Restricted Boltzman Machine Variational parametrisation of the time-evolution operator of periodically driven systems

German A. Sinuco Leon
School of Mathematics and Physical Sciences, University of Sussex, Falmer, BN1 9QH, United Kingdom.
(Dated: April 27, 2020)

Here we explore the use of a Restricted Boltzman Machine to parametrise the Floquet states of harmonically driven qubit. The requiured complexity of the RBM with the number of Floquet manifolds. Solution of the Floquet spectrum converges from an approximated solution the Rotating Wave Approximation at. The training of the RBM presens all the standar challenges of slow convergency. Combination of loss functions .

Keywords: Floquet, Quantum dynamics, Dressed states, Restricted Boltzman Machine, Machine Learning, Periodic driving, qubit

I. INTRODUCTION

out-ofpequilibrium states driven by harmonic forces, transformation of states in . Dynamical decoupling with periodic drivings (several harmonics as in the quantum coputing talk I saw), or multiquibe gates by multiple harmonci forces, to floque topological insulators and frequency/time-domain quantum simulations.

The time-evolution operator is key, either to define the time-evolution or for defniniton of effective Hamiltonians. both of them cna be written interms of the Floquet states. Floquet states perturbatoin theory, RWA, Bloch-siegart, ... expansion.... All this approximaiton and general solution are important to uderststand, fore example.

ML representation of quantum states has been recently very active, showing ... . and compatibility with on-line l earnign and experimental realsiation, in partiuclar for the control of quantums states in quantum computing architectures .... control with floquet.... The reductin of the parametere required, reflection on the ... of the physically accessible states of the Hilbert space, corresponding to very restriced subste, which can be described efffecgively in nothe such as MPS, tensor Netowors.. Such a relatio nbetween ML and ... is currently an active area of research ..

Hrmonically deriven systems can be in the ferquency space has the as a Hubbard model [**?**], and such that time-dependennt problem can be evaluated using toools of static systems [], which has not been explored sufficiently. Here I study the parametrisation of the Floquet operator using a RBM for the archetypical case of a driven qubit. The RBM is overkilling in this case,with other straight numerical diagonalisaiton of the Floquet Hamiltonian, however it allows to explore the capabilities of, which can be then applied to other more complex systems where exact diagonalisation or cannot be implemented so directly.

The docume is as follows. In Section **??** I present the representation of the time-evlutio operator in terms of Floquet states. In section **??**, presents the Restricted Boltzman Machine rerpesentatio of the states and the loss fucntions to evaluate the Floquet spectrum. Section IV shows the representation of Floquet states using the RBM. The central result in presented in section **??**, where I discuss the evalation of the Floquet spectrum (eigenvectors and eigenvalues) evaluated using an RBM parametrisaiton. Discussion of the applcations are presented and conclusion are in sectino...

FIG. 1. (a) Schematic energy level structure of a generic quantum system. The basis of states consist of a discrete set of energy states, which define several bands according to the level energy spacing. Inter and intra band coupling is induced by electromagnetic radiation tuned at the corresponding frequencies, as indicated by the coupling terms. The wide variety of physical systems described by this model includes (b) trapped ions [**?**], (c) superconducting qubits [**?**] and (d) diamond NV-centres [**?**].

II. FLOQUET FORMALISM

The openMMF library is developed to calculate the time-evolution operator, $U(t',t)$, $t'>t$, of systems whose Hamiltonian has the form:

$$H=\sum_{i,j}^{D}E_{i,j}\left|i\right\rangle\left\langle j\right|+\sum_{i,j}^{D}\sum_{\ell=1}^{N}\sum_{n\in Z}V_{i,j}^{\ell,n}e^{in\omega_{\ell}t}\left|i\right\rangle\left\langle j\right|+\text{h.c.} \tag{1}$$

where D is the dimension of the Hilbert space, $E_{i,j}$ defines the static component of the Hamiltonian H, $V_{i,j}^{\ell,n}$ is the coupling between the states i and j oscillating at frequency $n\omega_{\ell}$ (i.e. the n-th harmonic of the ℓ-th fundamental frequency ω_{ℓ}) and N is the number of incommensurately frequencies.

