

July 2021

دانشگاه تربیت مدرس



. In the name of GOD .

Neuroscience
project_part1_final

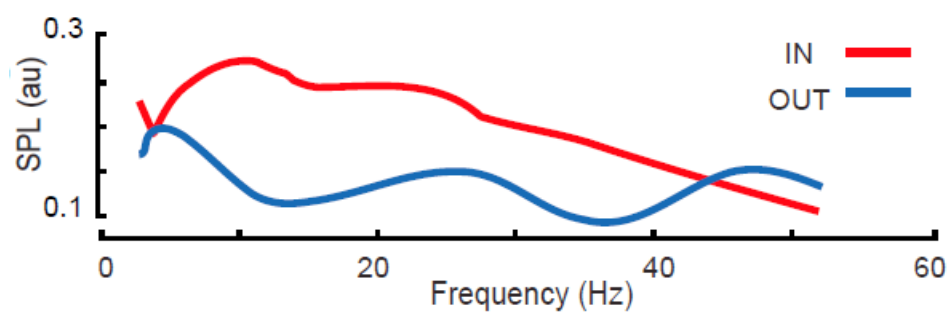
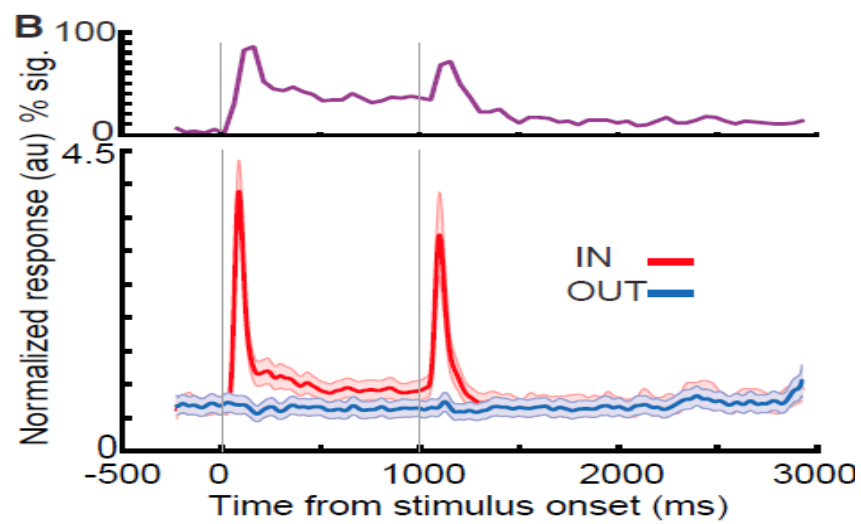
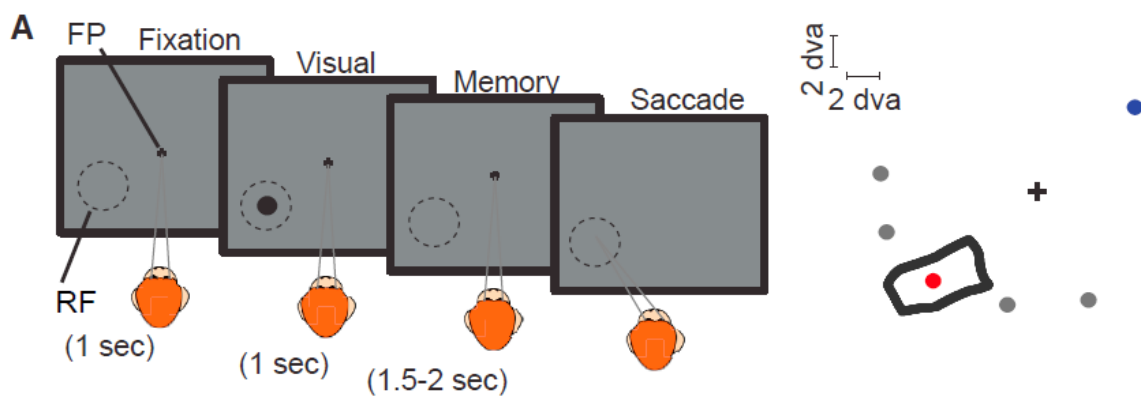
Presented to
Dr. Bahmani

Submitted by
Poorya Aghaomidi
9961391001



1.2. File name :

1.1. Figures :



1.3. Explanation :

هدف از این task شبیه سازی عملکرد نوروها در ناحیه MT می باشد. در ابتدا 1 ثانیه میمون به یک نقطه fixation می کند و در 1 ثانیه استیمولوس برای میمون نمایش داده می شود و به مدت 1 ثانیه بر روی صفحه باقی می ماند و پس از آن از روی صفحه محو می شود و بازه working memory برای به خاطر سپردن مکان استیمولوس آغاز می شود. یعنی 2-3.5 ثانیه.

