Exploratory Project

"Light propagation through one-dimensional interacting open quantum systems"

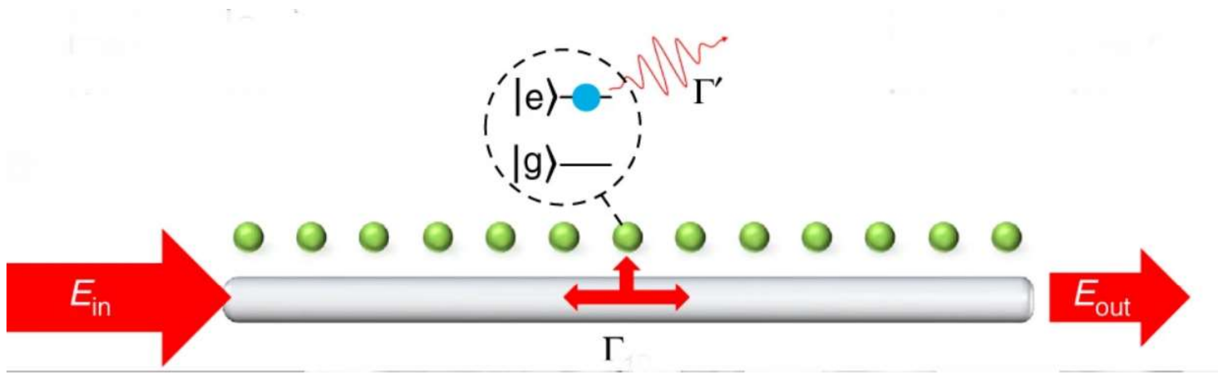
Project Advisor : Dr. Rajeev Singh

Ravi Kumar
Department of Physics
Indian Institute of Technology (B.H.U) Varanasi

Contents

1 Introduction

In this report ,I am presenting my work on "Light propagation through one-dimensional interacting open quantum systems".Our system is basically a one-dimensional interacting spin chain and the atomic medium is coupled to photon baths at the boundaries as shown in figure below .We want to describe the photon transport and its interaction with spin-chains.We here find both transient and steady-state transport properties of a monochromatic laser light passing through an atomic medium modeled as a Heisenberg-type interacting spin- 1/2 chain.We study several properties of the system such as reflection coefficients $\Re_2(t)$,transmission coefficients $\tau_2(t)$ at transient state and at steady state.



In this report, we use the Quantum Langevin Equation QLE approach to calculate nonlinear light propagation through one-dimensional (1D) interacting quantum lattice models connected to photon baths at the boundaries.The QLE approach for nonequilibrium transport is an extension of the Heisenberg-Langevin equation approach to nonequilibrium open quantum systems.We write Quantum Langevin Equation corresponding to the system and use these to describe the system properties.Here, we restrict ourselves to spin chain of two sites.We define the Hamiltonian of the system on the next page for two sites.

2 Hamiltonian for Two Sites

We first apply the QLE approach to an atomic medium of two-atoms modeled as a Heisenberg-type interacting spin- 1/2 chain with nearest-neighbor coupling.
The total Hamiltonian consisting of the atomic medium of two atoms(1),the photon baths(2,3), and the atom-photon coupling(4,5) terms is :

$$H_T = H_M + H_{LB} + M_{RB} + \nu_{LM} + \nu_{RB}$$

$$H_M = 2J_x\sigma_1\sigma_2^\dagger + 2J_x\sigma_1^\dagger\sigma_2 + 4J_z\sigma_1^\dagger\sigma_1\sigma_2^\dagger\sigma_2 + w_1\sigma_1^\dagger\sigma_1 + w_2\sigma_2^\dagger\sigma_2 ....(1)$$

$\sigma_i^\dagger$and σ_i are the raising and lowering operators of the ith 2LA

$$H_{LB} = \textstyle\int_{-\infty}^{\infty} w_k a_k^\dagger a_k dk ....(2)$$

$$H_{RB} = \textstyle\int_{-\infty}^{\infty} w_k b_k^\dagger b_k dk ....(3)$$

$a_k^\dagger$,$b_k^\dagger$create a photon with wave number k, respectively, at the left and right side photon baths.

$$H_{LM} = \textstyle\int_{-\infty}^{\infty} g_L(a_k^\dagger\sigma_1 + \sigma_1^\dagger a_k)dk ....(4)$$

$$H_{RM} = \textstyle\int_{-\infty}^{\infty} g_R(b_k^\dagger\sigma_N + \sigma_N^\dagger b_k)dk ....(5)$$

The atom-photon coupling strength at the left and right sides of the medium are g_L and g_R, respectively.

3 Photon Transport for Two Sites

We are here only interested in getting total transmitted and reflected power which in turn would give transmission and reflection coefficients of light. The time-dependent transmission and reflection coefficients for an incoming light from the left of the medium.

The time-dependent transmission coefficients :

$$\tau_2(t) = \frac{2\Gamma_R}{\nu_g I_{in}} S_{22}(t)$$

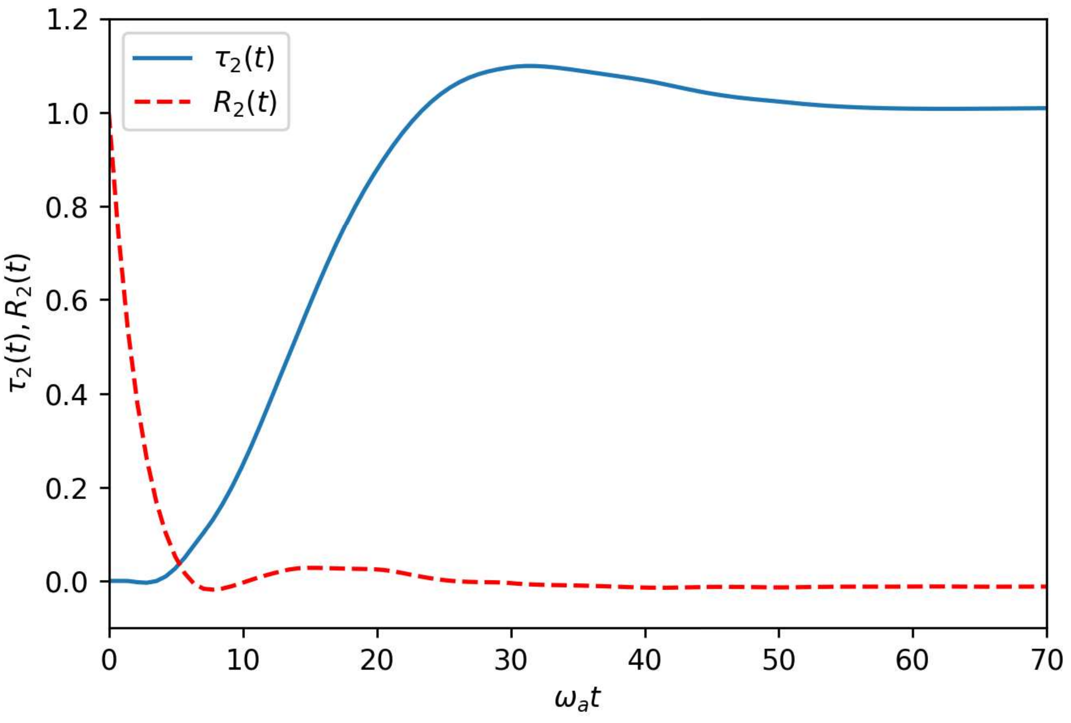
The time-dependent reflection coefficients :

$$\Re_2(t) = 1 + \frac{2\Omega_L}{\nu_g I_{in}} + \frac{2\Gamma_L}{\nu_g I_{in}} S_{11}(t)$$

