Optically-Driven Transmissive Time-Domain Electron Phase Modulators for Attosecond Electron Bunching

February 12, 2021

1 Introduction

A general semiclassical treatement of an electron wavepacket passing through a thin, time-varying phase-modulator is developed. The phase modulation is assumed to originate from an on-axis time-varying electric field. The electron passes through some region of space where upon exit it experiences a net change in potential that oscillates at the carrier-frequency of the electric field. This can result from the ponderomotive force from two-color fields in vacuum (Kozak 2018), or from evanescent near-fields generated near material surfaces or nanometer-scale features (Priebe 2017).

2 Theoretical Treatment

A key aspect of the kind of phase modulation that we will be discussing here is that there is a net potential step as a funciton of time experienced by the electron. This can only occur for a localized field transient induced by a surface, as an electron that interacts with a freely-propagating mode cannot experience a net energy shift, only a pure time-averaged phase shift.

The treatment developed here has a few assumptions. First, it is assumed that the electron obeys the slowly varying envelope approximation as it passes through the modulator, which in essence means that the temporal dilation of the wavepacket is minimal during passage. Note, this is not assumed after the modulator, just during passage. We can express the incoming wavepacket as

$$\Psi(x,t) = A(x-vt)e^{ik_0 x}e^{-iE_0 t/\hbar} \tag{1}$$

Assuming the transmission through the modulator is 100% and that the electron moves sufficiently fast through the modulator so that it does not significantly reshape during its transit through the fields, we find that at the output port of the modulator would have a phase modulation $\varphi(t)$, and would be expressed as:

$$\psi(0,t) = A(-vt)e^{-iE_0 t/\hbar}e^{-i\varphi(t)} \tag{2}$$

Where we take the output of the modulator at $x = 0$. Given the net potential ramp from the oscillatig electric field region it passed through, we find that:

$$\frac{1}{\hbar}\varphi(t) = \int^{t} V(0,t')\mathrm{d}t' \tag{3}$$

where $V(t)$ is the net potential experienced by the electron as it passes through the modulator.

Now that the modulation is accounted for, we want to know two things: (1) what is the new energy spectrum of the electron after interaction with the modulator; and (2) how does the electron wavepacket evolve in space and time as it travels away from the modulator element.

Starting with (1), we find the energy spectrum can easily be obtained through the Fourier transform of the probability current at the output port of the modulator. The wavepacket k-space amplitude is simply expressed as:

$$a(k) = \frac{(k + k_0)}{2\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathrm{d}t e^{iEt/\hbar} \psi(0, t), \tag{4}$$

where we have used the slowly varying envelope approximation for the spatial derivative. Note that E is the outgoing energy, and is expressed as $E = k^2\hbar^2/2m$, where m is the electron mass. This is nice because it allows us to use a simple Fourier transform with respect to E.

Next, for (2) the propagation, we simply need to move each plane-wave component with amplitude $a(k)$ forward in time and reconstruct the wavepacket in space. This can be done as follows for a general treatment of $\psi(x, t)$ such that $x > 0$.

$$\psi(x > 0, t) = \int_{-\infty}^{\infty} \mathrm{d}k a(k) e^{-iE(k)t/\hbar} e^{ikx}. \tag{5}$$

This represents the entire problem that we set out to solve. In the next sections, we perform these calculations and discuss the results.

3 Notes About The Code

The code examples below use the MATLAB/Octave scripts that can be found in the following github repository: https://github.com/qnngroup/electron-phase-modulation.

In the code, the steps outlined in the theory development section are performed almost exactly, but with some extra numerical considerations. For every fourier transform, the central energy and momenta are typically removed to avoid very high-frequency oscillating terms. By taking out these central energy and momenta values, we relax sampling and memory constraints. However, the removal of these terms leads to a bit of acrobatics where you have to make sure to put it in later, and keep track of the central location in time and space of the wavepacket. Regardless, the code handles all of this for a user, so it is no problem.

4 Example: Phase Modulation, Sub-Cycle Bunching, and Demodulation

In this example, we perform an example calculation of phase modulation, attosecond bunching, and then demodulation.

4.1 Modulation

We start by setting up our initial conditions. We assume an input electron wavepacket with a time duration of 10 femtoseconds. We keep it short here to keep things quick, but this packet could be longer. The wavepacket envelope is defined by `A`.

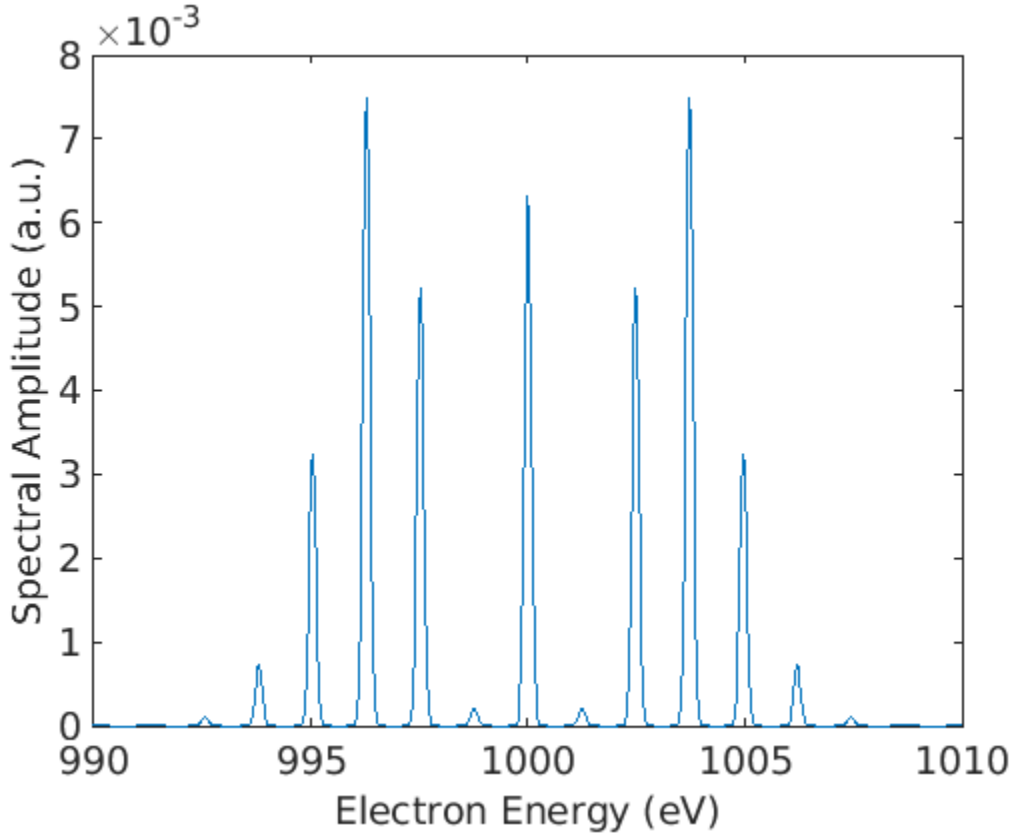
Summary of wavepacket settings: * Duration: 10 fs * Central Energy: 1 keV