همچنین نرونی که از آن ثبت انجام شده است نسبت به یک مکان خاص حساس است و به آن بیشترین پاسخ را نشان می دهد. و هرچه از آن استیمولوس دور تر باشد پاسخ ضعیف تر خواهد شد.

در این شکل نیز می توان دید که در ثانیه 0 و 1000 میلی ثانیه پس از رفتن fixation point و آمدن استیمولوس بر روی صفحه تغییرات firing rate وجود دارد، که اولی به دلیل آمدن استیمولوس است و دومی به دلیل رفتن استیمولوس بوجود آمده است. همچنین adaptation نیز در این بازه اتفاق افتاده است.

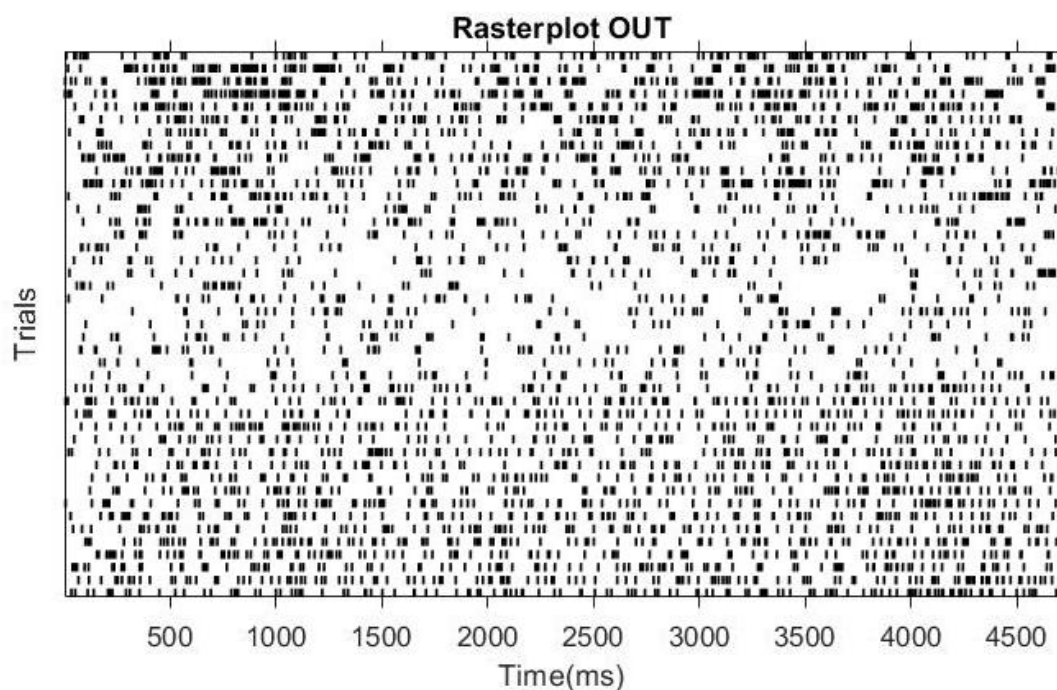
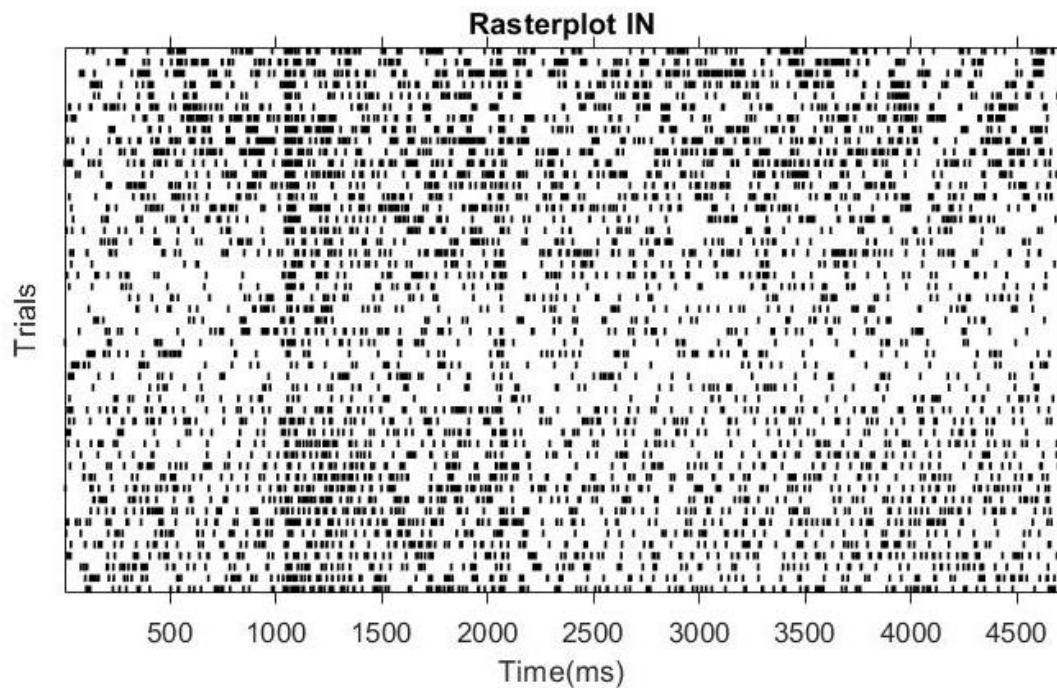
در مورد داده می توان گفت که از 3 قسمت lfp و spike و event تشکیل شده که اطلاعات در مورد نحوه ثبت و انجام گرفتن task در event قرار دارد. در این برنامه از ردیف اول و بیشتر از نورو 4 استفاده شده. کانال 4 و سایر کانال ها هر یک در خود چندین ماتریس دارند که برای task های متفاوتی هستند که اکثرا از بخش mgs استفاده شد. همچنین شرایط و مکان ها در ستون 5 ام ماتریس event قرار دارند.

