Frenkel. *Wave mechanics: Advanced general theory*, volume 1. Clarendon Press, Oxford, 1934.

[66] J. M. Geremia and Herschel Rabitz. Optimal Identification of Hamiltonian Information by Closed-Loop Laser Control of Quantum Systems. *Physical Review Letters*, 89(26):263902, December 2002. Publisher: American Physical Society.

[67] Dimitris Giannakis, Amelia Henriksen, Joel A Tropp, and Rachel Ward. Learning to forecast dynamical systems from streaming data. *arXiv preprint arXiv:2109.09703*, 2021.

[68] Steffen J Glaser, Ugo Boscain, Tommaso Calarco, Christiane P Koch, Walter Köckenberger, Ronnie Kosloff, Ilya Kuprov, Burkhard Luy, Sophie Schirmer, Thomas Schulte-Herbrüggen, et al. Training Schrödinger's cat: Quantum optimal control. *The European Physical Journal D*, 69(12):1–24, 2015.

[69] Michael Goerz, Daniel Basilewitsch, Fernando Gago-Encinas, Matthias G Krauss, Karl P Horn, Daniel M Reich, and Christiane Koch. Krotov: A Python implementation of krotov's method for quantum optimal control. *SciPost Physics*, 7(6):080, 2019.

[70] Michael H Goerz, Daniel M Reich, and Christiane P Koch. Optimal control theory for a unitary operation under dissipative evolution. *New Journal of Physics*, 16(5):055012, 2014.

[71] Alex Goeßmann, M Götte, Ingo Roth, Ryan Sweke, Gitta Kutyniok, and Jens Eisert. Tensor network approaches for learning non-linear dynamical laws. *arXiv preprint arXiv:2002.12388*, 2020.

[72] Andy Goldschmidt, Eurika Kaiser, Jonathan L Dubois, Steven L Brunton, and J Nathan Kutz. Bilinear dynamic mode decomposition for quantum control. *New Journal of Physics*, 23(3):033035, 2021.

[73] Andy J Goldschmidt, Jonathan L DuBois, Steven L Brunton, and J Nathan Kutz. Model predictive control for robust quantum state preparation. *arXiv preprint arXiv:2201.05266*, 2022.

[74] Daniel Görges. Relations between model predictive control and reinforcement learning. *IFAC-PapersOnLine*, 50(1):4920–4928, 2017.

[75] Ion Victor Gosea and Igor Pontes Duff. Toward fitting structured nonlinear systems by means of dynamic mode decomposition. In *Model Reduction of Complex Dynamical Systems*, pages 53–74. Springer, 2021.

[76] Debdipta Goswami and Derek A Paley. Global bilinearization and controllability of control-affine nonlinear systems: A koopman spectral approach. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 6107–6112. IEEE, 2017.

[77] Sébastien Gros, Mario Zanon, Rien Quirynen, Alberto Bemporad, and Moritz Diehl. From linear to nonlinear MPC: Bridging the gap via the real-time iteration. *International Journal of Control*, 93(1):62–80, 2020.

[78] Stefanie Günther, N. Anders Petersson, and Jonathan L. DuBois. Quandary: An open-source C++ package for high-performance optimal control of open quantum systems. In *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*, pages 88–98, 2021.

[79] Jutho Haegeman, J Ignacio Cirac, Tobias J Osborne, Iztok Pižorn, Henri Verschelde, and Frank Verstraete. Time-dependent variational principle for quantum lattices. *Physical review letters*, 107(7):070601, 2011.

[80] Serge Haroche and J.-M. Raimond. *Exploring the Quantum: Atoms, cavities, and photons*. Oxford University Press, 2006.

[81] Robin Harper, Steven T. Flammia, and Joel J. Wallman. Efficient learning of quantum noise. *Nature Physics*, 16(12):1184–1188, August 2020.

[82] IN Hincks, CE Granade, Troy W Borneman, and David G Cory. Controlling quantum devices with nonlinear hardware. *Physical Review Applied*, 4(2):024012, 2015.