[88]:
```
clear all;
addpath('../');

% Constants:
c = 299.79245; %speed of light in nm/fs

%Time axis and wavepacket pulse envelope for electron:
t = linspace(-100, 100, 2000); %Time axis in fs
W0 = 1e3; %Central energy in eV
t_electron = 10; %FWHM of electron wavepacket in fs

%Calculate the wavepacket envelope
[A, garbage] = gaussian_pulse(t, t_electron, 0, 0);
```

Next we enter the settings for our modulator. For this we assume a modulation potential having a duration of 200 femtoseconds, a central wavelength of 1000 nm, and a peak value of 5 eV. The function `calc_energy_spec` is used to calculate the outgoing energy spectrum of the modulator for the specified wavepacket envelope `A`.

Summary of modulator settings: * Duration: 200 fs * Modulation Strength: 5 eV

[90]:
```
%Modulator settings:
lambda = 1000; %Wavelength in nm
V_mag = 5; %Modulator potential strength in eV
t_V = 200; %Duration of modulation function (FWHM in fs)

%Calculate the central frequency of the modulator and its function
omega = 2*pi*c/lambda;
[V_env, garbage] = gaussian_pulse(t, t_V, omega, 0); %Potential function
V = V_mag*V_env; %Construct the actual potential profile

%Calculate and plot the energy spectrum
[k, a_k, W, P_W] = calc_energy_spec(t, A, W0, V);

figure(1);
plot(W, P_W);
set(gca, 'fontsize', 14);
xlabel('Electron Energy (eV)', 'fontsize', 14);
ylabel('Spectral Amplitude (a.u.)', 'fontsize', 14);
xlim([990, 1010])
```



Note that the central energy is maintained, but sidebands have been formed at multiples of the modulation photon energy due to the time-domain phase modulation. This phase modulation also imparts regions of quadratic phase that lead to "temporal focusing" that we will explore in the next section.

4.2 Sub-Cycle Bunching

Due to the phase modulation imparted by passing through the modulator element, there are regions of quadratic phase temporally that lead to sub-optical-cycle bunching of the electron wavepacket as it propagates. Just like a spatial lens, this focusing effect leads to bunching at a region in space after the modulator element, after which the electron packet starts to break apart again.

We can propagate the wavepacket forward in space to see this as described earlier. There are two functions that do this based on the output of `calc_energy_spec()`.

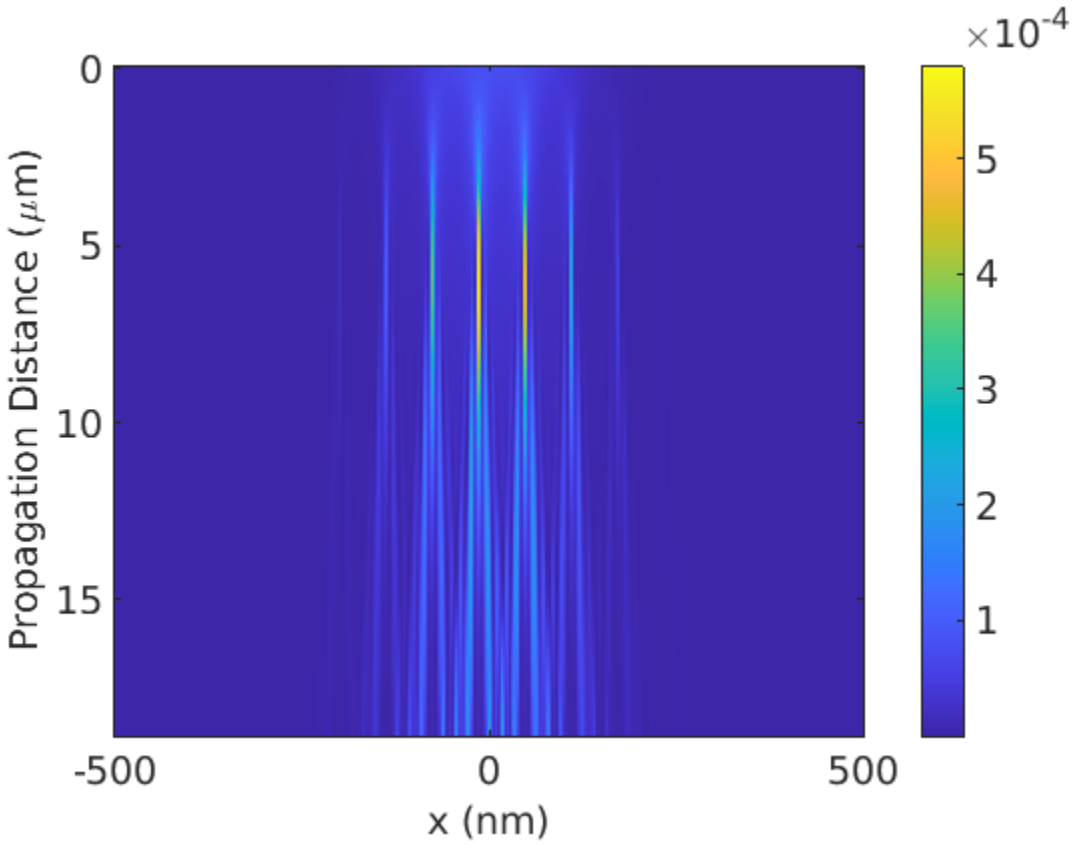
Propagation Functions: * `propagate_fixed_time()` – Propagates the electron forward for a fixed amount of time and returns the wavepacket as a function of space. * `propagate_fixed_space()` – Outputs the electron wavepacket as a function of time as it passes a fixed region of space.