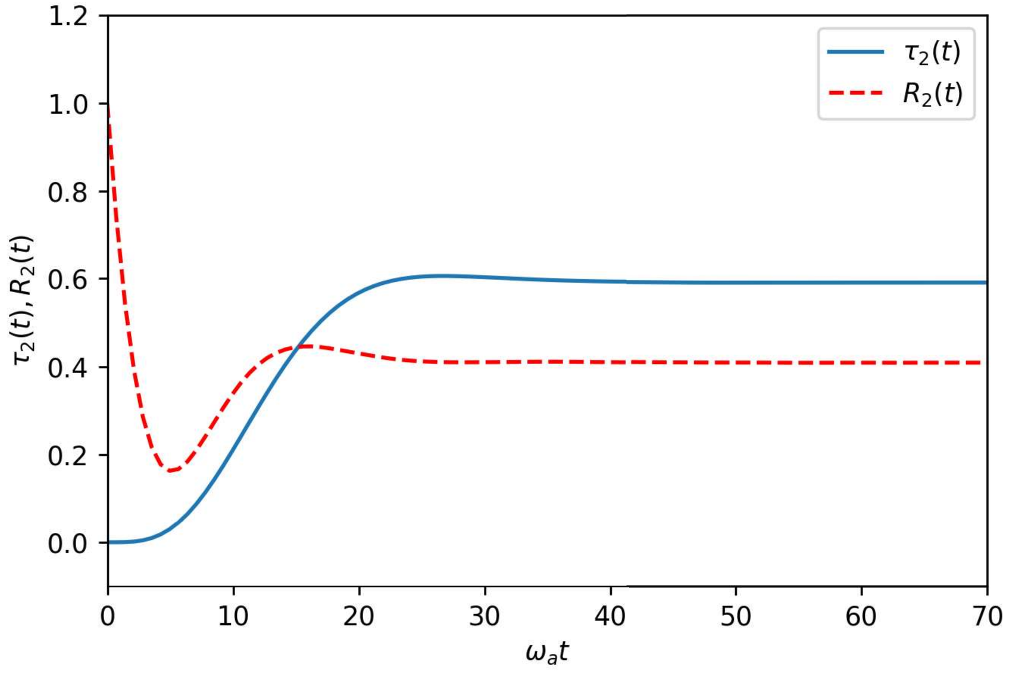
4 Transient Properties of Scattered Light

Transient properties of the scattered light and the atoms in a medium of two atoms driven by a laser light. Figure (1,2) depict time-evolution of transmission coefficient $\tau_2(t)$ and reflection coefficient $\Re_2(t)$ of a laser light from the left of the medium modeled as an interacting spin chain. Figure (3,4) depict time evolution of excited atoms scaled by power of the incident laser.

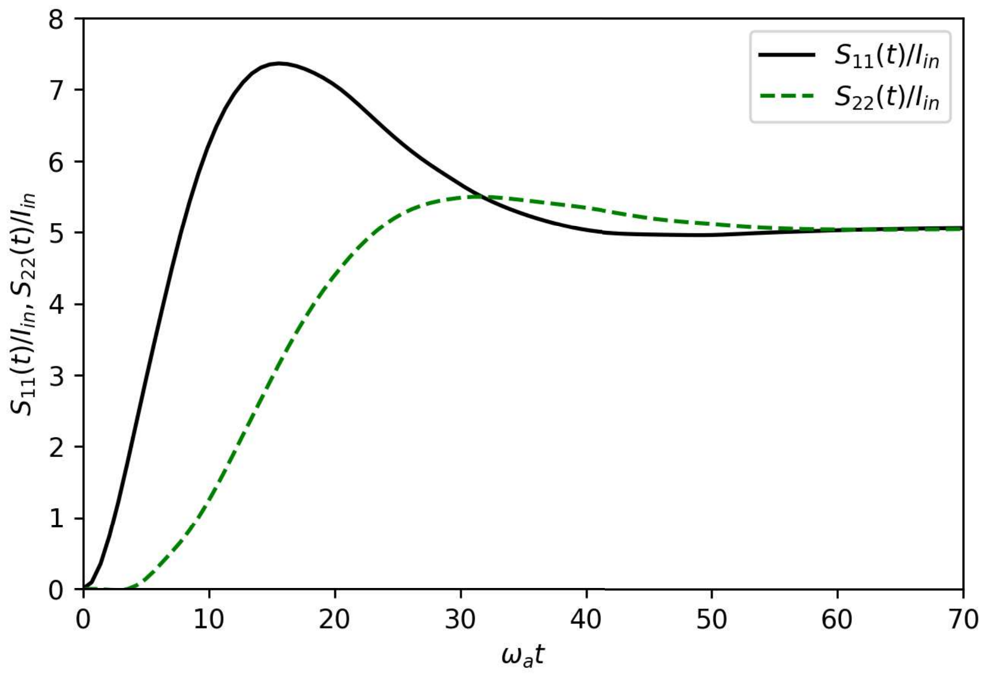
The common parameters are
$\omega_1 = \omega_2 = \omega_p = \omega_a, J_x = J_z = 0.05\omega_a, \Gamma_L = \Gamma_R = 0.1\omega_a.$