[83] Foek T Hioe and Joseph H Eberly. N-level coherence vector and higher conservation laws in quantum optics and quantum mechanics. *Physical Review Letters*, 47(12):838, 1981.

[84] Bin-Lun Ho and Rudolf E Kálmán. Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548, 1966.

[85] H. J. Hogben, M. Krzystyniak, G. T. P. Charnock, P. J. Hore, and Ilya Kuprov. Spinach – A software library for simulation of spin dynamics in large spin systems. *Journal of Magnetic Resonance*, 208(2):179–194, February 2011.

[86] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[87] Brian E Jackson, Tarun Punnoose, Daniel Neamati, Kevin Tracy, Rianna Jitosho, and Zachary Manchester. ALTRO-C: A fast solver for conic model-predictive control. In *International Conference on Robotics and Automation (ICRA), Xi'an, China*, page 8, 2021.

[88] Daniel FV James, Paul G Kwiat, William J Munro, and Andrew G White. On the measurement of qubits. In *Asymptotic Theory of Quantum Statistical Inference: Selected Papers*, pages 509–538. World Scientific, 2005.

[89] Juan L Jerez, Paul J Goulart, Stefan Richter, George A Constantinides, Eric C Kerrigan, and Manfred Morari. Embedded online optimization for model predictive control at megahertz rates. *IEEE Transactions on Automatic Control*, 59(12):3238–3251, 2014.

[90] J Robert Johansson, Paul D Nation, and Franco Nori. QuTiP: An open-source Python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 183(8):1760–1772, 2012.

[91] J.R. Johansson, P.D. Nation, and Franco Nori. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 184(4):1234–1240, apr 2013.

[92] Kadierdan Kaheman, Urban Fasel, Jason J Bramburger, Benjamin Strom, J Nathan Kutz, and Steven L Brunton. The experimental multi-arm pendulum on a cart: A benchmark system for chaos, learning, and control. *arXiv preprint arXiv:2205.06231*, 2022.

[93] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society of London A*, 474(2219), 2018.

[94] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of koopman eigenfunctions for control. *Machine Learning: Science and Technology*, 2(3):035023, 2021.

[95] Rudolf Emil Kalman. Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(2):152–192, 1963.

[96] Julian Kelly, Rami Barends, Brooks Campbell, Yu Chen, Zijun Chen, Ben Chiaro, Andrew Dunsworth, Austin G Fowler, I-C Hoi, Evan Jeffrey, et al. Optimal quantum control using randomized benchmarking. *Physical review letters*, 112(24):240504, 2014.

[97] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J. Glaser. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance*, 172(2):296 – 305, 2005.

[98] Morten Kjaergaard, Mollie E Schwartz, Jochen Braumüller, Philip Krantz, Joel I-J Wang, Simon Gustavsson, and William D Oliver. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, 11:369–395, 2020.

[99] Martin Kliesch and Ingo Roth. Theory of quantum system certification. *PRX Quantum*, 2(1):010201, 2021.

[100] Stefan Klus, Patrick Gelß, Sebastian Peitz, and Christof Schütte. Tensor-based dynamic mode decomposition. *Nonlinearity*, 31(7):3359–3380, June 2018.

[101] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, may 1931.

[102] B. O. Koopman and J. v. Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, mar 1932.

[103] Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

[104] Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018.

[105] Robert L Kosut, Tak-San Ho, and Herschel Rabitz. Quantum system compression: A hamiltonian guided walk through hilbert space. *Physical Review A*, 103(1):012406, 2021.

[106] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, 2019.

[107] I. Kurniawan, G. Dirr, and U. Helmke. Controllability aspects of quantum dynamics: A unified approach for closed and open systems. *IEEE Transactions on Automatic Control*, 57(8):1984–1996, August 2012.