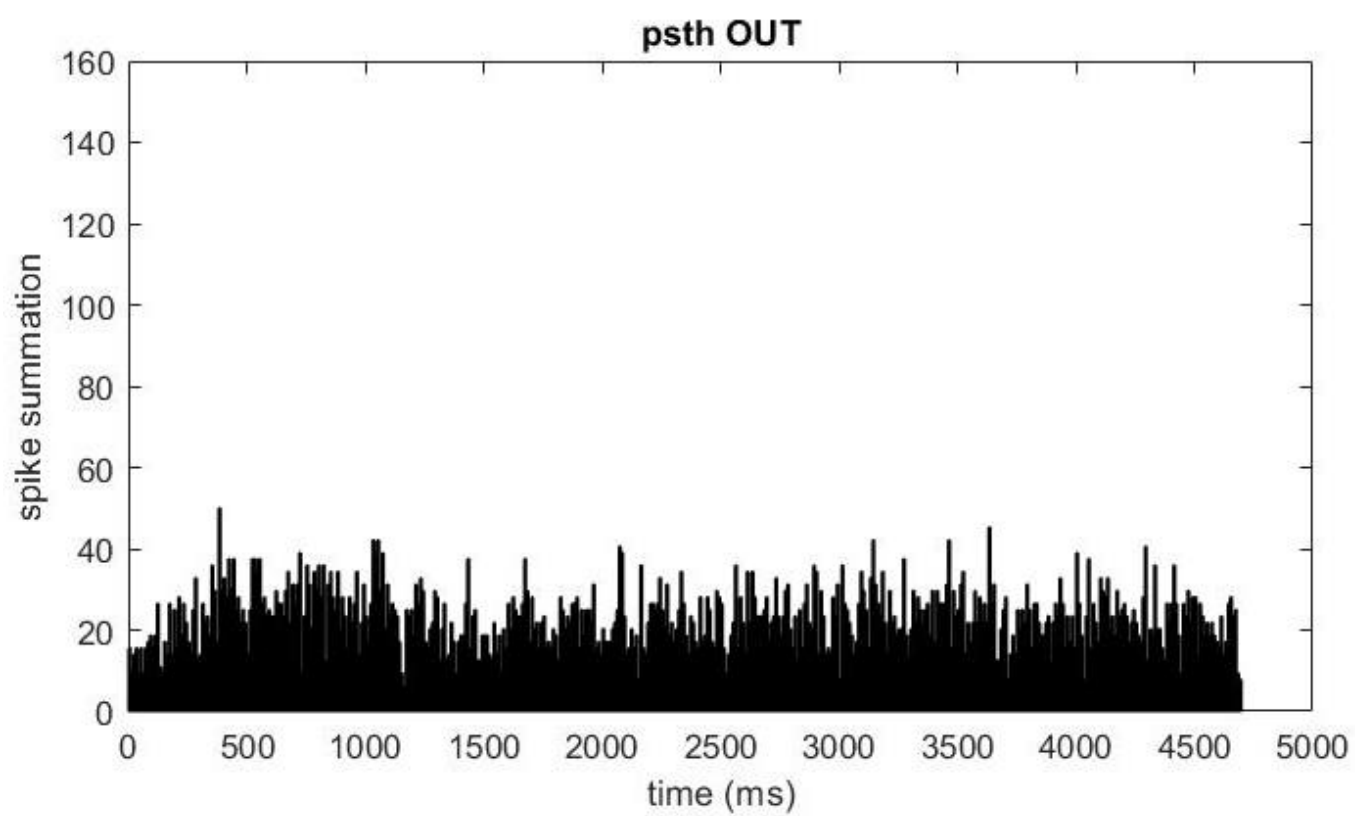
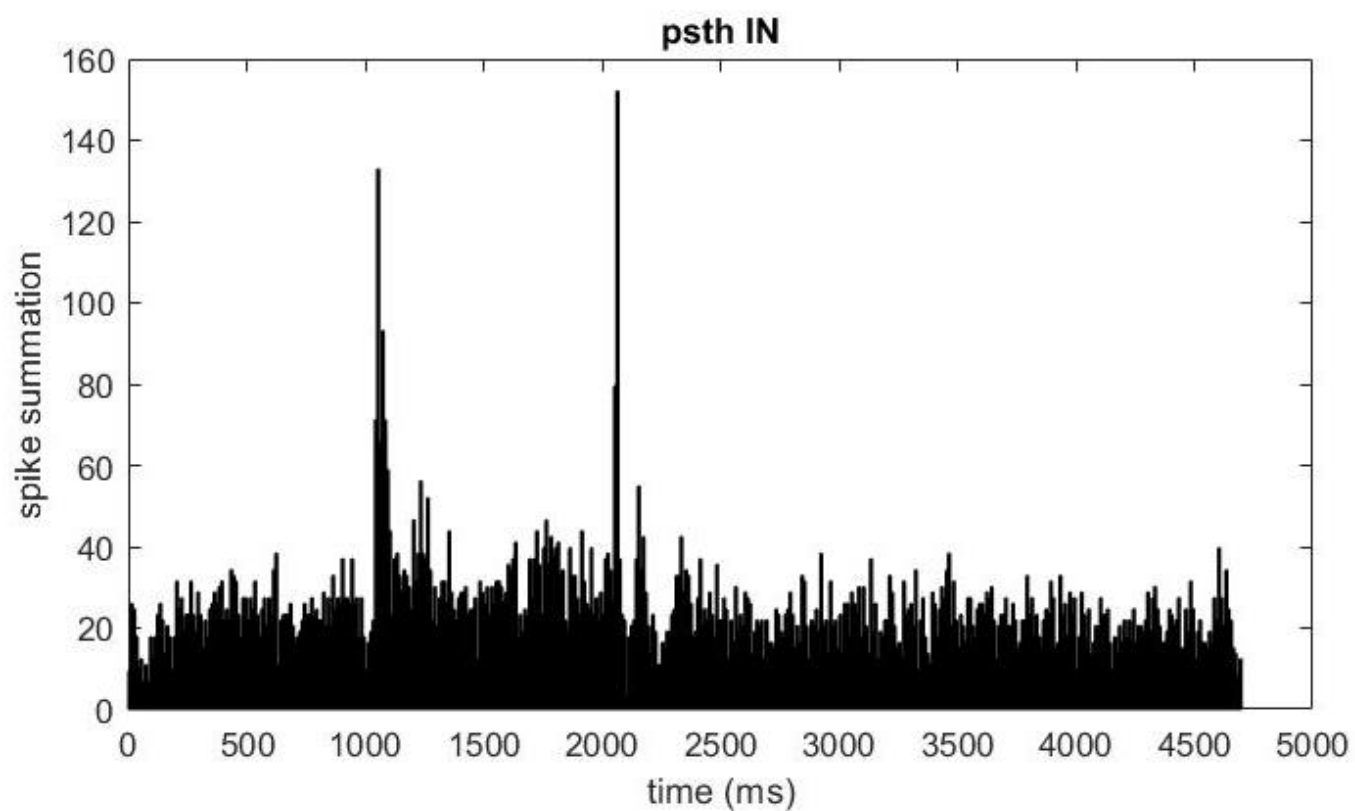
در سوال آخر از دو تسک vodor و vabor برای رسم RF و tuning curve استفاده شد. در این تمرین زمان بر حسب ms و فرکانس بر حسب sample/ms نوشته شده اند. تمرکز در این تمرین بر روی به دست آوردن و کار بر روی فرکانس است.