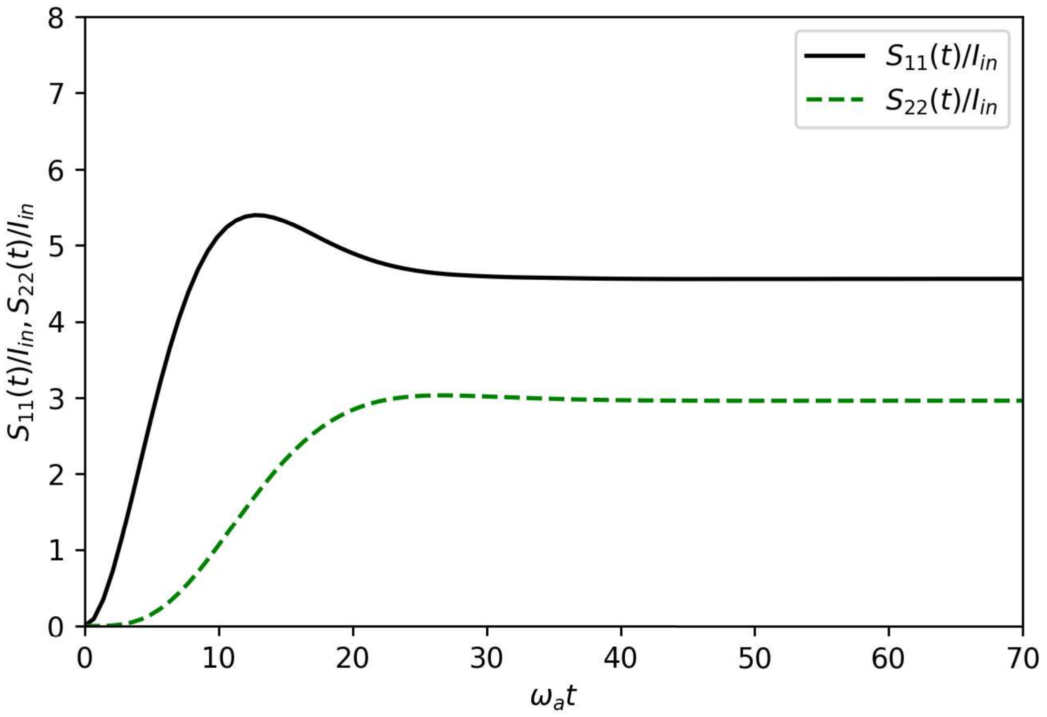
Time-evolution of transmission coefficient $\tau_2(t)$ and reflection coefficient $\Re_2$ when $I_{in} = 1.6x10^{-5}$(in units of ω_a/v_g)



Time-evolution of transmission coefficient $\tau_2(t)$ and reflection coefficient $\Re_2$ when $I_{in} = 0.04$(in units of ω_a/v_g)



Time evolution of excited atoms scaled by power of the incident laser when $I_{in} = 1.6x10^{-5}$(in units of ω_a/v_g)

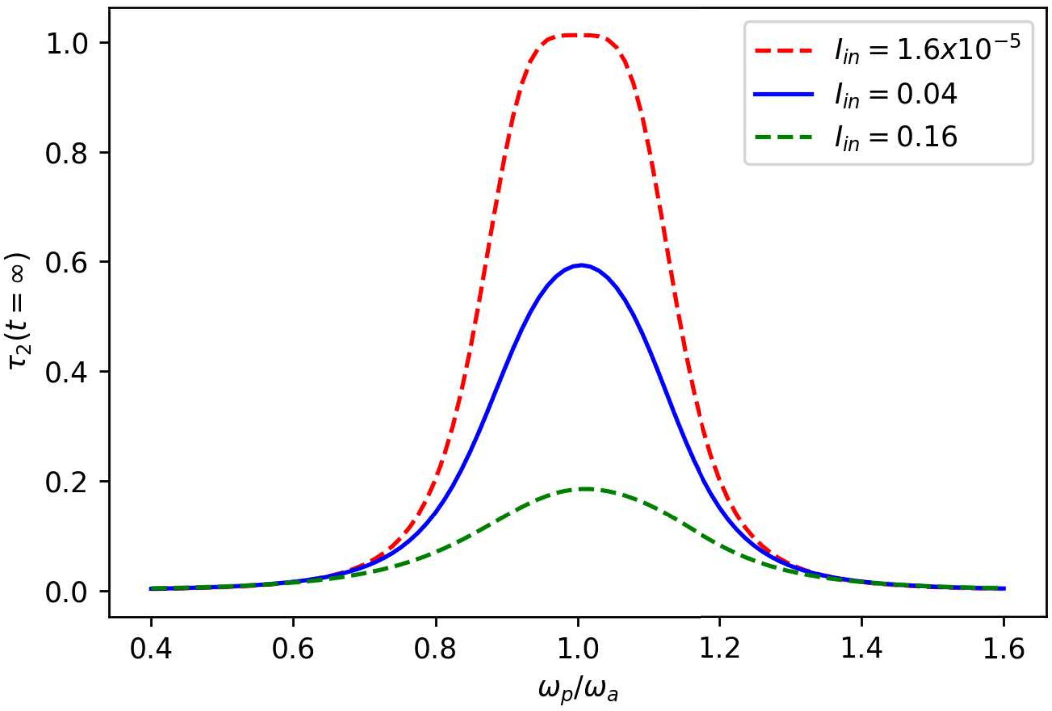


Time evolution of excited atoms scaled by power of the incident laser when $I_{in} = 0.04$(in units of ω_a/v_g)

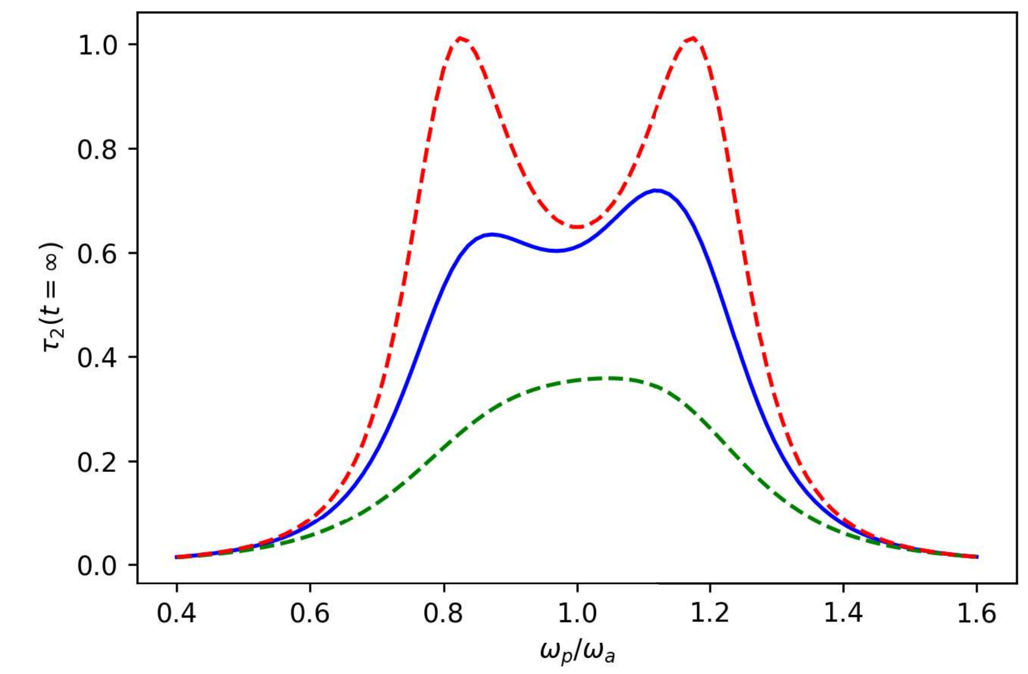
5 Steady State Properties

Linear and nonlinear laser transmission through an atomic medium of two atoms modeled as an interacting spin chain.The steady state transmission coefficient $\tau_2(t=\infty)$ vs scaled frequency w_p/w_a of the laser for various intensities ($I_{in} \propto E_p^2$) of the incident laser.