To calculate the time-evolution operator we generalise the Rotating (or Resonant) Wave Approximation (RWA), taking into account the complex time dependence of eq. (1). For this, we rephrase the problem in terms of building a time-dependent unitary transformation, $U_F(t)$ to a new basis $\{|\bar{i}\rangle\}$, that leads to a *time-independent* and diagonal Hamiltonian, $\bar{H}$. After applying the standard quantum-mechanical transformation rule to the Schrödinger equation [**?** **?**], this condition

becomes:

$$U_F^\dagger(t)\left[H(t)-i\hbar\partial_t\right]U_F(t)=\sum_{\bar{i}}\bar{E}_{\bar{i}}\left|\bar{i}\right\rangle\left\langle\bar{i}\right| \tag{2}$$

Importantly, in the basis of states defined by this transformation the time evolution operator is diagonal and has the form:

$$\bar{U}(t',t)=\sum_{\bar{i}}e^{-i\bar{E}_{\bar{i}}(t'-t)}\left|\bar{i}\right\rangle\left\langle\bar{i}\right| \tag{3}$$

which let us to calculate the time evolution operator in the original basis $\{|i\rangle\}$, just by inverting the transformation $U_F(t)$, according to [**?**]:

$$U(t',t)=U_F(t')\bar{U}(t',t)U_F(t) \tag{4}$$

To formulate a fully defined computational problem, we express the micromotion operator $U_F(t)$ as the multifrequency Fourier series [**?**]:

$$U_F(t)=\sum_{\vec{n}}U^{\vec{n}}_{i,\bar{i}}e^{-i\vec{\omega}\cdot\vec{n}t}\left|i\right\rangle\left\langle\bar{i}\right| \tag{5}$$

where $\vec{\omega}=(\omega_1,\omega_2,\ldots,\omega_N)$ and $\vec{n}$ is a N-dimensional vector of integers. After plugging this expansion in eq. (2) and performing an integral over time, we obtain a fully defined eigenproblem for the eigenvalues $\bar{E}_{\bar{i}}$ and Fourier components of the unitary transformation $U^{\vec{n}}_{i,\bar{E}}$:

$$\sum_j(E_{i,j}-\hbar\vec{n}\cdot\vec{\omega})U^{\vec{n}}_{j,\bar{i}}+\sum_j\sum_{\vec{m}}\left[V^{\vec{m}}_{i,j}U^{\vec{n}+\vec{m}}_{j,\bar{i}}+V^{\vec{m}*}_{ji}U^{\vec{m}-\vec{n}}_{j,\bar{i}}\right]=\bar{E}_{\bar{i}}U^{\vec{n}}_{i,\bar{i}} \tag{6}$$

where the couplings $V^{\ell,n}_{i,j}$ define $V^{\vec{n}}_{i,j}$ and the vector $\vec{n}=(0,\ldots,m,\ldots,0)$ with the value m located at the $\ell-$th position. To obtain a finite matrix representation of this problem we truncate the sum over the number of modes of the Fourier expansion eq. (5). Below, in Appendix A, we show an specific example of the shape of the matrix for a bichromatic driven problem.

This formulation to calculate the time-evolution operator is equivalent to the multimode Floquet representation of the Hamiltonian that introduces the extended Hilbert space $|E_i,\vec{n}\rangle$ [**?** **?** **?**]. However, the semiclassical description presented here makes emphasis in the experimentally accessible states, which usually are used to express the static part of the Hamiltonian eq. (1).

In this paper we consider the Hamiltonian of quaibit system:

$$H=\hbar\omega_0S_z+\hbar\Omega S_x\cos(\omega t+\phi) \tag{7}$$

In the regime of off-resonant + weak-to-strong driving. More precisely, my numerical results are obtained for the parameters $\omega=1.7\omega_0$ and a Rabi frequency $\Omega\in 0,10\times\omega_0$. Figure presents the energy gap between the dressed states as a function of the Rabi frequency Ω and the eigen states of selected parameters.

The wavefucntion has the symmetry: some symmetry

which is used to reduce the number of paramerter below.