We can start by using `propagate_fixed_time()` to see how the wavepacket evolves spatially as it moves. Note that these functions actually only calculate the wavepacket over a narrow region in space near its center, and keep track of the center location. This is essentially staying in the moving frame of the electron.

In the pseudocolor plot below we show the electron in the moving frame as a function of its central position.

5 eV Modulation

[91]:
```
%Now propagate to t_prime and output the real-space wavefunction:
t_prime = linspace(0, 1000, 100);

%Propagate...
[x_center, x, u_out] = propagate_fixed_time(t_prime, W0, k, a_k);

%Plot output wavefunction u_out:
figure(2);
imagesc(x, x_center/1000, abs(u_out).^2);
set(gca, 'fontsize', 14)
xlim([-500, 500]);
shading interp;
colorbar('fontsize', 14);
xlabel('x (nm)', 'fontsize', 14);
ylabel('Propagation Distance (\mum)', 'fontsize', 14);
```

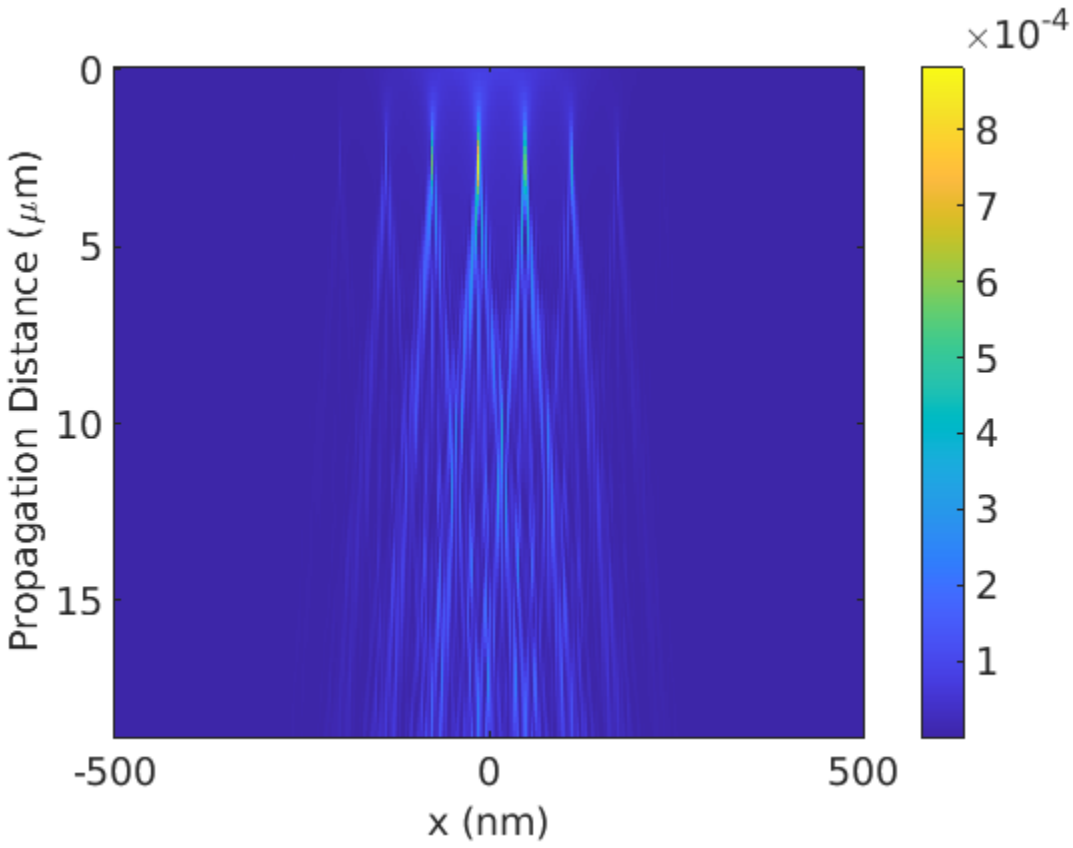


As you can see, the wavepacket comes to a "temporal focus" roughly 5 micrometers away from the

modulator plate. By increasing the modulator strength we can move the temporal focus to a closer region in space. By decreasing it, to a further region in space.

10 eV Modulation

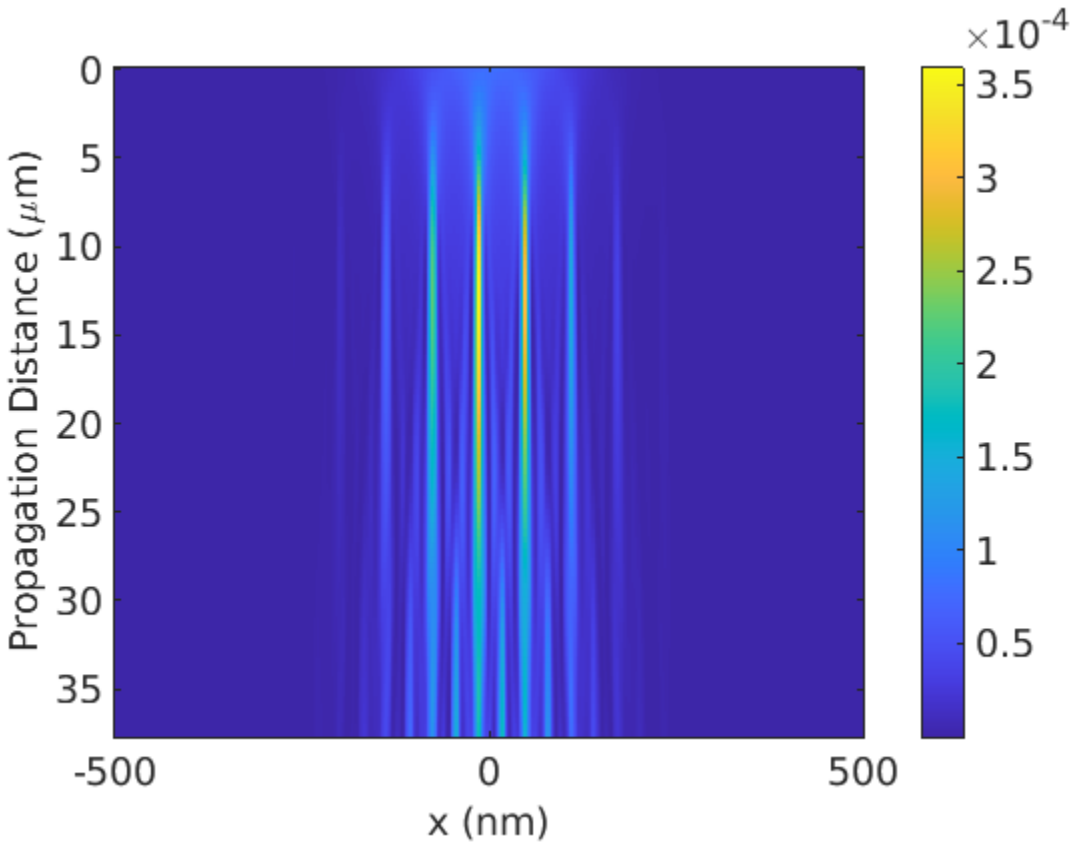
[92]:
```
V_mag = 10;
V = V_mag*V_env; %Construct the actual potential profile

[k, a_k, W, P_W] = calc_energy_spec(t, A, W0, V);


%Now propagate to t_prime and output the real-space wavefunction:
t_prime = linspace(0, 1000, 100);

%Propagate...
[x_center, x, u_out] = propagate_fixed_time(t_prime, W0, k, a_k);

%Plot output wavefunction u_out:
figure(2);
imagesc(x, x_center/1000, abs(u_out).^2);
set(gca, 'fontsize', 14)
xlim([-500, 500]);
shading interp;
colorbar('fontsize', 14);
xlabel('x (nm)', 'fontsize', 14);
ylabel('Propagation Distance (\mum)', 'fontsize', 14);
```