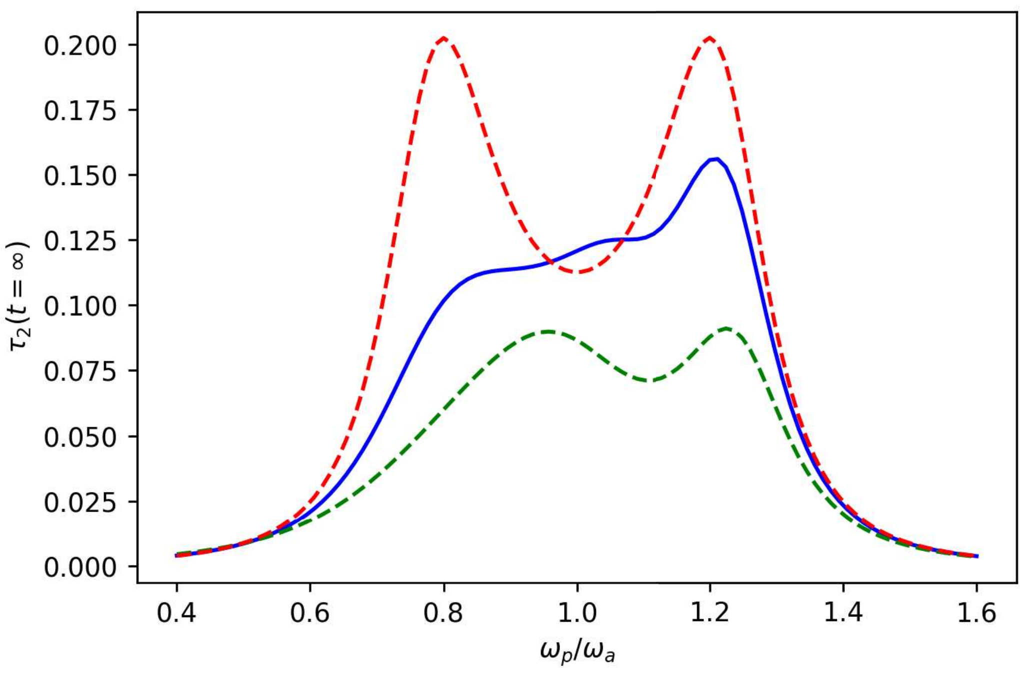
In all the plots $\Gamma_L = \Gamma_R = 0.1, J_z = 0.05$. The rates $\Gamma_L, \Gamma_R, J_x, J_z$ and ω_1, ω_2 are in units of ω_a, and I_{in} is in units of ω/v_g.



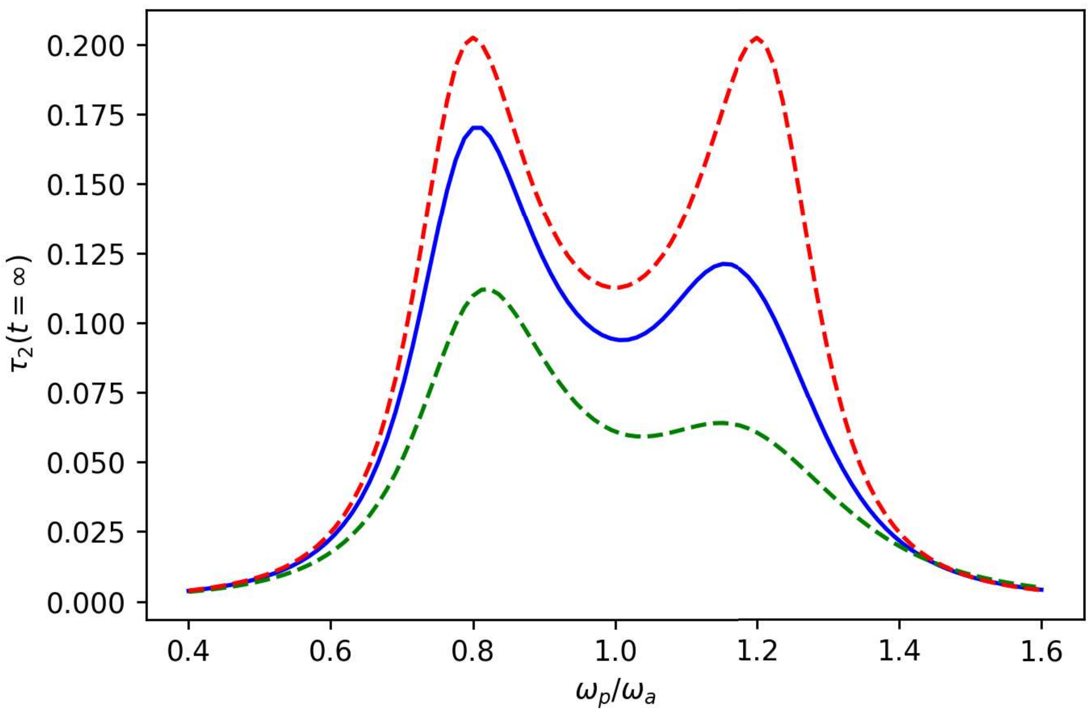
(a)$\omega_1 = \omega_2 = 1, J_x = 0.05$



