

Advanced Laser Technologies Homework #2

Chih-Han Lin student ID: r99245002
Graduate Institute of Applied Physics, National Taiwan University

I. DEMONSTRATION OF MODE-LOCKING PULSES

Assume that complex amplitude has following expression,

$$A(t) = \sum_a A_q \exp\left(\frac{iq2\pi t}{T_f}\right)$$

and pulse intensity $I(t) = |A(t)|^2$. We choose modes $M = 11$ (q sums from 1 to 11) with different complex coefficient A_q as follows,

- (a) Equal magnitude and equal phases, i.e. $A_q = 1$.
- (b) Magnitude that obey the Gaussian spectral profile and equal phase, i.e. $A_q = \exp\left[-\frac{1}{2}\left(\frac{q}{5}\right)^2\right]$.
- (c) Equal magnitude and random phases (uniform distributing between 0 and 2π), i.e. $A_q = \exp(\text{rand}() * 2\pi)$.
- (d) Equal magnitude and phase that obey the Gaussian distribution, i.e. $A_q = \exp\left[-i \cdot 2\pi \frac{1}{2} \left(\frac{q}{5}\right)^2\right]$

II. SIMULATION RESULT

A. Equal magnitude with equal phases

FIG. 1. shows additive intensity function with equal relative phase in time domain and frequency domain respectively. Square Mask in frequency domain would generate ripples between pulses in time domain (oscillation with decay property of sinc function).

B. Gaussian distributed magnitude with equal phases

First, we demonstrate Gaussian distributed magnitude with equal phase, i.e. $A_q = \exp\left[-\frac{1}{2}\left(\frac{q-6}{5}\right)^2\right]$, and the simulation result is shown in FIG. 2. Since the square mask effect is almost neglected comparing to standard deviation of Gaussian distributed spectral profile, the additive ultrashort pulse in time domain would have less ripple than previous case (modes with equal amplitude).

We then increase the mode number ($M=101$) with symmetrical spectrum and observe the dependence between pulse duration and Gaussian spectral width by varying the complex mode's amplitude with:

$$A_q = \sum_a A_q \exp\left(-4\frac{q-50}{50k}\right), \quad k = 1, 2, \dots, 5.$$

The result shown in FIG. 3. says that pulse duration becomes shorter as k increases. If the standard deviation σ of the Gaussian spectrum is large compared with total mode spacing, the shortest pulse duration in time domain is limited by total mode spacing in frequency domain. We may slowly change the σ via computer simulation (seeing FIG. 4.) to specify different operation regime of spectrum control.

C. Equal magnitude with random phases

If mode relative phase is a random number between 0 and 2π , the additive pulse would have lower AC component and generate randomly periodic signal which not specified as any usable pulses. The simulation result is shown in FIG. 5.