2.5 eV Modulation

[93]:
```
V_mag = 2.5;
V = V_mag*V_env; %Construct the actual potential profile

[k, a_k, W, P_W] = calc_energy_spec(t, A, W0, V);


%Now propagate to t_prime and output the real-space wavefunction:
t_prime = linspace(0, 2000, 100);

%Propagate...
[x_center, x, u_out] = propagate_fixed_time(t_prime, W0, k, a_k);

%Plot output wavefunction u_out:
figure(2);
imagesc(x, x_center/1000, abs(u_out).^2);
set(gca, 'fontsize', 14)
xlim([-500, 500]);
shading interp;
colorbar('fontsize', 14);
```

```
xlabel('x (nm)', 'fontsize', 14);
ylabel('Propagation Distance (\mum)', 'fontsize', 14);
```



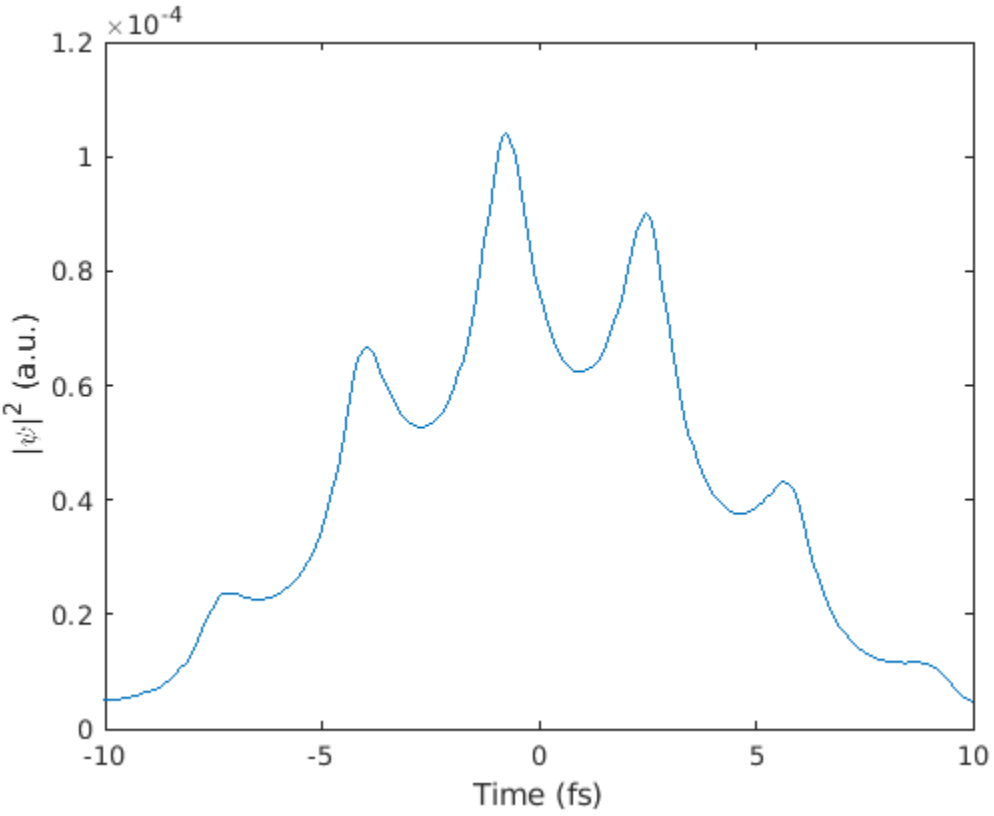
[95]:
```
%Reset back to 5 eV modulation
V_mag = 5;
V = V_mag*V_env; %Construct the actual potential profile

[k, a_k, W, P_W] = calc_energy_spec(t, A, W0, V);
```

We can also look at the temporal structure of the wavepacket at a fixed regions in space.