(b)$\omega_1 = \omega_2 = 1, J_x = 0.1$



(c)$\omega_1 = 0.8, \omega_2 = 1.2, J_x = 0.05$



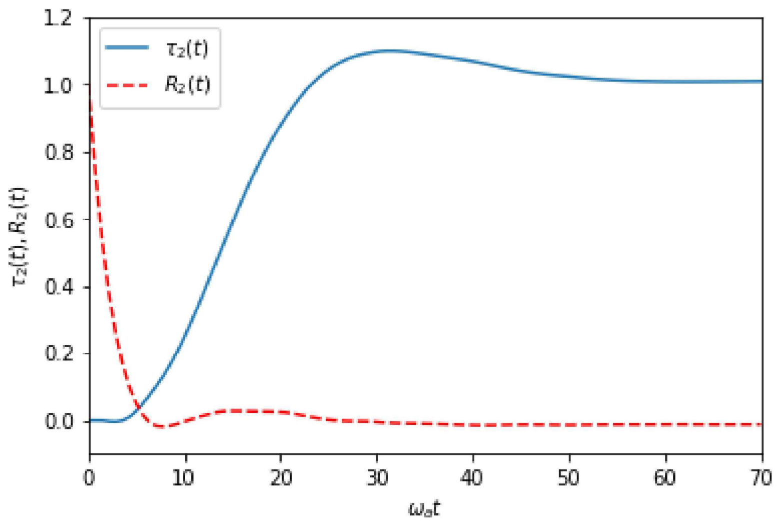
(d)$\omega_1 = 1.2, \omega_2 = 0.8, J_x = 0.05$

In [1]:

```
def TP(In,Ep):
    import numpy as np
    Jx =0.05
    Jz=0.05
    w1=1
    w2=1
    wp=1
    TL=0.1
    TR=0.1
    gl=0.18
    vg=1
    wl=(gl*Ep)/vg
    Z2=np.zeros((15,15),dtype=complex)


    Z2[0][2]=-2J*wl
    Z2[0][3]=2j*Jx
    Z2[0][6]=-4j*Jx
    Z2[0][12]=4j*Jz


    Z2[1][2]=2j*wl
    Z2[1][7]=-2j*Jx
    Z2[1][10]=4j*Jx
    Z2[1][13]=-4j*Jz


    Z2[2][0]=-1j*wl
    Z2[2][1]=1j*wl
    Z2[2][2]=-2*TL
    Z2[2][5]=2j*Jx
    Z2[2][8]=-2j*Jx


    Z2[3][0]=2j*Jx
    Z2[3][6]=4j*Jz
    Z2[3][12]=-4j*Jx

    Z2[4][3]=1j*wl
    Z2[4][6]=-2j*wl

    Z2[5][2]=2j*Jx
    Z2[5][3]=-1j*wl
    Z2[5][5]=-1j*(w1-w2)-TL-TR
    Z2[5][6]=2j*wl
    Z2[5][11]=-2j*Jx

    Z2[6][4]=-1j*wl
    Z2[6][5]=1j*wl
    Z2[6][12]=-2j*Jx

    Z2[7][1]=-2j*Jx
    Z2[7][10]=-4j*Jz
    Z2[7][13]=4j*Jx

    Z2[8][2]=-2j*Jx
    Z2[8][7]=1j*wl
    Z2[8][8]=1j*(w1-w2)-TL-TR
    Z2[8][10]=-2j*wl
    Z2[8][11]=2j*Jx

    Z2[9][7]=-1j*wl
    Z2[9][10]=2j*wl

    Z2[10][8]=-1j*wl
    Z2[10][9]=1j*wl
    Z2[10][13]=2j*Jx

    Z2[11][5]=-2j*Jx
    Z2[11][8]=2j*Jx
    Z2[11][11]=-2*TR

    Z2[12][6]=-2J*Jx
    Z2[12][11]=1j*wl
    Z2[12][14]=-2J*wl

    Z2[13][10]=2j*Jx
    Z2[13][11]=-1j*wl
    Z2[13][14]=2j*wl

    Z2[14][12]=-1j*wl
    Z2[14][13]=1j*wl
    Z2[14][14]=-2*(TL+TR)

    dw1=w1-wp
    dw2=w2-wp
    Z2[0][0]=1j*dw1-TL
    Z2[1][1]=-1j*dw1-TL
    Z2[3][3]=1j*dw2-TR
    Z2[4][4]=1j*(dw1+dw2+4*Jz)-TL-TR
    Z2[6][6]=1j*(dw2+4*Jz)-2*TL-TR
    Z2[7][7]=-1j*dw2-TR
    Z2[9][9]=-1*(1j*(dw1+dw2+4*Jz)+TL+TR)
    Z2[10][10]=-1*(1j*(dw2+4*Jz)+2*TL+TR)
    Z2[12][12]=(1j*(dw1+4*Jz)-TL-2*TR)
    Z2[13][13]=-1*(1j*(dw1+4*Jz)+TL+2*TR)

    W=np.array([1j*wl,-1j*wl,0,0,0,0,0,0,0,0,0,0,0,0,0])

    from scipy.integrate import solve_ivp
    t=np.linspace(0,70,100)
    S0=np.array([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],dtype=complex)
    def S_t(t, S):
        return np.dot(Z2,S) + W
    sol = solve_ivp(S_t,[0,70], S0,t_eval=t)

    S22=np.array((sol.y[11]).real)
    S1=np.array((sol.y[1]).imag)
    S11=np.array((sol.y[2]).real)
    T2=(2*TR)/(vg*In)*S22
    R2=np.array([1]*100)+np.array((2*wl)/(vg*In)*S1)+np.array((2*TR)/(vg*In)*S11)

    return T2 , R2
```

In [2]:

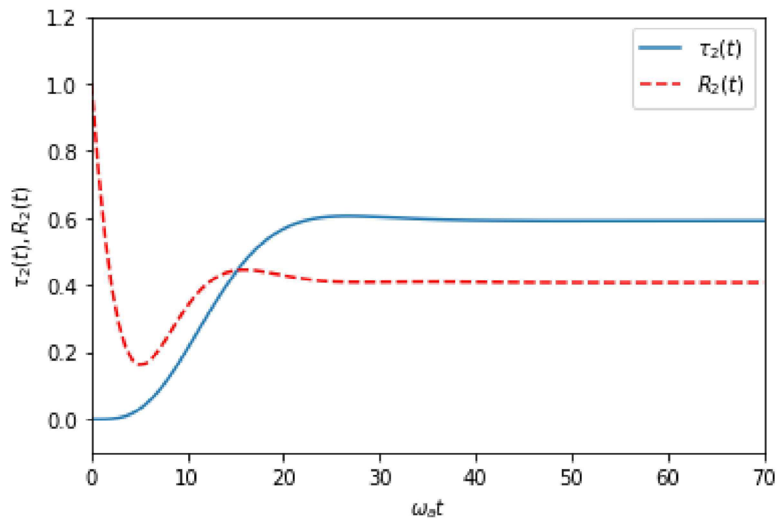
```
import numpy as np
t=np.linspace(0,70,100)
T2,R2=TP(1.6e-5,0.01)
import matplotlib.pyplot as plt
plt.plot(t,T2,label=r'$\tau_2(t)$')
plt.plot(t,R2,"r--",label=r'$R_2(t)$')
plt.axis([0,70,-0.1,1.2])
plt.xlabel(r'$\omega_a t$')
plt.ylabel(r'$\tau_2 (t),R_2(t)$')
plt.legend()

txt="Your Text will appear here like $\omega_a t$"
plt.figtext(0.5, -0.05, txt, wrap=True, horizontalalignment='center', fontsize=12)
plt.savefig("001.png",dpi=250)
plt.show()
```



