Numerical Simulation of Inversion Dynamics in a Two-Level Atom Interacting with a Squeezed Coherent State

Ismaele Lorenzon

January 2025

1 Introduction

This project simulates inversion dynamics for a two-level atom driven by a superposition of Fock states, namely coherent or squeezed. Using the rotating frame of light approximation, dynamics are studied using the interaction term of the Jaynes-Cummings model, a cornerstone of quantum optics theory.

1.1 Theoretical Framework

1.1.1 Bare States

Using an N_{TH}-dimensional Hilbert space approximation, the program aims to simulate the inversion dynamics in the atom by imposing coherent or squeezed states probability distributions as initial conditions for the coefficients of the ansatz:

$$|\psi\rangle = c_{1,n+1}\,|1, n+1\rangle + c_{2,n}\,|2, n\rangle$$

where $|1, n+1\rangle$ and $|2, n\rangle$ represent the light field-atom coupled states (also called bare states). In a coherent state for example we have:

$$|\langle 2, n\,|\psi\rangle\,|^2\Big|_{t=0} = e^{-|\alpha|^2}\,\frac{|\alpha|^{2n}}{n!} = |c_{2,n}(0)|^2$$

as the probability of finding the atom excited state for each Fock state $|n\rangle$ if we choose to prepare the atom in its excited state.

1.1.2 Inversion Dynamics

The global dynamics coincide with the single Fock state dynamics, bounded by the initial conditions described above. Taking the ansatz as described above and plugging it back into Schrodinger's equation using the Jaynes-Cummings interaction hamiltonian

$$\hat{H}_I = \hbar g(\hat{a}^\dagger\hat{\sigma} + \hat{a}\hat{\sigma}^\dagger)$$

we get a set of differential equations

$$\frac{d\ c_{1,n+1}(t)}{dt} = -ig\sqrt{n+1}\ e^{i\Delta t}c_{2,n}(t)$$

$$\frac{d\ c_{2,n}(t)}{dt} = -ig\sqrt{n+1}\ e^{-i\Delta t}c_{1,n+1}(t)$$

where g is the coupling constant of the system, and Δ represents the detuning. From this we get

$$c_{2,n}(t) = c_{2,n}(0)(e^{-\frac{1}{2}i(\Delta-\Omega_n^\Delta)} + e^{-\frac{1}{2}i(\Delta+\Omega_n^\Delta)})$$

with $\Omega_n^\Delta = \sqrt{\Delta^2 + 4g^2(n+1)}$ the generalized Rabi frequency. After some trivial calculations we get

$$|c_{2,n}(t)|^2 = |c_{2,n}(0)|^2 cos^2(t\frac{\Omega_n^\Delta}{2})$$

The inversion is basically the unbalance in probability of finding the atom in its excited state compared to the ground state and it's calculated as

$$w(t) = \sum_{n=0}^{N_{TH}} |c_{2,n}(t)|^2 - \sum_{n=0}^{N_{TH}} |c_{1,n+1}(t)|^2$$

in our N_{TH} dimensional Hilbert space approximation. We only need to calculate one coefficient for each photon number as the other can be derived through normalization requirements.
Mind that we are just analizing the interaction term of the Jaynes-Cummings model, so the state of the light field doesn't evolve as a consequence of the interaction with the atom. To understand better we take the example of a coherent state, which by definition satisfies

$$\hat{a}\,|\alpha\rangle = \alpha\,|\alpha\rangle$$

where a is the destruction operator. We can't say the same about the creation operator $\hat{a}^\dagger$, so the emission of a photon by the atom following spontaneous de-excitation would cause a non-trivial evolution as it is not an eigenstate of said operator. Squeezed coherent states follow even more complex dynamics, with the squeezing parameter changing and tracing a trajectory in the complex plane.

1.1.3 Squeezed States

Squeezed states are particular state of lights where one quadrature X_{ϕ_s} is prepared with an unbalanced certainty, which by Heisenberg uncertainty principle expands the variance on the orthogonal one. Namely

$$\Delta X_{\phi_s} \Delta X_{\phi_s+\frac{\pi}{2}} \geq \frac{1}{4}$$

for generalized quadrature operators.This is usually represented by the action of a squeezing operator $S(\xi)$ where ξ is the squeezing complex parameter

$$\xi = re^{i\phi_s}$$

Making use of the displacement operator to obtain our coherent starting point we get that the final squeezed coherent state can be written as

$$|\alpha, \xi\rangle = S(\xi)D(\alpha)\,|0\rangle$$

where $|0\rangle$ is the vacuum state. Efforts to derive a simplified, computationally optimized, form for the probability distribution of Fock states in squeezed coherent states were tried but the best derivation was found to be

$$p(n) = |\langle n|\alpha, \xi\rangle|^2 = N(n)\frac{1}{\mu n!}\left|\frac{v}{2\mu}\right|^n \left|H_n(\frac{\alpha}{\sqrt{2\mu v}})\right|^2$$

where $\mu = coshr$, $v = sinhr\ e^{i\phi_s}$, H_n represents the Hermite polynomial of n-th order and $N(n)$ is a normalization function.[1]
From this probability distribution we can derive the inversion dynamics as we did in coherent states.

2 Simulation

2.1 Prerequisites

Of course, having Python installed and the *parameters.json* in the same directory as the *PopulationDynamics.py*. Anyway, the parameter file path can be changed in any moment by simply modifying the global variable $PARAM\_FILE$ in the first few lines of the main executable.
The packages needed are

- numpy
- matplotlib
- qutip
- mpmath

Other used packages should be part of the standard distribution of Python 3.10.