1 micron propagation

[96]:
```
% -- What about de-modulation?
physical_constants_normalized;

k0 = sqrt(2*W0);
x_center = 1000;
%t_prime = 1000;
%x_center = t_prime*k0*x0/t0;
```

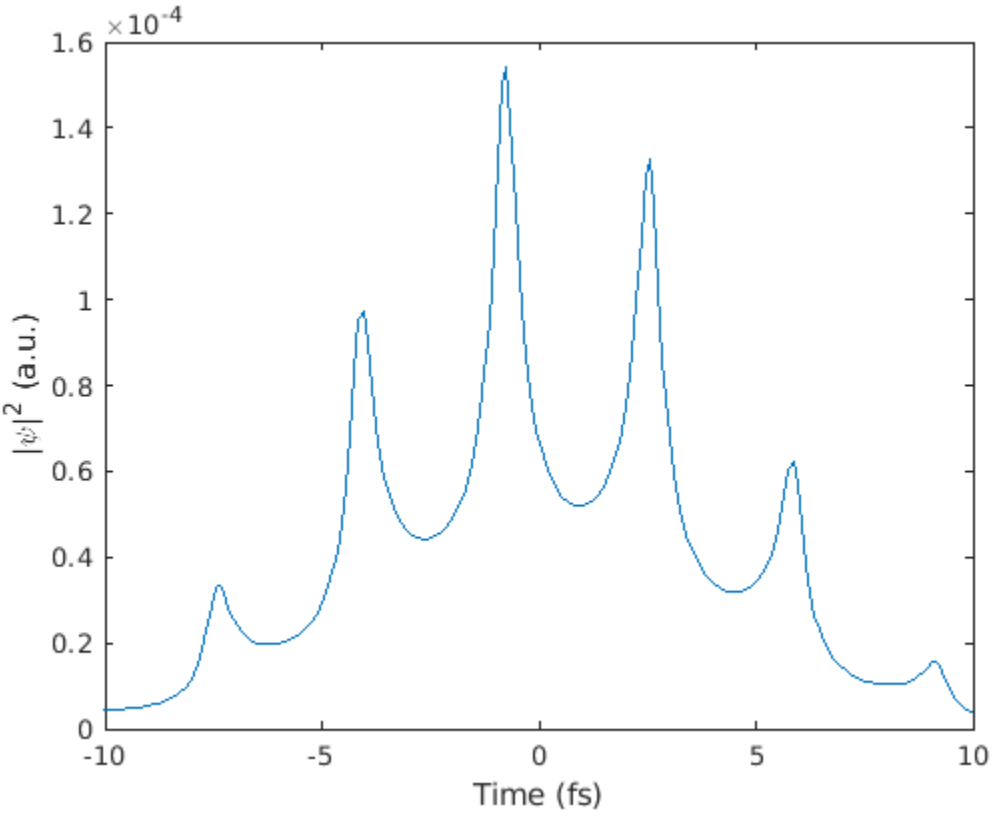
```
%t = linspace(-100, 100, 10000);


[t_center, u_out] = propagate_fixed_space(x_center, t, W0, k, a_k);

figure();
plot(t, abs(u_out).^2);
xlim([-10, 10]);
xlabel('Time (fs)');
ylabel('|\psi|^2 (a.u.)');
```



2 micron propagation

[97]:
```
% -- What about de-modulation?
physical_constants_normalized;

k0 = sqrt(2*W0);
x_center = 2000;
%t_prime = 1000;
%x_center = t_prime*k0*x0/t0;
```

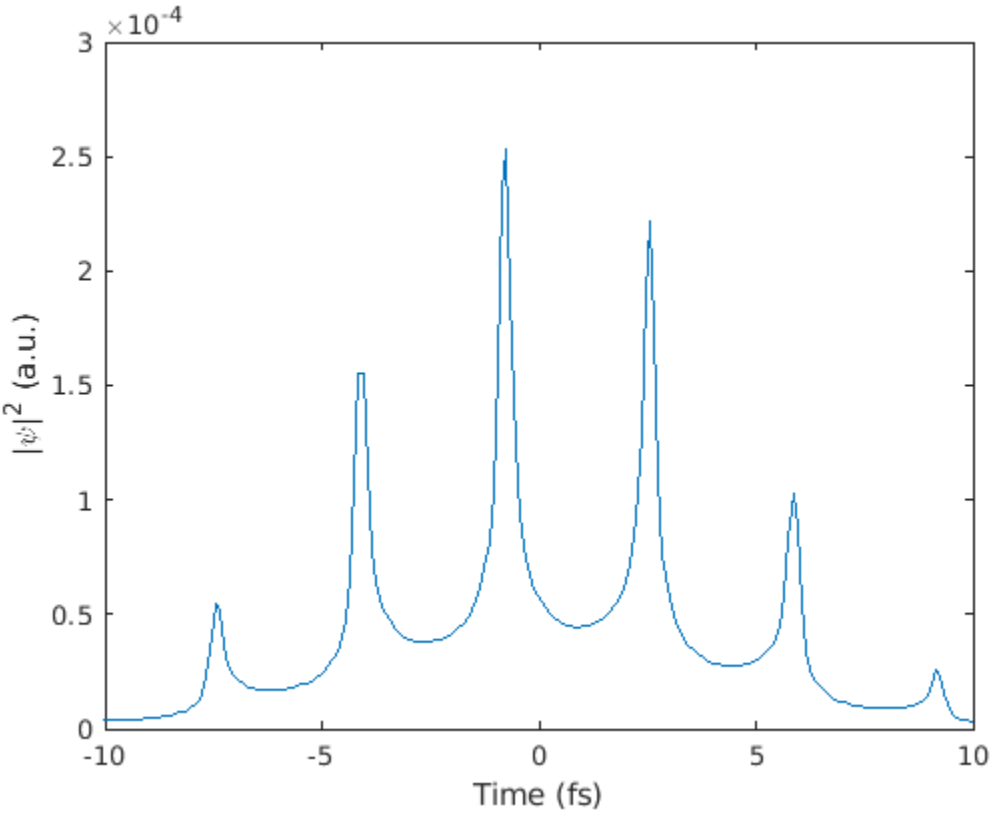
```
%t = linspace(-100, 100, 10000);


[t_center, u_out] = propagate_fixed_space(x_center, t, W0, k, a_k);

figure();
plot(t, abs(u_out).^2);
xlim([-10, 10]);
xlabel('Time (fs)');
ylabel('|\psi|^2 (a.u.)');
```



3 micron propagation

[98]:
```
% -- What about de-modulation?
physical_constants_normalized;

k0 = sqrt(2*W0);
x_center = 3000;
%t_prime = 1000;
%x_center = t_prime*k0*x0/t0;
```