Your Text will appear here like $\omega_a t$

In [3]:

```
T2,R2=TP(0.04,0.5)
import matplotlib.pyplot as plt
plt.plot(t,T2,label=r'$\tau_2(t)$')
plt.plot(t,R2,"r--",label=r'$R_2(t)$')
plt.axis([0,70,-0.1,1.2])
plt.xlabel(r'$\omega_a t$')
plt.ylabel(r'$\tau_2 (t),R_2(t)$')
plt.legend()
plt.savefig("002.png",dpi=250)
plt.show()
```

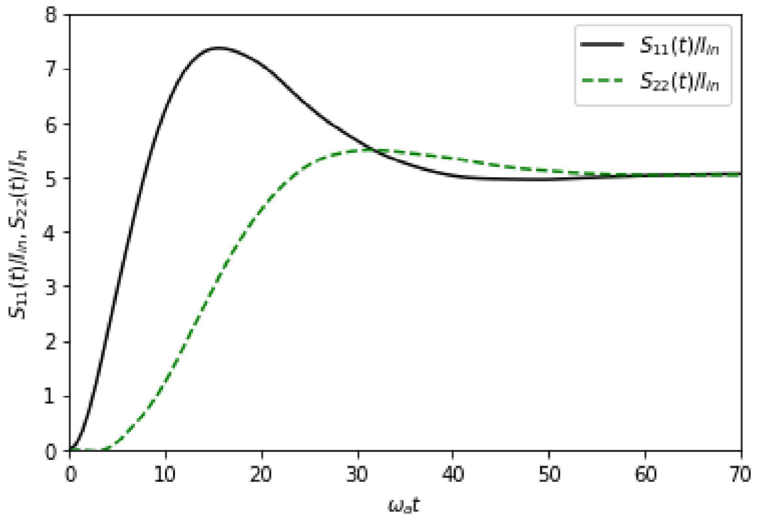


In [4]:

```
def TP2(In,Ep):
    import numpy as np
    Jx =0.05
    Jz=0.05
    w1=1
    w2=1
    wp=1
    TL=0.1
    TR=0.1
    gl=0.18
    vg=1
    wl=(gl*Ep)/vg
    Z2=np.zeros((15,15),dtype=complex)


    Z2[0][2]=-2J*wl
    Z2[0][3]=2j*Jx
    Z2[0][6]=-4j*Jx
    Z2[0][12]=4j*Jz


    Z2[1][2]=2j*wl
    Z2[1][7]=-2j*Jx
    Z2[1][10]=4j*Jx
    Z2[1][13]=-4j*Jz


    Z2[2][0]=-1j*wl
    Z2[2][1]=1j*wl
    Z2[2][2]=-2*TL
    Z2[2][5]=2j*Jx
    Z2[2][8]=-2j*Jx


    Z2[3][0]=2j*Jx
    Z2[3][6]=4j*Jz
    Z2[3][12]=-4j*Jx

    Z2[4][3]=1j*wl
    Z2[4][6]=-2j*wl

    Z2[5][2]=2j*Jx
    Z2[5][3]=-1j*wl
    Z2[5][5]=-1j*(w1-w2)-TL-TR
    Z2[5][6]=2j*wl
    Z2[5][11]=-2j*Jx

    Z2[6][4]=-1j*wl
    Z2[6][5]=1j*wl
    Z2[6][12]=-2j*Jx

    Z2[7][1]=-2j*Jx
    Z2[7][10]=-4j*Jz
    Z2[7][13]=4j*Jx

    Z2[8][2]=-2j*Jx
    Z2[8][7]=1j*wl
    Z2[8][8]=1j*(w1-w2)-TL-TR
    Z2[8][10]=-2j*wl
    Z2[8][11]=2j*Jx

    Z2[9][7]=-1j*wl
    Z2[9][10]=2j*wl

    Z2[10][8]=-1j*wl
    Z2[10][9]=1j*wl
    Z2[10][13]=2j*Jx

    Z2[11][5]=-2j*Jx
    Z2[11][8]=2j*Jx
    Z2[11][11]=-2*TR

    Z2[12][6]=-2J*Jx
    Z2[12][11]=1j*wl
    Z2[12][14]=-2J*wl

    Z2[13][10]=2j*Jx
    Z2[13][11]=-1j*wl
    Z2[13][14]=2j*wl

    Z2[14][12]=-1j*wl
    Z2[14][13]=1j*wl
    Z2[14][14]=-2*(TL+TR)

    dw1=w1-wp
    dw2=w2-wp
    Z2[0][0]=1j*dw1-TL
    Z2[1][1]=-1j*dw1-TL
    Z2[3][3]=1j*dw2-TR
    Z2[4][4]=1j*(dw1+dw2+4*Jz)-TL-TR
    Z2[6][6]=1j*(dw2+4*Jz)-2*TL-TR
    Z2[7][7]=-1j*dw2-TR
    Z2[9][9]=-1*(1j*(dw1+dw2+4*Jz)+TL+TR)
    Z2[10][10]=-1*(1j*(dw2+4*Jz)+2*TL+TR)
    Z2[12][12]=(1j*(dw1+4*Jz)-TL-2*TR)
    Z2[13][13]=-1*(1j*(dw1+4*Jz)+TL+2*TR)

    W=np.array([1j*wl,-1j*wl,0,0,0,0,0,0,0,0,0,0,0,0,0])

    from scipy.integrate import solve_ivp
    t=np.linspace(0,70,100)
    S0=np.array([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],dtype=complex)
    def S_t(t, S):
        return np.dot(Z2,S) + W
    sol = solve_ivp(S_t,[0,70], S0,t_eval=t)

    S22=np.array((sol.y[11]).real)
    S11=np.array((sol.y[2]).real)

    return S22 ,S11
```