[108] J. Nathan. Kutz, Steven L. Brunton, Bingni W. Brunton, and Joshua L. Proctor. *Dynamic Mode Decomposition*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, November 2016.

[109] J. Nathan Kutz, Xing Fu, and Steven L. Brunton. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, January 2016.

[110] Henning Lange, Steven L Brunton, and J Nathan Kutz. From Fourier to Koopman: Spectral Methods for Long-term Time Series Prediction. *J. Mach. Learn. Res.*, 22(41):1–38, 2021.

[111] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[112] Ernest Bruce Lee and Lawrence Markus. Foundations of optimal control theory. Technical report, Minnesota Univ Minneapolis Center For Control Sciences, 1967.

[113] Jay H Lee and N Lawrence Ricker. Extended Kalman filter based nonlinear model predictive control. *Industrial & Engineering Chemistry Research*, 33(6):1530–1541, 1994.

[114] Gerd Leuchs and Dagmar Bruss. *Quantum Information: From foundations to quantum technology applications*. John Wiley & Sons, 2019.

[115] Boxi Li, Shahnawaz Ahmed, Sidhant Saraogi, Neill Lambert, Franco Nori, Alexander Pitchford, and Nathan Shammah. Pulse-level noisy quantum circuits with qutip. *Quantum*, 6:630, 2022.

[116] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. Hybrid quantum-classical approach to quantum optimal control. *Physical review letters*, 118(15):150503, 2017.

[117] Ying Li and Simon C Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7(2):021050, 2017.

[118] Li-zhi Liao and Christine A Shoemaker. Advantages of differential dynamic programming over newton's method for discrete-time optimal control problems. Technical report, Cornell University, 1992.

[119] Daniel Liberzon. *Calculus of Variations and Optimal Control Theory*. Princeton University Press, 2011.

[120] Daniel A Lidar. Lecture notes on the theory of open quantum systems. *arXiv preprint arXiv:1902.00967*, 2019.

[121] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.

[122] Brett T Lopez, Jean-Jacques E Slotine, and Jonathan P How. Dynamic tube MPC for nonlinear systems. In *2019 American Control Conference (ACC)*, pages 1655–1662. IEEE, 2019.

[123] Hannah Lu and Daniel M Tartakovsky. Prediction accuracy of dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 42(3):A1639–A1662, 2020.

[124] Christian Lubich. *From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis*. European Mathematical Society Publishing House, September 2008.

[125] Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1), November 2018.

[126] Senka Maćešić, Nelida Črnjarić-Žic, and Igor Mezić. Koopman operator family spectrum for nonautonomous systems. *SIAM Journal on Applied Dynamical Systems*, 17(4):2478–2515, January 2018.

[127] A. G. J. MacFarlane, Jonathan P. Dowling, and Gerard J. Milburn. Quantum technology: the second quantum revolution. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 361(1809):1655–1674, August 2003. Publisher: Royal Society.

[128] Shai Machnes, Elie Assémat, David Tannor, and Frank K Wilhelm. Tunable, flexible, and efficient optimization of control pulses for practical qubits. *Physical review letters*, 120(15):150401, 2018.

[129] Shai Machnes, Urgen Sander, Steffen J Glaser, Pierre de Fouquieres, Audrunas Gruslys, Sonia Schirmer, and Thomas Schulte-Herbrüggen. Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework. *Physical Review A*, 84(2):022305, 2011.

[130] Easwar Magesan, J. M. Gambetta, and Joseph Emerson. Scalable and robust randomized benchmarking of quantum processes. *Phys. Rev. Lett.*, 106:180504, May 2011.

[131] Easwar Magesan and Jay M Gambetta. Effective Hamiltonian models of the cross-resonance gate. *Physical Review A*, 101(5):052308, 2020.

[132] Wilhelm Magnus. On the exponential solution of differential equations for a linear operator. *Communications on Pure and Applied Mathematics*, 7(4):649–673, November 1954.

[133] Krithika Manohar, Bingni W Brunton, J Nathan Kutz, and Steven L Brunton. Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86, 2018.