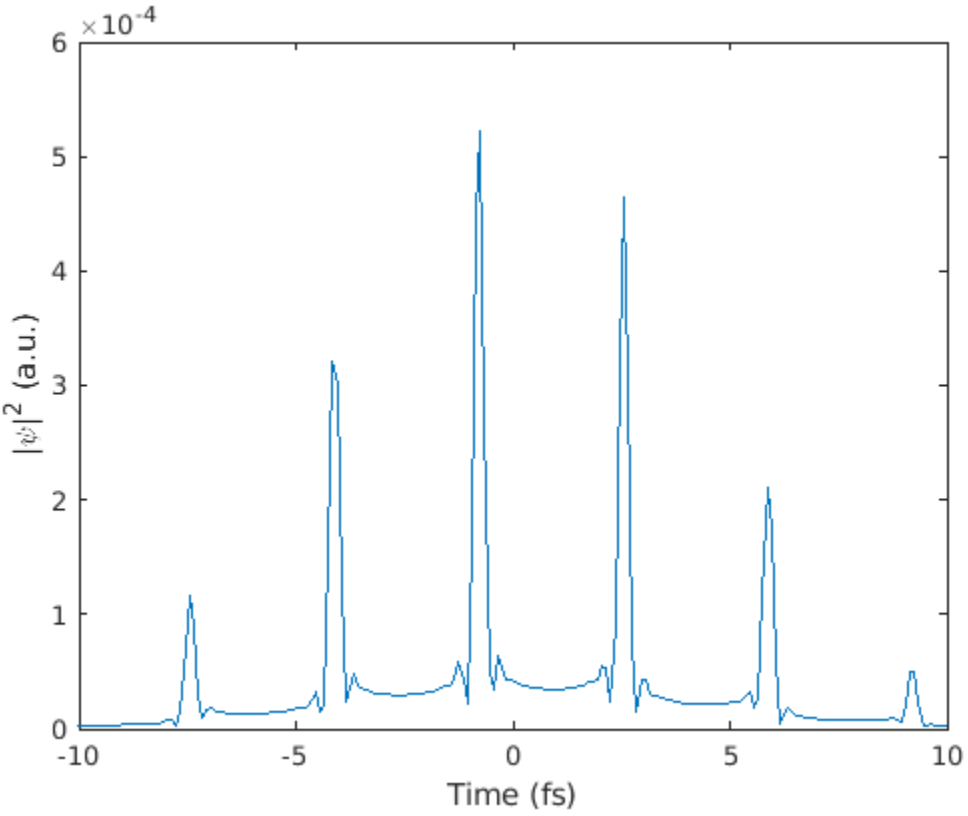
```
%t = linspace(-100, 100, 10000);


[t_center, u_out] = propagate_fixed_space(x_center, t, W0, k, a_k);

figure();
plot(t, abs(u_out).^2);
xlim([-10, 10]);
xlabel('Time (fs)');
ylabel('|\psi|^2 (a.u.)');
```



5 micron propagation

[99]:
```
% -- What about de-modulation?
physical_constants_normalized;

k0 = sqrt(2*W0);
x_center = 5000;
%t_prime = 1000;
%x_center = t_prime*k0*x0/t0;
```

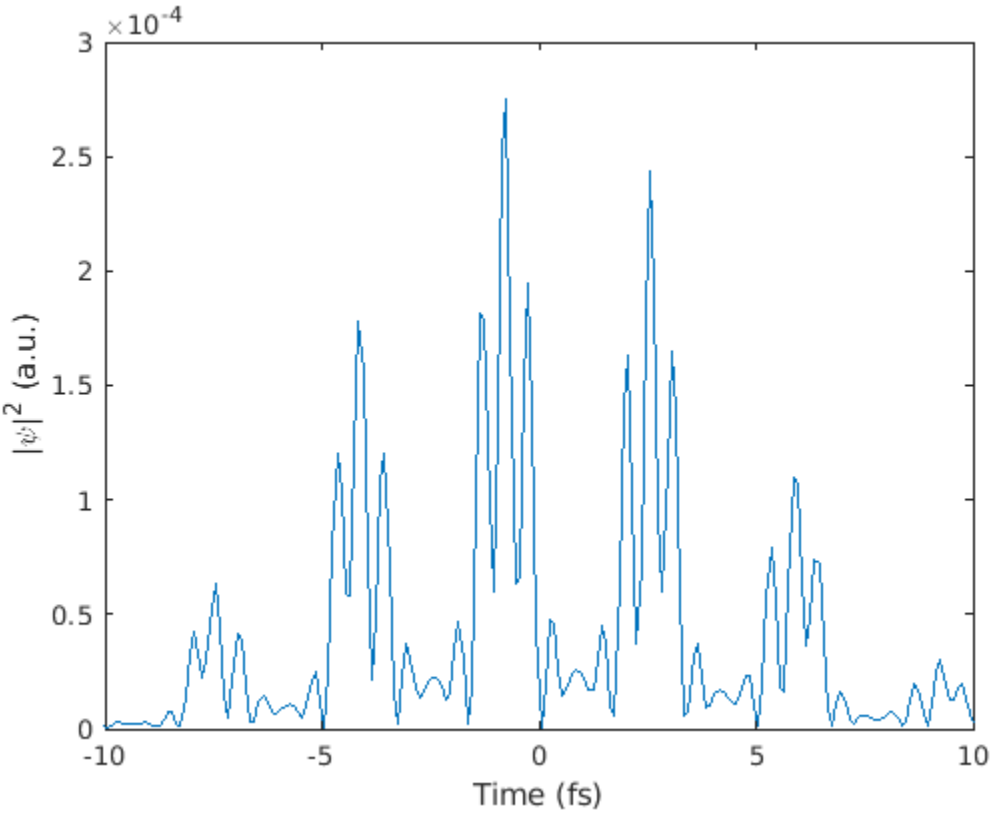
```
%t = linspace(-100, 100, 10000);


[t_center, u_out] = propagate_fixed_space(x_center, t, W0, k, a_k);

figure();
plot(t, abs(u_out).^2);
xlim([-10, 10]);
xlabel('Time (fs)');
ylabel('|\psi|^2 (a.u.)');
```



10 micron propagation

[100]:
```
% -- What about de-modulation?
physical_constants_normalized;

k0 = sqrt(2*W0);
x_center = 10000;
%t_prime = 1000;
%x_center = t_prime*k0*x0/t0;
```

```
%t = linspace(-100, 100, 10000);


[t_center, u_out] = propagate_fixed_space(x_center, t, W0, k, a_k);

figure();
plot(t, abs(u_out).^2);
xlim([-10, 10]);
xlabel('Time (fs)');
ylabel('|\psi|^2 (a.u.)');
```