III. RBM PARAMETRISATION OF THE MICROMOTION OPERATOR

Any wavefumction of the form:

$$|\bar{i}\rangle=\sum_{j,n}u^n_{j,\bar{i}}e^{in\omega t}\left|j\right\rangle \tag{8}$$

The coeffecients of the expansion can be parametrised using:

$$u^n_{j,\bar{i}}=\sqrt{\frac{P\lambda(\sigma)}{Z}}\exp(\phi_\mu(\sigma)) \tag{9}$$

where

$$P_\kappa(\sigma)=\sum_{\vec{h}}p_\kappa(\sigma,\vec{h}) \tag{10}$$

with

$$p_\kappa(\sigma,\vec{h})=exp(\vec{b}_{\vec{\sigma}}\cdot\vec{\sigma}+\vec{c}_{\vec{h}}\cdot\vec{h}+\vec{h}^t\cdot W^{\vec{h}}_{\vec{\sigma}}\cdot\vec{\sigma}) \tag{11}$$

IV. RBM PARAMETRISATION OF FLOQUET STATES

The numerically exact Floquet states (the fourier coeffiecients of expansion....) straigh forward by diagonalisin eq [**?** **?**].

teh Kull-lll loss function: k $=\sum_i p_i\log\frac{p_i}{p_i}$

to train both RBm parametrisation of the amplitude and the phase, I loop swapping one, starting of a random initiailization of the RBM parameters.

Typical training problems and parameters. Accuracy.

V. EVALUATING THE FLOQUET SPECTRUM USING ARBM

Finding the full spectrum of quasienergies and the .

As an initial guess for the parameters, we train the RBM to fit the RWA, which can be evluated in . The second is the definition of a loss fucntion. IN this case we wnat a that the matrix ooperion... lead to a diagnonal form .we define distance as the difference between the of the values with the correponding diagonal elelmemtn. The training of this .. shown in ...

As a second form of the loss function is a quantification of the diffenece between the lefhs vectors and the initila vectors. They should only differr in the scale, such . the candidate eigenvalue is chooslen as teh ratio between ... and .... Then we evaluate the difference. The traiing .. suffer similar difficulties taking a long number of steps and requrieing a small loss rate.

Combination of the two loss functions during traingin in a randomly between loss functions. We observe a spped up of the traiing, improve of the fidelity with the esxac, as well an improvment of the numerically exact Floquet spectrum.

VI. DISCUSSION

Loss function with slow learning. Investigation in better loss function as well as dragging tools from ML to speep up ..

The difficutly of training consittues a tool for charactersing physicsl system as ... . Hre we have observed that complex wavefuntion rquire more traiing effort. application for example to floque driven systems.

Applicaton for evaluating the longtime evluation. nitial guess functions to improve convegencey. COnversely, the training wave functin and the distribution/correlation of the coefficients correlated with othe critier of the system..

The application of symmetries an boundary coditions to the floquet states. Also, this can be readly extended to multimode scaling of multimodedriven system (with unconmensurable frequencies).

RBM parametrisation fits any function.

Floquet states requiring more Fourier components are harder to train.

several ways to define a loss function.

Slow gradident

random selection of criteria similar to ensemble learning

the RWA approximation is a good start generically and evolvs towards the solution. sppeding up the convergence ML .

More interesting is the constuction of the iniital guess and restriction of the solutions explored, for example that the amplitude of Floquet manifolds should be small.

here we explored the RBM parametrisation to build the micromotion operator. Other parameteisation can be more efficient for optimisation. Such construction of the initial guess and constrains can be come from using Tensor Networks parametrisation, or ...

VII. CONCLUSSION

In this paper we present a premier explorative study of the use of RBM for the parametriation of Floquet operators, in an archetypical periodically driven system. We obtain that trining of the can be done , which is equivalent to the experimentally demonstrated in .. .with online learning of the wavefunction.

The evaluation of the Floquet spectrum is a mor difficult task. he initial guess guides the minimun of th e defined loss function. simple definitos vse on the prpertis of a diaognal matris and present low grdients . combination of loss funciton sa better, reflection on the ensemble learning combination fo other simple archigecture of the wave fucntion, eg a NN can led to for more comple systems tahtn the studeid here.