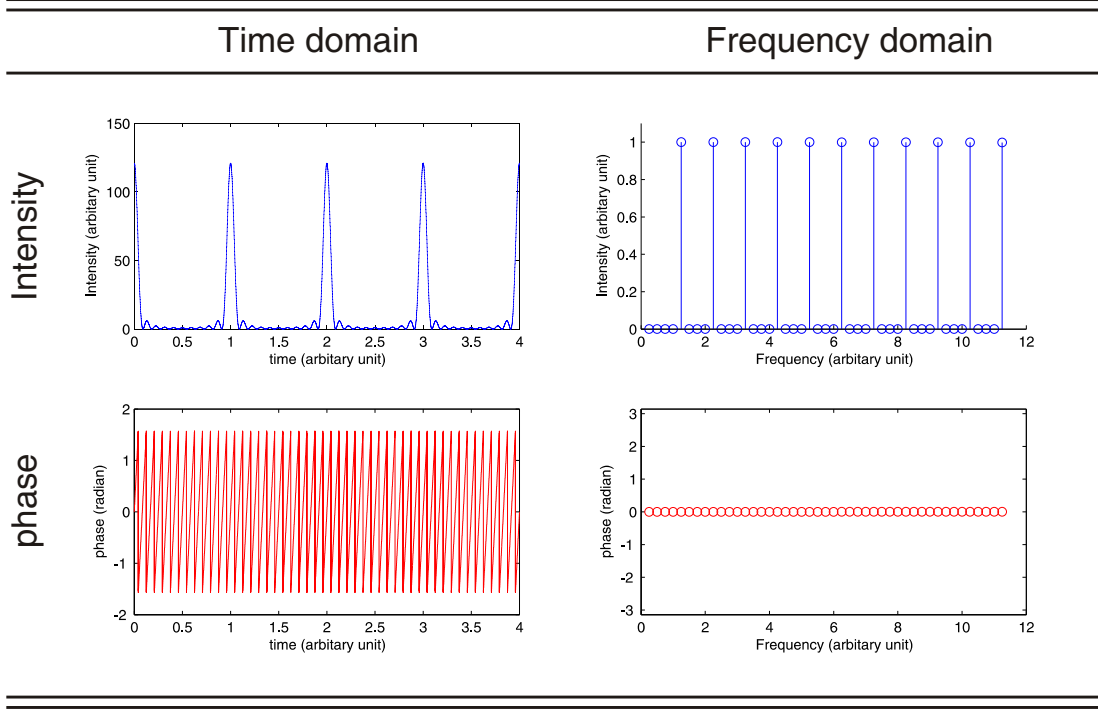


FIG. 1: Equal magnitude and equal phases

D. Equal magnitude with Gaussian distributed phases

We may simulate Gaussian shape relative phase with parameters taking values $A_q = \exp \left[-i \cdot \theta_c \frac{1}{2} \left(\frac{q}{5} \right)^2 \right]$ with $\theta_c = 0.5\pi, \pi$ and 2π . We observe the pulse duration broaden as neighbor mode relative phase getting larger (seeing FIG. 6.).

APPENDIX A: MATLAB CODE

We use a for loop to directly sum the additive pulse in time domain and adjust parameter `amp` and `phase` to simulate different cases (Gaussian distribution or constant).

```
T_f=1;
samp_time=1E-4;
t=0:samp_time:3;
E_total=zeros(size(t));
for mode=1:11,
    amp=1;
    %amp=exp(-0.5*(mode/5)^2);
    phase=-0.5*pi*(mode/5)^2;
    %phase=rand()*2*pi;
    %phase=0;
    A_q=amp*exp(j*phase);
    E_total=E_total+A_q*exp(j*mode*2*pi*t/T_f);
end
```