4.3 Demodulation

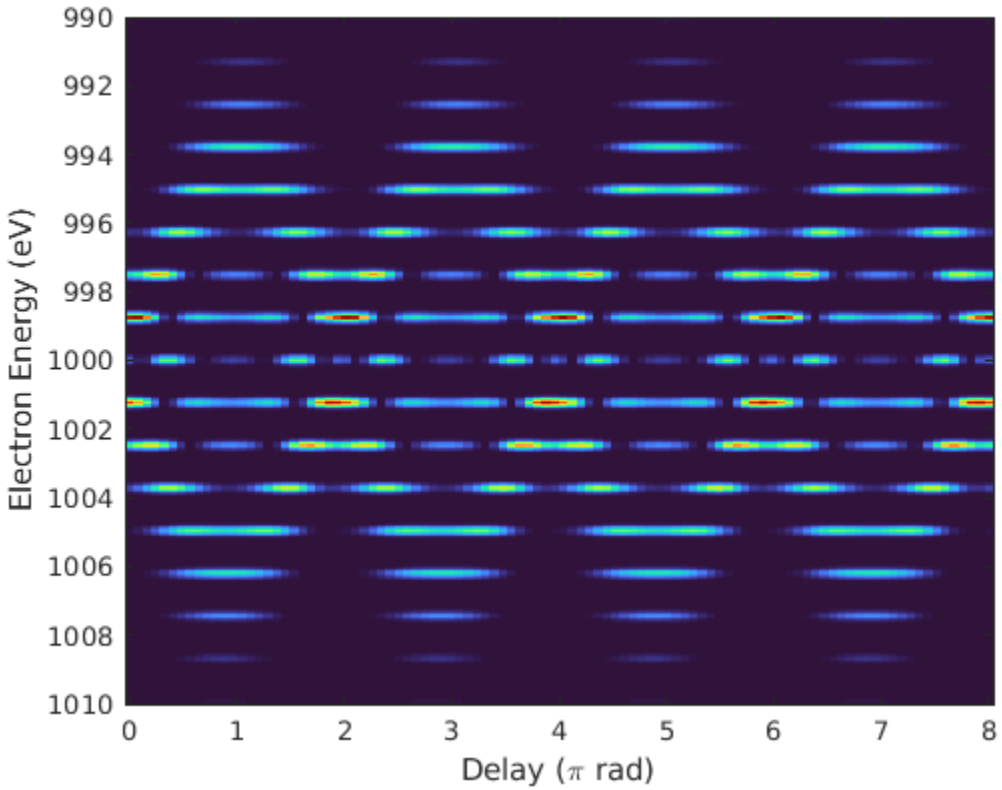
By placing another modulator at a fixed distance from the first, we can demodulate the incoming electron beam. This demodulation leads to an interference in the different energy regions of the incoming wavepacket. By changing the delay between the modulating and demodulating pulse, we can essentially drive a time-domain interferometer with variable phase delay. By examining the phase delay of each energy peak as a function of the modulation delay, you can discern the relative phase of each energy component. This then allows you to retrieve the periodic pulse train profile based on a Fourier series analysis. This is highly analagous to the RABBITT technique used for attosecond pulse train reconstruction (Muller 2002), and similar techniques have been adapted for

attosecond free-electron wavepackets (Priebe 2017).

Lets start by demodulating the wavepackets at 1 micron, 5 micron, and 10 micron spacings.

1 micron spacing

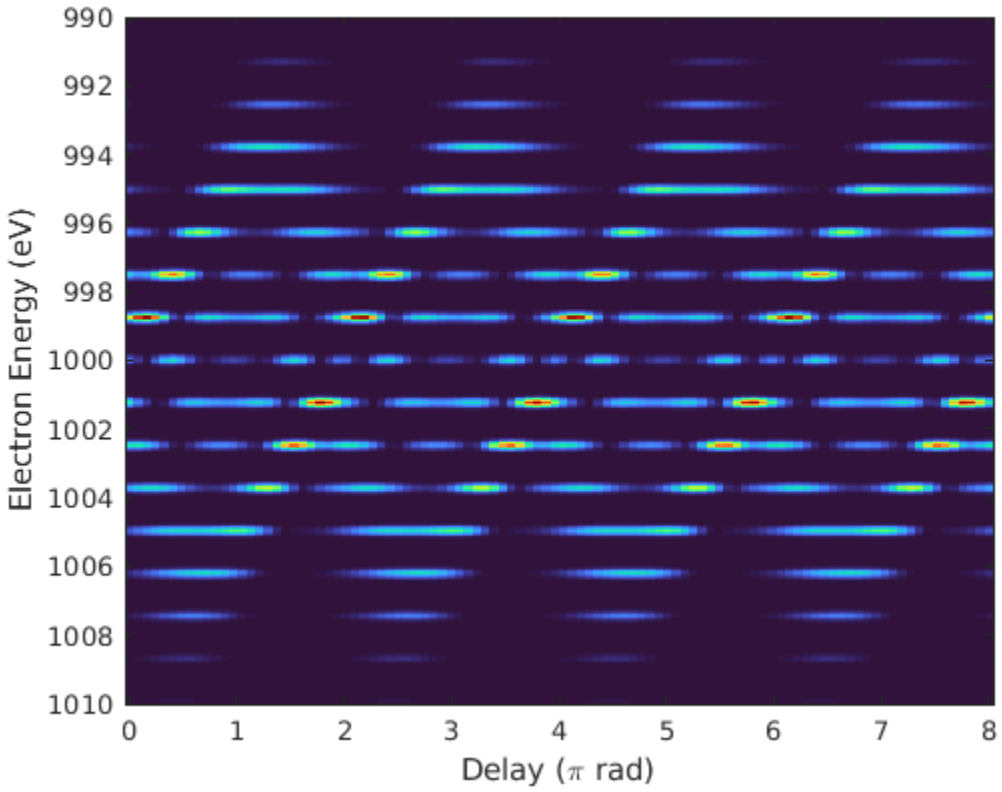
[101]:
```
%Demodulator Settings
V_mag_dm = 2.5;
phase_dm = linspace(0, 8*pi, 100);

x_center = 1000;

[t_center, u_out] = propagate_fixed_space(x_center, t, W0, k, a_k);


for a = 1:length(phase_dm)

  [V_env_dm, garbage] = gaussian_pulse(t, t_V, omega, phase_dm(a)); %Potential␣
↪function
  V_dm = V_mag_dm*V_env_dm; %Construct the actual potential profile

  [k_dm, a_k_dm, W_dm, P_W_dm(a, :)] = calc_energy_spec(t, u_out, W0, V_dm);

end

figure(3);
imagesc(phase_dm/pi, W_dm, P_W_dm.');
ylim([990, 1010]);
ylabel('Electron Energy (eV)');
xlabel('Delay (\pi rad)');
colormap turbo
```