In this work we present a initial exploration of using RBM for Floqeut problems. the Floquet states staes can be parametrised efficiently following similar approches, even with an altenative fitting of the complex and the wavefunction amplitude. this task can be qualified as easy, rapid from random distributon.

Finding the Floquet sates is a harder task. using as initial guess the RWA training . the function has a slow after a fast decline. however a combination of loss funcitons satisfied by any dagnonal matrix or the eigne vectors helps to guide the search. configuraiotn.. . Perhaps a different . The micromotion operator is the timeevolution operator, then other ML architehqure mighth present better. Also

This starting point for exploratioon of the use of parametrisatio for more complex ...like .. . The numerical effor for the case studied here is overkilling, however we explore typical taht might be present in other driven quantum systems.

ACKNOWLEDGEMENTS

We acknowledge fruitful comments and input from Dr. Juan Sebastian Totero Gongora. This work has been supported by the School of Mathematics and Physics of the University of Sussex.

APPENDIX A: RESTRICED BOLTZMAN MACHINE PARAMETRISATION OF $u^n_{j,\bar{i}}$

APPENDIX B: TYPICAL AND NO SO TYPICAL TRAINING RESULTS

APPENDIX A: TYPICAL EXAMPLE OF THE MATRIX REPRESENTATION OF THE MULTIMODE HAMILTONIAN

A typical shape of the matrix representation of the RHS of eq. (6). Here we consider a static system H_0

driven by the three frequencies: $\omega_1, 2\omega_1, \omega_2$. The integer array describing this set of frequencies must be $(0, 2, 1)$. For the two fundamental frequencies, the number of Fourier modes in the decomposition of the micromotion operator is chosen to be $N_F = 3$. With this, the driving of ω_1 and its first harmonic leads to the matrix:

$$\mathcal{H}_1 = \begin{pmatrix} H_0 + 3\hbar\omega_1 & V^{1,1} & V^{1,2} & 0 & 0 & 0 & 0 \\ V^{1,1\dagger} & H_0 + 2\hbar\omega_1 & V^{1,1} & V^{1,2} & 0 & 0 & 0 \\ V^{1,2\dagger} & V^{1,1\dagger} & H_0 + \hbar\omega_1 & V^{1,1} & V^{1,2} & 0 & 0 \\ 0 & V^{1,2\dagger} & V^{1,1\dagger} & H_0 & V^{1,1} & V^{1,2} & 0 \\ 0 & 0 & V^{1,2\dagger} & V^{1,1\dagger} & H_0 - \hbar\omega_1 & V^{1,1} & V^{1,2} \\ 0 & 0 & 0 & V^{1,2\dagger} & V^{1,1\dagger} & H_0 - 2\hbar\omega_1 & V^{1,1} \\ 0 & 0 & 0 & 0 & V^{1,2\dagger} & V^{1,1\dagger} & H_0 - 3\hbar\omega_1 \end{pmatrix}$$

and the full matrix results in:

$$\mathcal{H} = \begin{pmatrix} \mathcal{H}_1 + 3\hbar\omega_2 & \mathcal{V}^{2,1} & 0 & 0 & 0 & 0 & 0 \\ \mathcal{V}^{2,1\dagger} & \mathcal{H}_1 + 2\hbar\omega_2 & \mathcal{V}^{2,1} & 0 & 0 & 0 & 0 \\ 0 & \mathcal{V}^{2,1\dagger} & \mathcal{H}_1 + \hbar\omega_2 & \mathcal{V}^{2,1} & 0 & 0 & 0 \\ 0 & 0 & \mathcal{V}^{2,1\dagger} & \mathcal{H}_1 & \mathcal{V}^{2,1} & 0 & 0 \\ 0 & 0 & 0 & \mathcal{V}^{2,1\dagger} & \mathcal{H}_1 - \hbar\omega_2 & \mathcal{V}^{2,1} & 0 \\ 0 & 0 & 0 & 0 & \mathcal{V}^{2,1\dagger} & \mathcal{H}_1 - 2\hbar\omega_2 & \mathcal{V}^{2,1} \\ 0 & 0 & 0 & 0 & 0 & \mathcal{V}^{2,1\dagger} & \mathcal{H}_1 - 3\hbar\omega_2 \end{pmatrix}$$

with

$$\mathcal{V}^{2,1} = \begin{pmatrix} V^{2,1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & V^{2,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & V^{2,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & V^{2,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & V^{2,1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & V^{2,1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & V^{2,1} \end{pmatrix}$$