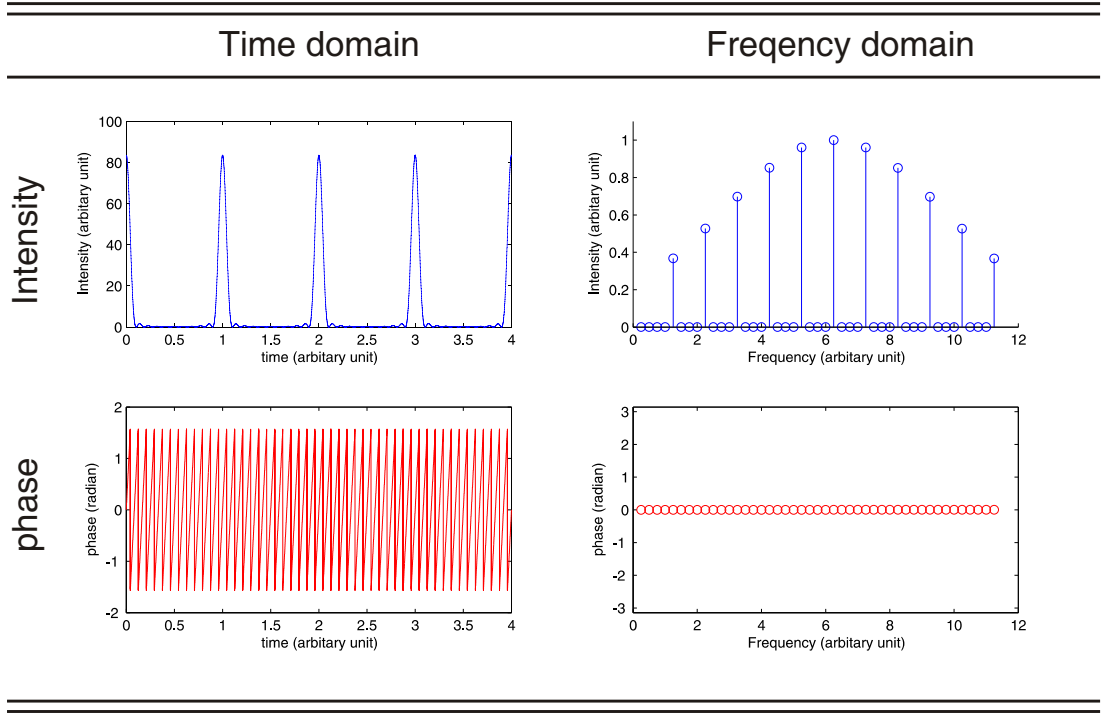


FIG. 2: Gaussian distributed magnitude with equal phases

To reconstruct the spectrum information, we use embedded `fft` function in Matlab.

```
samp=size(t,2);
Spectrum=fft(E_total,samp);
f=1/samp_time*(1:samp)/samp;
I_f=Spectrum.*conj(Spectrum)/samp;
I_fphase=atan(imag(Spectrum)./real(Spectrum));
```

Finally we plot intensity function with phase both in time and frequency domain via following codes:

```
subplot(221)
plot(t,abs(E_total).^2);
xlabel('time (arbitrary unit)');
ylabel('Intensity (arbitrary unit)')
subplot(223)
plot(t,atan(imag(E_total)./real(E_total)),'r')
xlabel('time (arbitrary unit)');
ylabel('phase (radian)')
subplot(222)
plot(f(1:cutoff_freq),I_f(1:cutoff_freq));
xlabel('Frequency (arbitrary unit)');
ylabel('Intensity (arbitrary unit)')
subplot(224)
plot(f(1:cutoff_freq),I_fphase(1:cutoff_freq),'r');
xlabel('Frequency (arbitrary unit)');
ylabel('phase (radian)')
```

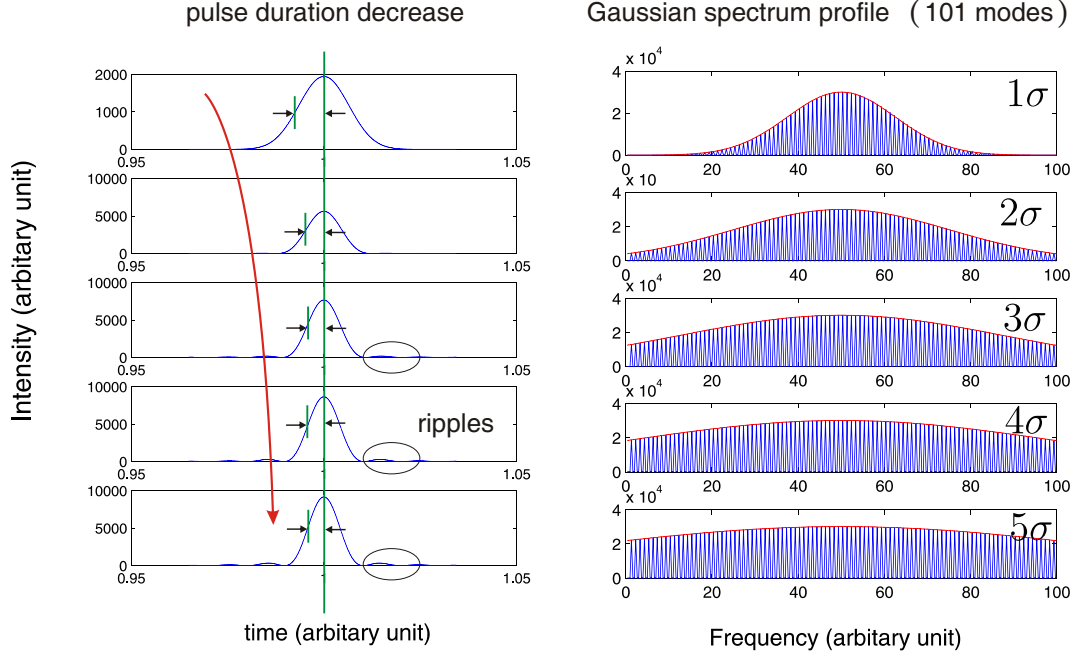
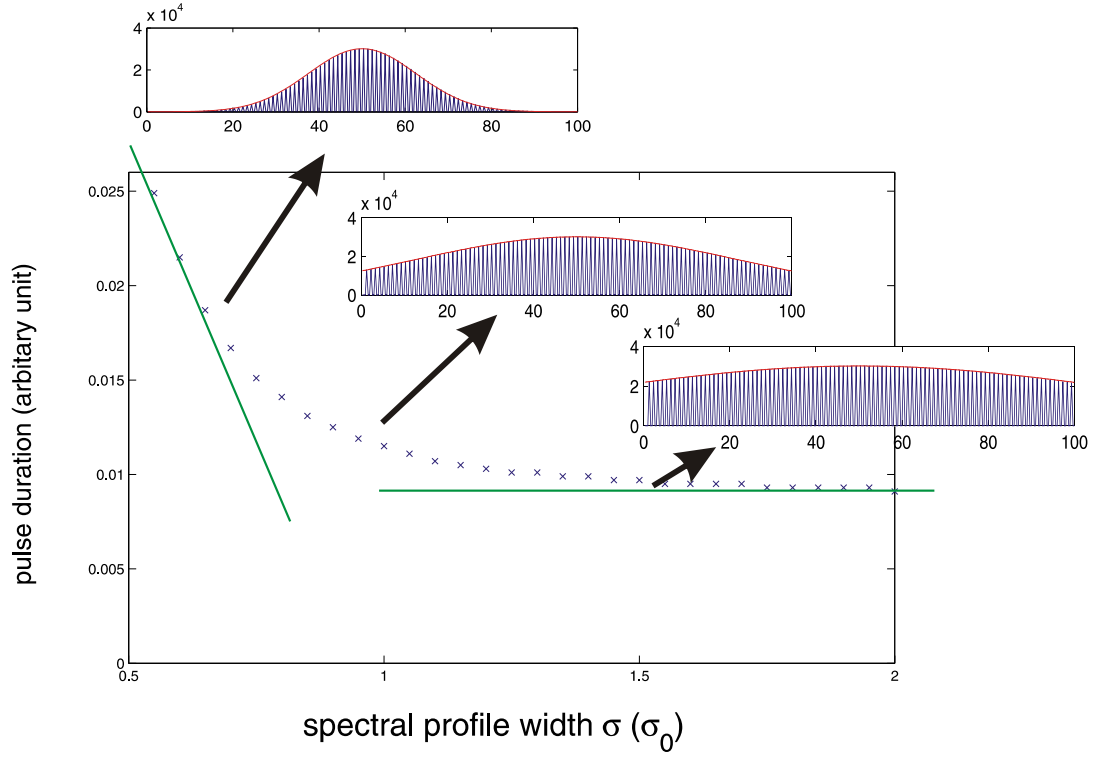