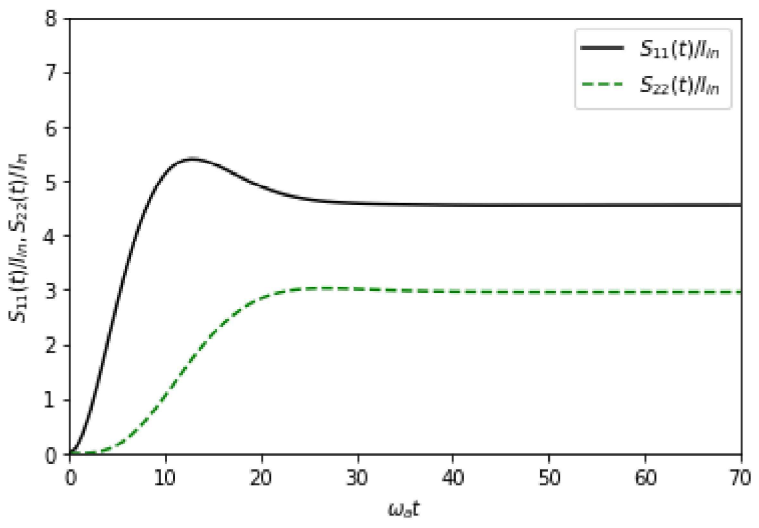
In [5]:

```
S22,S11=TP2(1.6e-5,0.01)
In=1.6e-5
import matplotlib.pyplot as plt
plt.plot(t,S11/In,"k",label=r'$S_{11}(t)/I_{in}$')
plt.plot(t,S22/In,"g--",label=r'$S_{22}(t)/I_{in}$')
plt.xlabel(r'$\omega_a t$')
plt.ylabel(r'$S_{11}(t)/I_{in} , S_{22}(t)/I_{in}$')
plt.axis([0,70,0,8])
plt.legend()
plt.savefig("003.png",dpi=250)
plt.show()
```



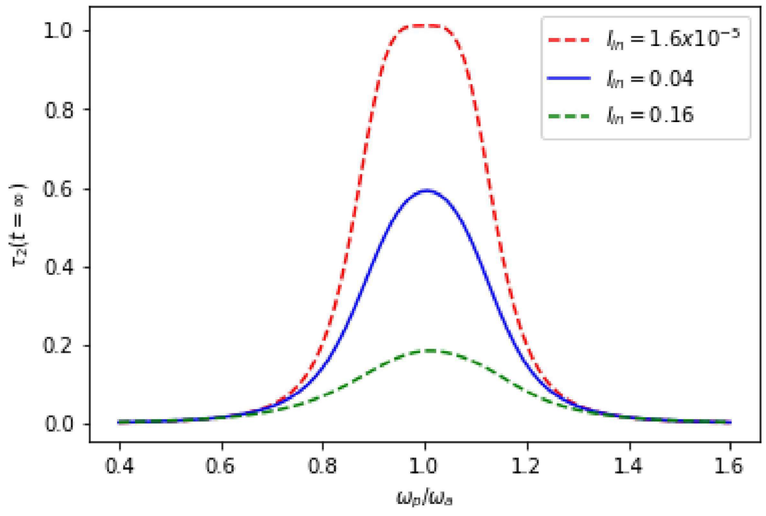
In [6]:

```
S22,S11=TP2(0.04,0.5)
In=0.04
import matplotlib.pyplot as plt
plt.plot(t,S11/In,"k",label=r'$S_{11}(t)/I_{in}$')
plt.plot(t,S22/In,"g--",label=r'$S_{22}(t)/I_{in}$')
plt.xlabel(r'$\omega_a t$')
plt.ylabel(r'$S_{11}(t)/I_{in} , S_{22}(t)/I_{in}$')
plt.axis([0,70,0,8])
plt.legend()
plt.savefig("004.png",dpi=250)
plt.show()
```



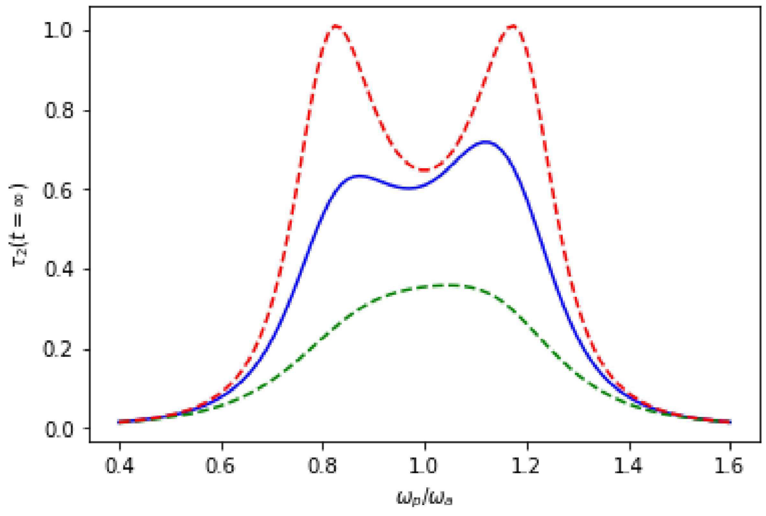
In [1]:

```
def t(In,Ep,Jx,w1,w2):
    import numpy as np
    Jz=0.05
    TL=0.1
    TR=0.1
    gl=0.18
    vg=1
    wl=(gl*Ep)/vg
    Z2=np.zeros((15,15),dtype=complex)


    Z2[0][2]=-2J*wl
    Z2[0][3]=2j*Jx
    Z2[0][6]=-4j*Jx
    Z2[0][12]=4j*Jz


    Z2[1][2]=2j*wl
    Z2[1][7]=-2j*Jx
    Z2[1][10]=4j*Jx
    Z2[1][13]=-4j*Jz


    Z2[2][0]=-1j*wl
    Z2[2][1]=1j*wl
    Z2[2][2]=-2*TL
    Z2[2][5]=2j*Jx
    Z2[2][8]=-2j*Jx


    Z2[3][0]=2j*Jx
    Z2[3][6]=4j*Jz
    Z2[3][12]=-4j*Jx

    Z2[4][3]=1j*wl
    Z2[4][6]=-2j*wl

    Z2[5][2]=2j*Jx
    Z2[5][3]=-1j*wl
    Z2[5][5]=-1j*(w1-w2)-TL-TR
    Z2[5][6]=2j*wl
    Z2[5][11]=-2j*Jx

    Z2[6][4]=-1j*wl
    Z2[6][5]=1j*wl
    Z2[6][12]=-2j*Jx

    Z2[7][1]=-2j*Jx
    Z2[7][10]=-4j*Jz
    Z2[7][13]=4j*Jx

    Z2[8][2]=-2j*Jx
    Z2[8][7]=1j*wl
    Z2[8][8]=1j*(w1-w2)-TL-TR
    Z2[8][10]=-2j*wl
    Z2[8][11]=2j*Jx

    Z2[9][7]=-1j*wl
    Z2[9][10]=2j*wl

    Z2[10][8]=-1j*wl
    Z2[10][9]=1j*wl
    Z2[10][13]=2j*Jx

    Z2[11][5]=-2j*Jx
    Z2[11][8]=2j*Jx
    Z2[11][11]=-2*TR

    Z2[12][6]=-2J*Jx
    Z2[12][11]=1j*wl
    Z2[12][14]=-2J*wl

    Z2[13][10]=2j*Jx
    Z2[13][11]=-1j*wl
    Z2[13][14]=2j*wl

    Z2[14][12]=-1j*wl
    Z2[14][13]=1j*wl
    Z2[14][14]=-2*(TL+TR)

    W=np.array([-1j*wl,+1j*wl,0,0,0,0,0,0,0,0,0,0,0,0,0])
    w_p=np.linspace(0.4,1.6,100)
    s22=[]

    for wp in w_p:
        dw1=w1-wp
        dw2=w2-wp
        Z2[0][0]=1j*dw1-TL
        Z2[1][1]=-1j*dw1-TL
        Z2[3][3]=1j*dw2-TR
        Z2[4][4]=1j*(dw1+dw2+4*Jz)-TL-TR
        Z2[6][6]=1j*(dw2+4*Jz)-2*TL-TR
        Z2[7][7]=-1j*dw2-TR
        Z2[9][9]=-1*(1j*(dw1+dw2+4*Jz)+TL+TR)
        Z2[10][10]=-1*(1j*(dw2+4*Jz)+2*TL+TR)
        Z2[12][12]=(1j*(dw1+4*Jz)-TL-2*TR)
        Z2[13][13]=-1*(1j*(dw1+4*Jz)+TL+2*TR)
        x= np.linalg.solve(Z2,W)
        s22.append(x[11].real)
        s2=np.array(s22)
        k=(2*TR)/(vg*In)
        T=k*s2
    return T

import numpy as np
wp=np.linspace(0.4,1.6,100)
import matplotlib.pyplot as plt

plt.plot(wp,t(1.6e-5,0.01,0.05,1,1),"r--",label=r'$I_{in} =1.6 x 10^{-5}$')
plt.plot(wp,t(0.04,0.5,0.05,1,1),"b",label=r'$I_{in} =0.04$')
plt.plot(wp,t(0.16,1,0.05,1,1),"g--",label=r'$I_{in} =0.16$')

plt.xlabel(r'$\omega_p/\omega_a}$')
plt.ylabel(r'$\tau_2 (t=\infty) $')
plt.legend()
plt.savefig("005.png",dpi=250)
plt.show()
```