C++ WRAPPER

The library includes wrappers to use with C++. The interfaces to the Fortran subroutines share the name of the aimed function with the appended particle `_C_`. The full set of wrappers are declared in the file `MultimodeFloquet.h` , which must be included in the C++ source code. As an example we present the C++ codes required to evaluate the time-evolution of a driven qubit. The library distribution includes Fortran and C++ examples.

```
#include <iostream>
#include <complex>
#include <stdio.h>
#include <math.h>
#include <string.h>

using namespace std;
typedef std::complex<double> dcmplx;

#include "MultimodeFloquet.h"

extern "C" int h_floquet_size;
```

```
int main(){

  atom_c id;
  int info,N_;
  int jtotal;
  char name[]     = "qubit";
  char manifold[] = "U";

  int r,m,l;
  int d_bare,total_frequencies;

  double t1,t2;

  info   = 0;
  jtotal = 2;
  floquetinit_c(&id,name,&info);

  d_bare = id.d_bare;

  dcmplx * U_AUX = new dcmplx [d_bare*d_bare];

  int nm = 2;
  int * modes_num = new int [nm];

  modes_num[0] = 1;
  modes_num[1] = 1;

  total_frequencies = 0;
  for(r=0;r<nm;r++){
    total_frequencies += modes_num[r];
  }

  mode_c * fields = new mode_c [total_frequencies];


  // --- SET DRIVING PARAMETERS
  fields[0].x    = 0.0;
  fields[0].y    = 0.0;
  fields[0].z    = 2.0;
  fields[0].phi_x = 0.0;
  fields[0].phi_y = 0.0;
  fields[0].phi_z = 0.0;
  fields[0].omega = 0.0;
  fields[0].N_Floquet = 0;

  fields[1].x     = 4.0;
  fields[1].y     = 0.0;
  fields[1].z     = 0.0;
  fields[1].phi_x = 0.0;
  fields[1].phi_y = 0.0;
  fields[1].phi_z = 0.0;
  fields[1].omega = 2.0;
  fields[1].N_Floquet = 3;

  // --- SET THE HAMILTONIAN COMPONENTS
  sethamiltoniancomponents_c_
              (&id,&nm,&total_frequencies,modes_num,fields,&info);
```

```
  // --- BUILD THE MULTIMODE FLOQUET MATRIX AND  FIND ITS SPECTRUM
  multimodefloquetmatrix_c_
   (&id,&nm,&total_frequencies,modes_num,fields,&info);

  double * e_floquet = new double [h_floquet_size];
  dcmplx * U_F =  new dcmplx [h_floquet_size*h_floquet_size];

  lapack_fulleigenvalues_c_
   (U_F,&h_floquet_size,e_floquet,&info);
// the diagonalization is done with the internal (Fortran)
// Hamiltonian (H_FLOQUET) and the calculated
// U_F is the transformation that diagonalise this Hamiltonian
// On the Fortran side, H_FLOQUET is deallocated after
// diagonalization.

   //--- EVALUATE TIME-EVOLUTION OPERATOR IN THE BARE BASIS

   N_ = 256;
   t1= 0.0;
   for(r=1;r<N_;r++){
     t2 = r*100.0/N_;
     multimodetimeevolutionoperator_c_
       (&h_floquet_size,&nm,modes_num,U_F,e_floquet,&d_bare,fields,&t1,&t2,U_AUX,&info);
     for(l=0;l<d_bare*d_bare;l++) p_avg[l] = pow(abs(U_AUX[l]),2);
     write_matrix_c_(p_avg,&d_bare);
   }
    delete(e_floquet);
    delete(U_F);
  }

  return 0;
}
```