example-quasi-steadystate-driven-system

June 25, 2014

1 QuTiP example: Calculate the quasi-steadystate of a time-dependent (period) quantum system

J.R. Johansson and P.D. Nation

For more information about QuTiP see http://qutip.org

Find the steady state of a driven qubit, by finding the eigenstates of the propagator for one driving period

```
In [1]: %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
In [2]: from qutip import *
```

```
In [17]: def hamiltonian_t(t, args):
             #
             # evaluate the hamiltonian at time t.
             #
             H0 = args['H0']
             H1 = args['H1']
             w  = args['w']

             return H0 + H1 * sin(w * t)
```

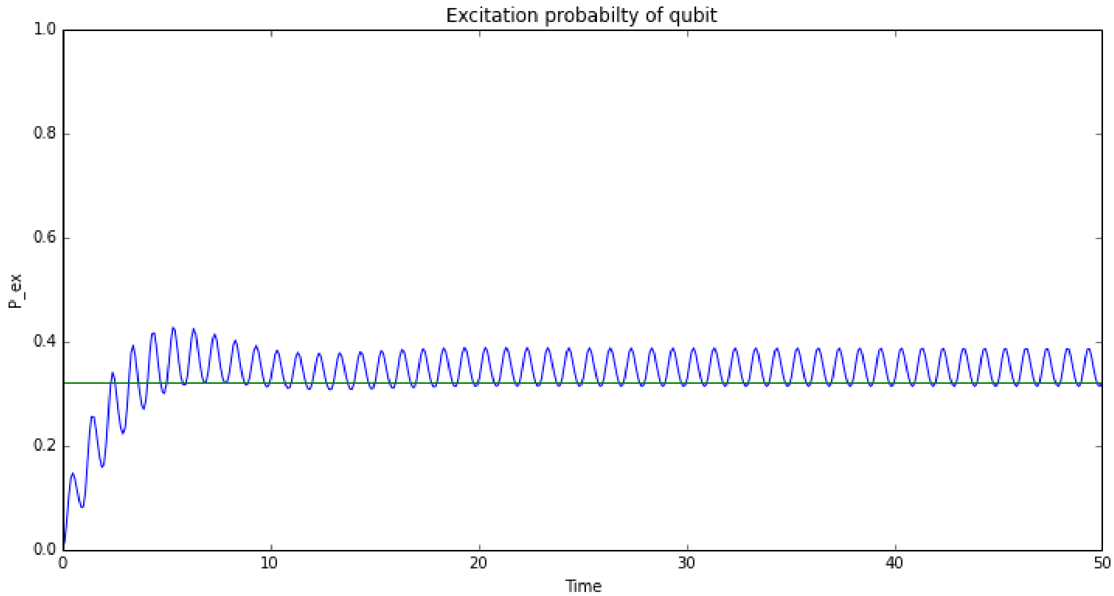
```
In [18]: def sd_qubit_integrate(delta, eps0, A, w, gamma1, gamma2, psi0, tlist):

             # Hamiltonian
             sx = sigmax()
             sz = sigmaz()
             sm = destroy(2)

             H0 = - delta/2.0 * sx - eps0/2.0 * sz
             H1 = - A * sx

             H_args = {'H0': H0, 'H1': H1, 'w': w}
             # collapse operators
             c_op_list = []

             n_th = 0.5 # zero temperature

             # relaxation
             rate = gamma1 * (1 + n_th)
             if rate > 0.0:
                 c_op_list.append(sqrt(rate) * sm)
```

```
    # excitation
    rate = gamma1 * n_th
    if rate > 0.0:
        c_op_list.append(sqrt(rate) * sm.dag())

    # dephasing
    rate = gamma2
    if rate > 0.0:
        c_op_list.append(sqrt(rate) * sz)


    # evolve and calculate expectation values
    output = mesolve(hamiltonian_t, psi0, tlist, c_op_list, [sm.dag() * sm], H_args)

    T = 2 * pi / w

    U = propagator(hamiltonian_t, T, c_op_list, H_args)

    rho_ss = propagator_steadystate(U)

    return output.expect[0], expect(sm.dag() * sm, rho_ss) * ones(shape(tlist))
```

In [19]:
```
delta = 0.3  * 2 * pi   # qubit sigma_x coefficient
eps0  = 1.0  * 2 * pi   # qubit sigma_z coefficient
A     = 0.05 * 2 * pi   # driving amplitude (sigma_x coupled)
w     = 1.0  * 2 * pi   # driving frequency

gamma1 = 0.15          # relaxation rate
gamma2 = 0.05           # dephasing  rate

# intial state
psi0 = basis(2,0)
tlist = linspace(0,50,500)
```

In [20]:
```
p_ex, p_ex_ss = sd_qubit_integrate(delta, eps0, A, w, gamma1, gamma2, psi0, tlist)
```

In [21]:
```
figure(figsize=(12,6))

plot(tlist, real(p_ex))
plot(tlist, real(p_ex_ss))
xlabel('Time')
ylabel('P_ex')
ylim(0,1)
title('Excitation probabilty of qubit');
```



1.1 Software version:

```
In [ ]: from qutip.ipynbtools import version_table

        version_table()
```