5 micron spacing

[102]:
```
%Demodulator Settings
V_mag_dm = 2.5;
phase_dm = linspace(0, 8*pi, 100);

x_center = 5000;

[t_center, u_out] = propagate_fixed_space(x_center, t, W0, k, a_k);


for a = 1:length(phase_dm)

  [V_env_dm, garbage] = gaussian_pulse(t, t_V, omega, phase_dm(a)); %Potential␣
↪function
  V_dm = V_mag_dm*V_env_dm; %Construct the actual potential profile

  [k_dm, a_k_dm, W_dm, P_W_dm(a, :)] = calc_energy_spec(t, u_out, W0, V_dm);

end
```

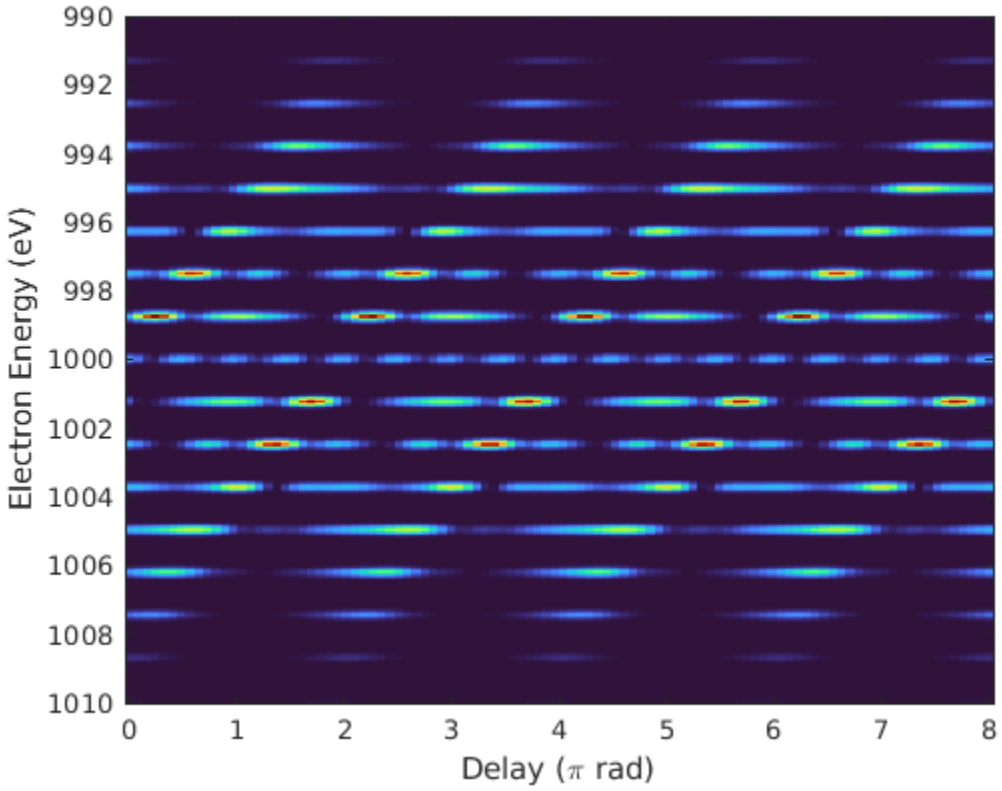
```
figure(3);
imagesc(phase_dm/pi, W_dm, P_W_dm.');
ylim([990, 1010]);
ylabel('Electron Energy (eV)');
xlabel('Delay (\pi rad)');
colormap turbo
```



10 micron spacing

[103]:
```
%Demodulator Settings
V_mag_dm = 2.5;
phase_dm = linspace(0, 8*pi, 100);

x_center = 10000;

[t_center, u_out] = propagate_fixed_space(x_center, t, W0, k, a_k);


for a = 1:length(phase_dm)
```

```
  [V_env_dm, garbage] = gaussian_pulse(t, t_V, omega, phase_dm(a)); %Potential␣
↪function
  V_dm = V_mag_dm*V_env_dm; %Construct the actual potential profile

  [k_dm, a_k_dm, W_dm, P_W_dm(a, :)] = calc_energy_spec(t, u_out, W0, V_dm);

end

figure(3);
imagesc(phase_dm/pi, W_dm, P_W_dm.');
ylim([990, 1010]);
ylabel('Electron Energy (eV)');
xlabel('Delay (\pi rad)');
colormap turbo
```



Note that for each case, a unique spectrogram is created which relates to the precise temporal structure of the wavepacket at the demodulator. Likewise, by understanding the wavepacket at the demodulator, we can reverse propagate it and discern its evolution in space and time. This will give unprecedented detail in understanding the temporal structure and evolution of electrons as they pass over or through media between the modulator/demodulator stack.

5 References

(Kozak 2018) Kozák, M., N. Schönenberger, and P. Hommelhoff. 2018. "Ponderomotive Generation and Detection of Attosecond Free-Electron Pulse Trains." Physical Review Letters 120 (10): 103203. https://doi.org/10.1103/PhysRevLett.120.103203.

(Muller 2002) Muller, H.G. 2002. "Reconstruction of Attosecond Harmonic Beating by Interference of Two-Photon Transitions." Applied Physics B 74 (1): s17–21. https://doi.org/10.1007/s00340-002-0894-8.

(Priebe 2017) Priebe, Katharina E., Christopher Rathje, Sergey V. Yalunin, Thorsten Hohage, Armin Feist, Sascha Schäfer, and Claus Ropers. 2017. "Attosecond Electron Pulse Trains and Quantum State Reconstruction in Ultrafast Transmission Electron Microscopy." Nature Photonics 11 (12): 793–97. https://doi.org/10.1038/s41566-017-0045-8.