2.1.1. File name :

2.a

2.1.2. Figures :





2.1.3. Explanation :

در این سوال هدف محاسبه Raster plot و PSTH است. در ابتدا از روی داده Events شماره تریال های مربوط به مکان ها را پیدا می کنیم. حال از روی داده اسپایک که درکل روی 422 تریال انجام شده، اسپایک ها را مشاهده می کنیم. حال برای هر یک از مکان ها باید ببینیم کدام تریال مربوط به کدام مکان است و اسپایک همه ی مکان ها را از هم جدا کنیم.

موقعیت 51 را in در نظر می گیریم و 54 را out که دورترین فاصله را از receptive field دارد. فقط باید توجه داشته باشیم که از کانال 4 اسپایک استفاده کردیم.

اگر به حوالی 1000 و حوالی 2000 در raster plot دقت کنیم می توانیم اثر آمدن و رفتن استیمولوس را با یک تاخیری در این زمان ها در مکان in در مقایسه با out ببینیم که این تاخیر به دلیل فاصله زمانی رسیدن اطلاعات از چشم به MT است. همچنین در نمودار psth این اتفاق را می توان به وضوح دید و نتیجه گرفت که ناحیه MT همان طور که انتظار می رفت نسبت به تغییرات حسی واکنش نشان می دهد و به این را با firing rate کد می کند و این نکته را نیز باید در نظر گرفت که کمی بعد از آمدن استیمولوس adaptation اتفاق افتاده است. و نیز این کدینگ فقط بیشتر برای مکانی اتفاق می افتد که نوروں نسبت به آن حساس است و هر چه از آن مکان دورتر شویم پاسخ ضعیف تر خواهد بود.