[134] Alexandre Mauroy, Igor Mezić, and Yoshihiko Susuki. *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Springer International Publishing, 2020.

[135] David Q Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95, 1966.

[136] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[137] David Q Mayne, María M Seron, and SV Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[138] David C McKay, Thomas Alexander, Luciano Bello, Michael J Biercuk, Lev Bishop, Jiayin Chen, Jerry M Chow, Antonio D Córcoles, Daniel Egger, Stefan Filipp, et al. Qiskit backend specifications for OpenQASM and OpenPulse experiments. *arXiv preprint arXiv:1809.03452*, 2018.

[139] A. D. McLachlan. A variational solution of the time-dependent Schrodinger equation. *Molecular Physics*, 8(1):39–44, 1964.

[140] Seth T. Merkel, Jay M. Gambetta, John A. Smolin, Stefano Poletto, Antonio D. Córcoles, Blake R. Johnson, Colm A. Ryan, and Matthias Steffen. Self-consistent quantum process tomography. *Physical Review A*, 87(6):062119, June 2013. Publisher: American Physical Society.

[141] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.

[142] Igor Mezic. Analysis of fluid flows via spectral properties of the Koopman operator. *Annual Review of Fluid Mechanics*, 45:357–378, 2013.

[143] Igor Mezic and Amit Surana. Koopman mode Decomposition for Periodic/Quasi-periodic Time Dependence. *IFAC-PapersOnLine*, 49(18):690–697, January 2016.

[144] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. Model-based Reinforcement Learning: A survey. *arXiv preprint arXiv:2006.16712*, 2020.

[145] M. Mohseni, A. T. Rezakhani, and D. A. Lidar. Quantum-process tomography: Resource analysis of different strategies. *Physical Review A*, 77(3), March 2008.

[146] Felix Motzoi, Jay M Gambetta, Patrick Rebentrost, and Frank K Wilhelm. Simple pulses for elimination of leakage in weakly nonlinear qubits. *Physical review letters*, 103(11):110501, 2009.

[147] DM Murray and SJ Yakowitz. Differential dynamic programming and newton's method for discrete optimal control problems. *Journal of Optimization Theory and Applications*, 43(3):395–414, 1984.

[148] John A Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[149] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[150] Murphy Yuezhen Niu, Sergio Boixo, Vadim N Smelyanskiy, and Hartmut Neven. Universal quantum control through deep reinforcement learning. *npj Quantum Information*, 5(1):1–8, 2019.

[151] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[152] Feliks Nüske, Sebastian Peitz, Friedrich Philipp, Manuel Schaller, and Karl Worthmann. Finite-data error bounds for Koopman-based prediction and control. *arXiv preprint arXiv:2108.07102*, 2021.

[153] M Ohliger, V Nesme, and J Eisert. Efficient and feasible state tomography of quantum many-body systems. *New Journal of Physics*, 15(1):015024, January 2013.

[154] Samuel E Otto and Clarence W Rowley. Koopman operators for estimation and control of dynamical systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:59–87, 2021.

[155] José P Palao and Ronnie Kosloff. Optimal control theory for unitary transformations. *Physical Review A*, 68(6):062308, 2003.

[156] Pantita Palittapongarnpim, Peter Wittek, Ehsan Zahedinejad, Shakib Vedaie, and Barry C. Sanders. Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics. *Neurocomputing*, 268:116–126, December 2017.

[157] Adriano Macarone Palmieri, Egor Kovlakov, Federico Bianchi, Dmitry Yudin, Stanislav Straupe, Jacob D. Biamonte, and Sergei Kulik. Experimental neural network enhanced quantum tomography. *npj Quantum Information*, 6(1):1–5, February 2020. Number: 1 Publisher: Nature Publishing Group.

[158] Sebastian Peitz and Stefan Klus. Koopman operator-based model reduction for switched-system control of PDEs. *Automatica*, 106:184–191, 2019.