FIG. 3: pulse duration decrease with σ

To generate data like FIG. 3., we have to add some analysis to automatically recognize the FWHM of the short pulse, the modified code shows as follows,

```
T_f=1;
samp_time=1E-4;
t=0:samp_time:3;
num=5;
E_total=zeros(num,size(t,2));
for k=1:num,
    for mode=1:101,
        %amp=1;
        %amp=exp(-4*((mode-50)/(50*k))^2);
        phase=-2*pi*(mode/5)^2;
        %phase=rand()*2*pi;
        phase=0;
        A_q=amp*exp(j*phase);
        E_total(k,:)=E_total(k,:)+A_q*exp(j*mode*2*pi*t/T_f);
    end
a0=0;
It=abs(E_total(k,:)).^2;
for i=1:size(E_total(k,:),2),
    if It(1,i) > a0,
        a0=abs(E_total(k,i)).^2;
    end
end
```


FIG. 4: pulse duration v.s. σ

```

end
a1=0.5*a0;
p=1;
for i=1:size(E_total(k,:),2)-1,
    if (It(1,i)-a1)*(It(1,i+1)-a1)<0,
        a3(1,p)=t(1,i);
        a5(1,p)=i;
        p=p+1;
    end
end
a4(k)=a3(1,3)-a3(1,2);
clear a3;
samp=size(t,2);
Spectrum=fft(E_total(k,:),samp);
f=1/samp_time*(1:samp)/samp;
I_f=Spectrum.*conj(Spectrum)/samp;
I_fphase=atan(imag(Spectrum)./real(Spectrum));

```

A little modification is needed when you want to generate data like FIG. 5. We only change the parameter **amp** and neglect plotting procedure.