[1] Kam, CF., Zhang, WM., Feng, DH. (2023). Squeezed Coherent States. In: Coherent States. Lecture Notes in Physics, vol 1011. Springer, Cham. https://doi.org/10.1007/978-3-031-20766-2$_7$

2.2 Parameters

The simulation parameters are set by the *parameters.json* file in which:

- **G**: coupling constant of the system
- **A**: amplitude for the coherent state parameter α
- **PHI**: phase of the coherent state parameter divided by π (the multiplication by π is done by the software in order to keep a high precision)
- **R**: amplitude of the squeezing parameter ξ
  NB: This parameter should never be 0 for computational stability, in the case of a coherent state choose a very low squeezing with $R << 1$: an order of 10^{-4} can be good enough depending on the precision needed.
- **PHI_SQUEEZE**: phase of the squeezing parameter ϕ_s again divided by π.
- **DETUNING**: detuning defined as

  $$\Delta = \omega_{light} - \omega_{resonance}$$

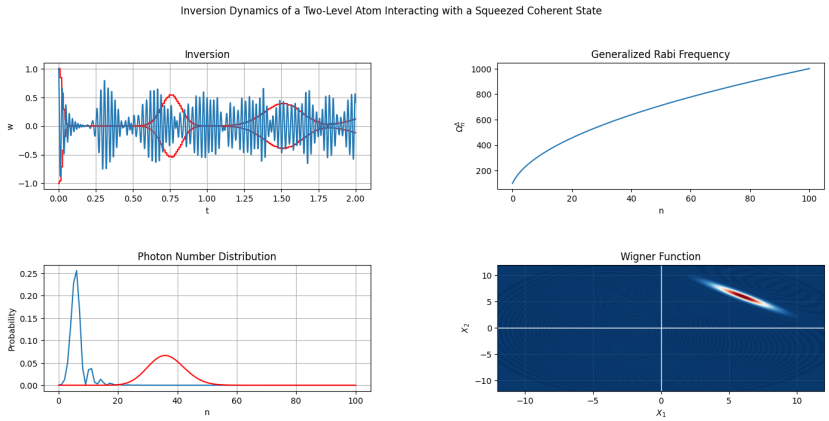
  Blue or red detuning are both admitted.
- **TIME_STOP**: stop time of the simulation
- **TIME_STEP**: amount of time between calculated inversions. Depending on the precision needed, a time step in the order of 10^{-3} of the simulated time can have good enough precision while still being computationally "light".
- **TIME_START**: start time of the simulation, it can be adjusted with TIME_STOP to analyze only a certain frame or part of the oscillation dynamics, not necessarily starting from the beginning.
- **N_TH**: The dimensionality of the approximated Hilbert space. Namely, choosing Fock basis, the threshold number for the highest photon number Fock state dynamics computed.

Another key parameter is defined in the first few lines of *data.py*, which is numerical precision of the simulation, under the $PRECISION$ constant. A standard value of 25 decimal places of precision is chosen, but it can be freely adjusted, although much higher precisions could impact the time required by the simulation (with N_TH and the number of time steps).

2.3 Running

The program can be run as any other Python program through the command prompt (PopulationDynamics.py) without any arguments or through tools like Visual Studio Code and various IDEs. A current simulated time and approximated remaining time are given, but they serve more as a general idea of the time that will be taken for the simulation to run than an accurate measure, as the number of calculations involved means that the running time is much dependent on cpu/gpu load and the parameters mentioned above, which make it quite hard to predict (which is not the scope of this program in any way).

2.4 Results



Here above the results of the simulation for

```
{
    "G":50,
    "A":6,
    "PHI":0.25,
    "R":1,
    "PHI_SQUEEZE":0.5,
    "DETUNING":0.0,
    "TIME_STOP":2,
    "TIME_STEP":1e-3,
    "TIME_START":0,
    "N_TH":100
}
```

- **Top-Left**: The classical inversion dynamics as a function of time, with the blue denoting the actual dynamics and in red we have the respective coherent state inversion envelope for reference.

- **Bottom-Left**: Photon number distribution, where the blue is the squeezed state and the red one is again the non-squeezed coherent state for reference.
- **Top-Right**: The generalized Rabi frequency is plotted with respect to n. This plot can be useful to determine which term dominated the Rabi frequency calculations, with a more linear evolution denoting a dominance of G with respect to the photon number, and a more curved one (tends to the radical function) denoting a dominance of the term $\sqrt{n+1}$
- **Bottom-Right**: Wigner function of the squeezed coherent state in standard quadrature space.

2.5 Complexity Analysis

Thanks to the implemented caching of the factorials and Hermite functions, which by starting the calculations from $n = 0$ going up take complexity $O(1)$, the complexity of the algorithm is $O(t + pn\ log(p)log(n))$ as the complexity for the time evolution is $O(1)$ for each of the time steps. The other term causes more trouble as the computational complexity of the exponential depends greatly on the package implementation and it is rather difficult to pinpoint a correct approximation. This term represents the complexity of the initial calculation for the coefficients, and the exponential needed for each term can be decomposed as roughly $O(p\ log(p)log(n))$ where p decimal point precision in the calculation, and n corresponds to the dimensionality of the approximated Hilbert space.

It is worth mentioning that some basic measures were introduced to increase the numerical stability of the simulation. Logarithmic decomposition was applied to the calculation of the coefficients to avoid overflow when dealing with large partial products. Instead of directly multiplying large numbers, the logarithms of the coefficients are calculated first. The products are then computed in the logarithmic space, where they become sums. Finally, the exponential is taken of the sum to get the final result. This technique helps to manage large values more efficiently and prevents overflow issues that can occur when directly multiplying large numbers.