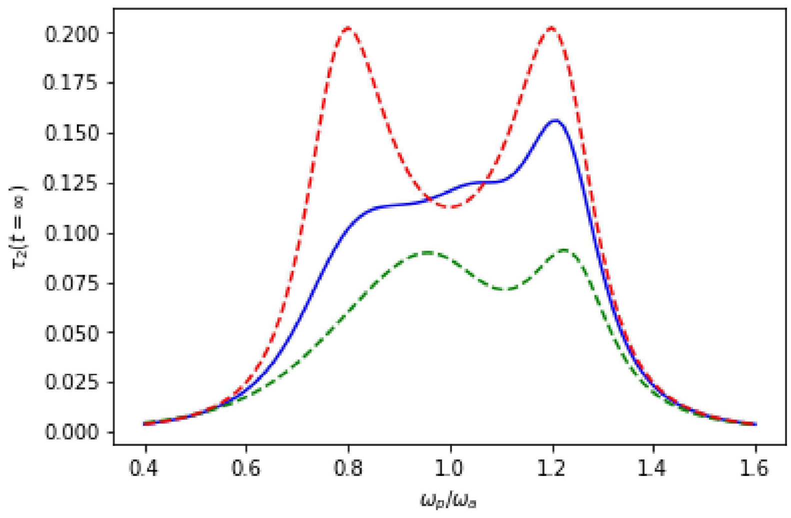
In [2]:

```
wp=np.linspace(0.4,1.6,100)
import matplotlib.pyplot as plt
plt.plot(wp,t(0.04,0.5,0.1,1,1),"b")
plt.plot(wp,t(0.16,1,0.1,1,1),"g--")
plt.plot(wp,t(1.6e-5,0.01,0.1,1,1),"r--")
plt.xlabel(r'$\omega_p/\omega_a}$')
plt.ylabel(r'$\tau_2 (t=\infty) $')
plt.savefig("006.png",dpi=250)
plt.show()
```



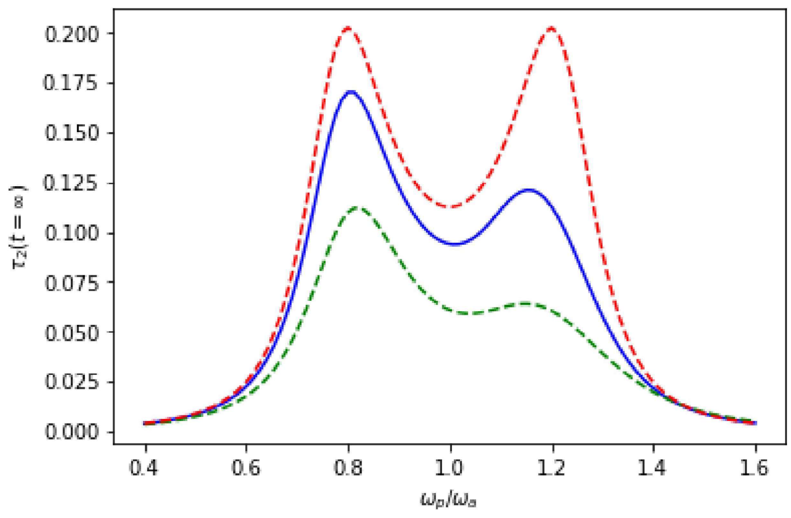
In [3]:

```
wp=np.linspace(0.4,1.6,100)
import matplotlib.pyplot as plt
plt.plot(wp,t(0.04,0.5,0.05,0.8,1.2),"b")
plt.plot(wp,t(0.16,1,0.05,0.8,1.2),"g--")
plt.plot(wp,t(1.6e-5,0.01,0.05,0.8,1.2),"r--")
plt.xlabel(r'$\omega_p/\omega_a}$')
plt.ylabel(r'$\tau_2 (t=\infty) $')
plt.savefig("007.png",dpi=250)
plt.show()
```



In [4]:

```
wp=np.linspace(0.4,1.6,100)
import matplotlib.pyplot as plt
plt.plot(wp,t(0.04,0.5,0.05,1.2,0.8),"b")
plt.plot(wp,t(0.16,1,0.05,1.2,0.8),"g--")
plt.plot(wp,t(1.6e-5,0.01,0.05,1.2,0.8),"r--")
plt.xlabel(r'$\omega_p/\omega_a}$')
plt.ylabel(r'$\tau_2 (t=\infty) $')
plt.savefig("008.png",dpi=250)
plt.show()
```


$\tau_2(t=\infty)$
ω_p/ω_a