[159] Sebastian Peitz, Samuel E Otto, and Clarence W Rowley. Data-driven model predictive control using interpolated koopman generators. *SIAM Journal on Applied Dynamical Systems*, 19(3):2162–2193, 2020.

[160] Seth D Pendergrass, J Nathan Kutz, and Steven L Brunton. Streaming GPU singular value and dynamic mode decompositions. *arXiv preprint arXiv:1612.07875*, 2016.

[161] Minh Q Phan and Seyed Mahdi B Azad. Model predictive Q-learning (MPQ-L) for bilinear systems. In *Modeling, Simulation and Optimization of Complex Processes HPSC 2018*, pages 97–115. Springer, 2021.

[162] Karl Popper. *Conjectures and Refutations: The Growth of Scientific Knowledge*. Routledge, 2002 [1963].

[163] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, January 2016.

[164] Thomas Propson, Brian E Jackson, Jens Koch, Zachary Manchester, and David I Schuster. Robust quantum optimal control with trajectory optimization. *Physical Review Applied*, 17(1):014036, 2022.

[165] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[166] Saša V Raković and William S Levine. *Handbook of model predictive control*. Springer, 2018.

[167] Mark S. Rudner and Netanel H. Lindner. Band structure engineering and non-equilibrium dynamics in floquet topological insulators. *Nature Reviews Physics*, 2(5):229–244, May 2020.

[168] Daniel T. Sank. *Fast, Accurate State Measurement in Superconducting Qubits*. PhD thesis, University of California, Santa Barbara, ProQuest Dissertations Publishing, 9 2014.

[169] Mohan Sarovar, Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. Detecting crosstalk errors in quantum information processors. *Quantum*, 4:321, 2020.

[170] Manuel Schaller, Karl Worthmann, Friedrich Philipp, Sebastian Peitz, and Feliks Nüske. Towards efficient and reliable prediction-based control using eDMD. *arXiv preprint arXiv:2202.09084*, 2022.

[171] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White. Efficient measurement of quantum dynamics via compressive sensing. *Physical Review Letters*, 106(10), March 2011.

[172] A. Shabani, M. Mohseni, S. Lloyd, R. L. Kosut, and H. Rabitz. Estimation of many-body quantum Hamiltonians via compressive sensing. *Physical Review A*, 84(1):012107, July 2011. Publisher: American Physical Society.

[173] Yuan Shi, Alessandro R Castelli, Ilon Joseph, Vasily Geyko, Frank R Graziani, Stephen B Libby, Jeffrey B Parker, Yaniv J Rosen, and Jonathan L DuBois. Quantum computation of three-wave interactions with engineered cubic couplings. *arXiv preprint arXiv:2004.06885*, 2020.

[174] Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I Schuster, Henry Hoffmann, and Frederic T Chong. Optimized compilation of aggregated instructions for realistic quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1031–1044, 2019.

[175] Jon H. Shirley. Solution of the schrödinger equation with a hamiltonian periodic in time. *Phys. Rev.*, 138:B979–B987, May 1965.

[176] Henrique Silvério, Sebastián Grijalva, Constantin Dalyac, Lucas Leclerc, Peter J Karalekas, Nathan Shammah, Mourad Beji, Louis-Paul Henry, and Loïc Henriet. Pulser: An open-source package for the design of pulse sequences in programmable neutral-atom arrays. *Quantum*, 6:629, 2022.

[177] John A Smolin, Jay M Gambetta, and Graeme Smith. Efficient method for computing the maximum-likelihood quantum state from measurements with additive gaussian noise. *Physical review letters*, 108(7):070502, 2012.

[178] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.

[179] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[180] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.

[181] Yuval Tassa, Nicolas Mansard, and Emo Todorov. Control-limited differential dynamic programming. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1168–1175. IEEE, 2014.