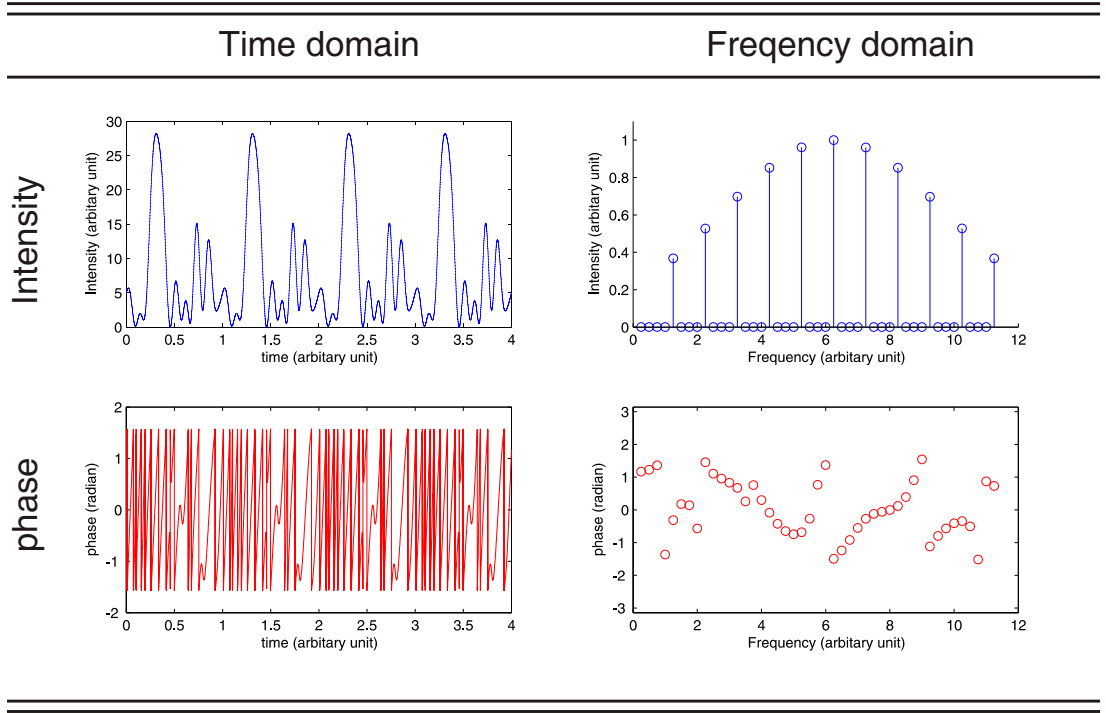


FIG. 5: equal magnitude with random phases

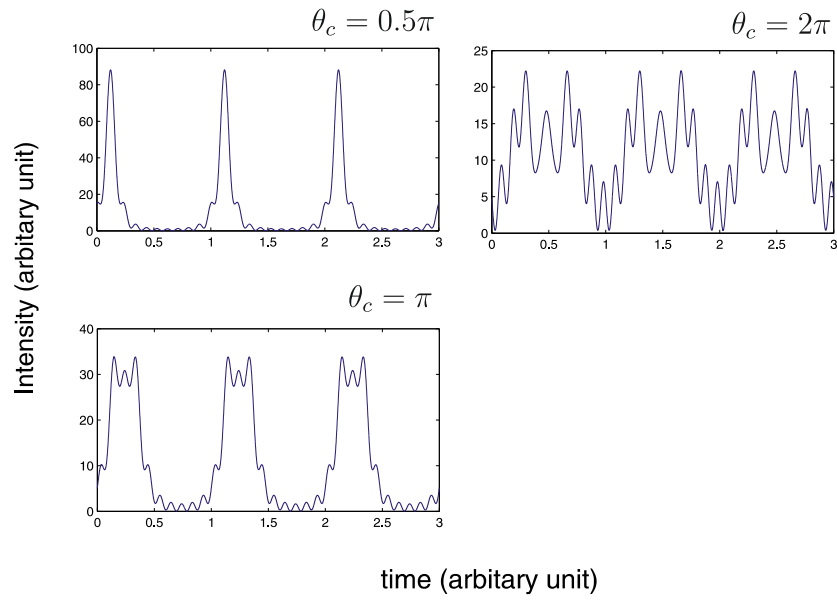


FIG. 6: equal magnitude with Gaussian distributed phases