[182] N Timoney, V Elman, S Glaser, C Weiss, M Johanning, W Neuhauser, and Chr Wunderlich. Error-resistant single-qubit gates with trapped ions. *Physical Review A*, 77(5):052334, 2008.

[183] Emanuel Todorov and Weiwei Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 300–306. IEEE, 2005.

[184] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, February 2018.

[185] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

[186] L. M. K. Vandersypen and I. L. Chuang. NMR techniques for quantum control and computation. *Reviews of Modern Physics*, 76(4):1037–1069, January 2005. Publisher: American Physical Society.

[187] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[188] John Von Neumann. The Mathematician. In R. B. Heywood, editor, *The Works of the Mind*, pages 180–196. University of Chicago Press, 1947.

[189] John von Neumann, Robert T. Beyer, and Nicholas A. Wheeler. *Mathematical foundations of quantum mechanics*. Princeton University Press, 2018 edition, 1932.

[190] Jianwei Wang, Stefano Paesani, Raffaele Santagati, Sebastian Knauer, Antonio A. Gentile, Nathan Wiebe, Maurangelo Petruzzella, Jeremy L. O'Brien, John G. Rarity, Anthony Laing, and Mark G. Thompson. Experimental quantum Hamiltonian learning. *Nature Physics*, 13(6):551–555, June 2017.

[191] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *IEEE Transactions on control systems technology*, 18(2):267–278, 2009.

[192] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to Control: A locally linear latent dynamics model for control from raw images. *Advances in Neural Information Processing Systems*, 28, 2015.

[193] Max Werninghaus, Daniel J Egger, Federico Roy, Shai Machnes, Frank K Wilhelm, and Stefan Filipp. Leakage reduction in fast superconducting qubit gates via optimal control. *npj Quantum Information*, 7(1):1–6, 2021.

[194] Nathan Wiebe, CE Granade, Christopher Ferrie, and D. G. Cory. Hamiltonian learning and certification using quantum resources. *Physical Review Letters*, 112(19), May 2014.

[195] Howard M Wiseman and Gerard J Milburn. *Quantum measurement and control*. Cambridge University Press, 2009.

[196] Nicolas Wittler, Federico Roy, Kevin Pack, Max Werninghaus, Anurag Saha Roy, Daniel J Egger, Stefan Filipp, Frank K Wilhelm, and Shai Machnes. Integrated tool set for control, calibration, and characterization of quantum devices applied to superconducting qubits. *Physical Review Applied*, 15(3):034080, 2021.

[197] Xian Wu, SL Tomarken, N Anders Petersson, LA Martinez, Yaniv J Rosen, and Jonathan L DuBois. High-fidelity software-defined quantum logic on a superconducting qudit. *Physical Review Letters*, 125(17):170502, 2020.

[198] Tao Xin, Sirui Lu, Ningping Cao, Galit Anikeeva, Dawei Lu, Jun Li, Guilu Long, and Bei Zeng. Local-measurement-based quantum state tomography via neural networks. *npj Quantum Information*, 5(1):1–8, 2019.

[199] Pengcheng Yang, Min Yu, Ralf Betzholz, Christian Arenz, and Jianming Cai. Complete quantum-state tomography with a local random field. *Physical Review Letters*, 124(1), January 2020.

[200] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, 2019.

[201] Ehsan Zahedinejad, Joydip Ghosh, and Barry C. Sanders. High-fidelity single-shot toffoli gate via quantum control. *Physical Review Letters*, 114(20), May 2015.

[202] Hao Zhang, Clarence W Rowley, Eric A Deem, and Louis N Cattafesta. Online dynamic mode decomposition for time-varying systems. *SIAM Journal on Applied Dynamical Systems*, 18(3):1586–1609, 2019.

[203] Jun Zhang and Mohan Sarovar. Quantum hamiltonian identification from measurement time traces. *Physical Review Letters*, 113(8), August 2014.

[204] Tianhao Zhang, Gregory Kahn, Sergey Levine, and Pieter Abbeel. Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 528–535. IEEE, 2016.