2.1.4. Codes :

```
%% initialize
locations = unique(Event.mgs.codes(:,5));
neuron = spike{4, 1} ;

L51 = find(Event.mgs.codes(:,5) == locations(1));
L52 = find(Event.mgs.codes(:,5) == locations(2));
L53 = find(Event.mgs.codes(:,5) == locations(3));
L54 = find(Event.mgs.codes(:,5) == locations(4));
L55 = find(Event.mgs.codes(:,5) == locations(5));
L56 = find(Event.mgs.codes(:,5) == locations(6));

for i=1:numel(L51)
    spike_51(i,:) = neuron.mgs(L51(i),:);
end

for i=1:numel(L52)
    spike_52(i,:) = neuron.mgs(L52(i),:);
end

for i=1:numel(L53)
    spike_53(i,:) = neuron.mgs(L53(i),:);
end

for i=1:numel(L54)
    spike_54(i,:) = neuron.mgs(L54(i),:);
end

for i=1:numel(L55)
    spike_54(i,:) = neuron.mgs(L55(i),:);
end

for i=1:numel(L56)
    spike_55(i,:) = neuron.mgs(L56(i),:);
end

%% plot
Q = unique(spike_51);
spike_51 = spike_51';
spike_54 = spike_54';
W= unique(spike_54);

yek_in  = find(spike_51 == Q(2));
yek_out = find(spike_54 == W(2));

% raster plot
figure ;
subplot(211)
myrasterplot(yek_in,73,4701)
title('Rasterplot IN')
subplot(212)
myrasterplot(yek_out,64,4701)
title('Rasterplot OUT')

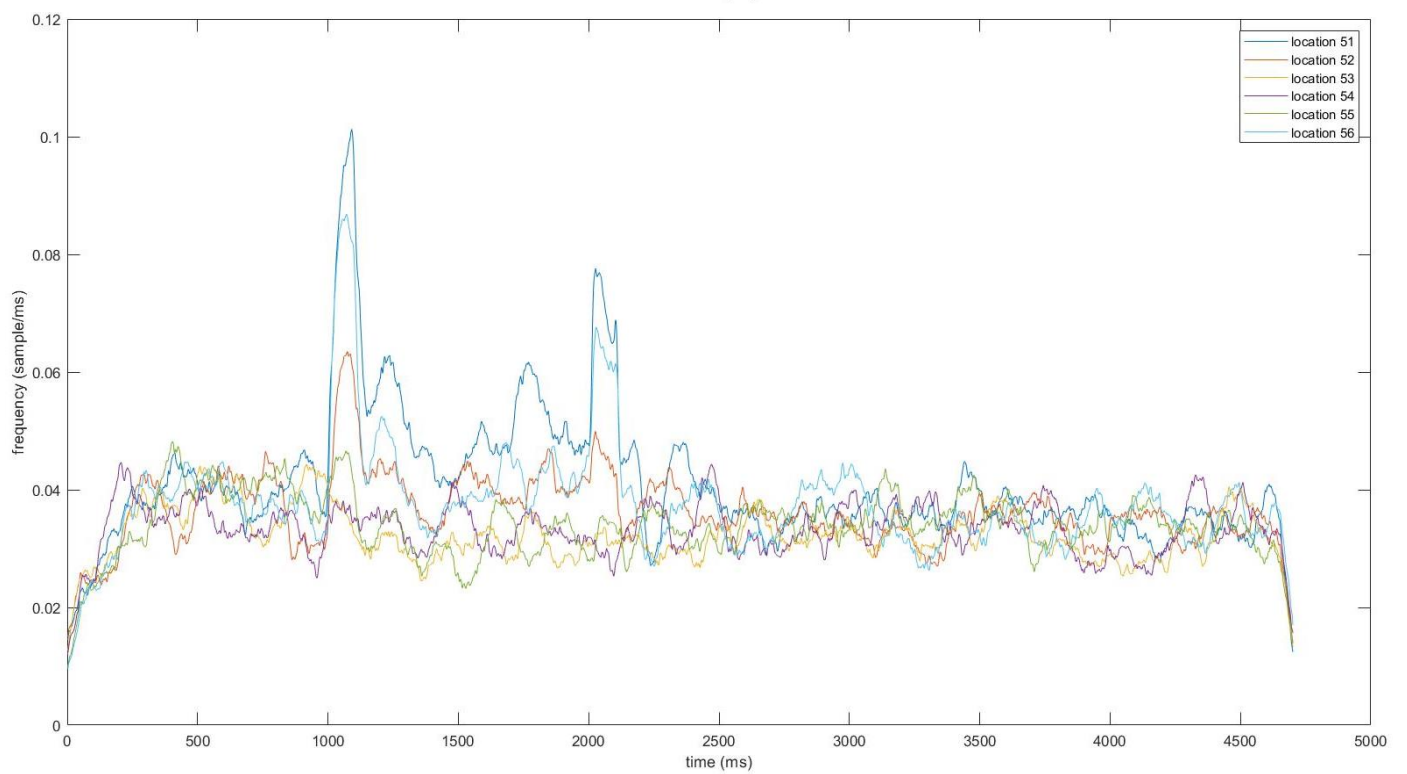
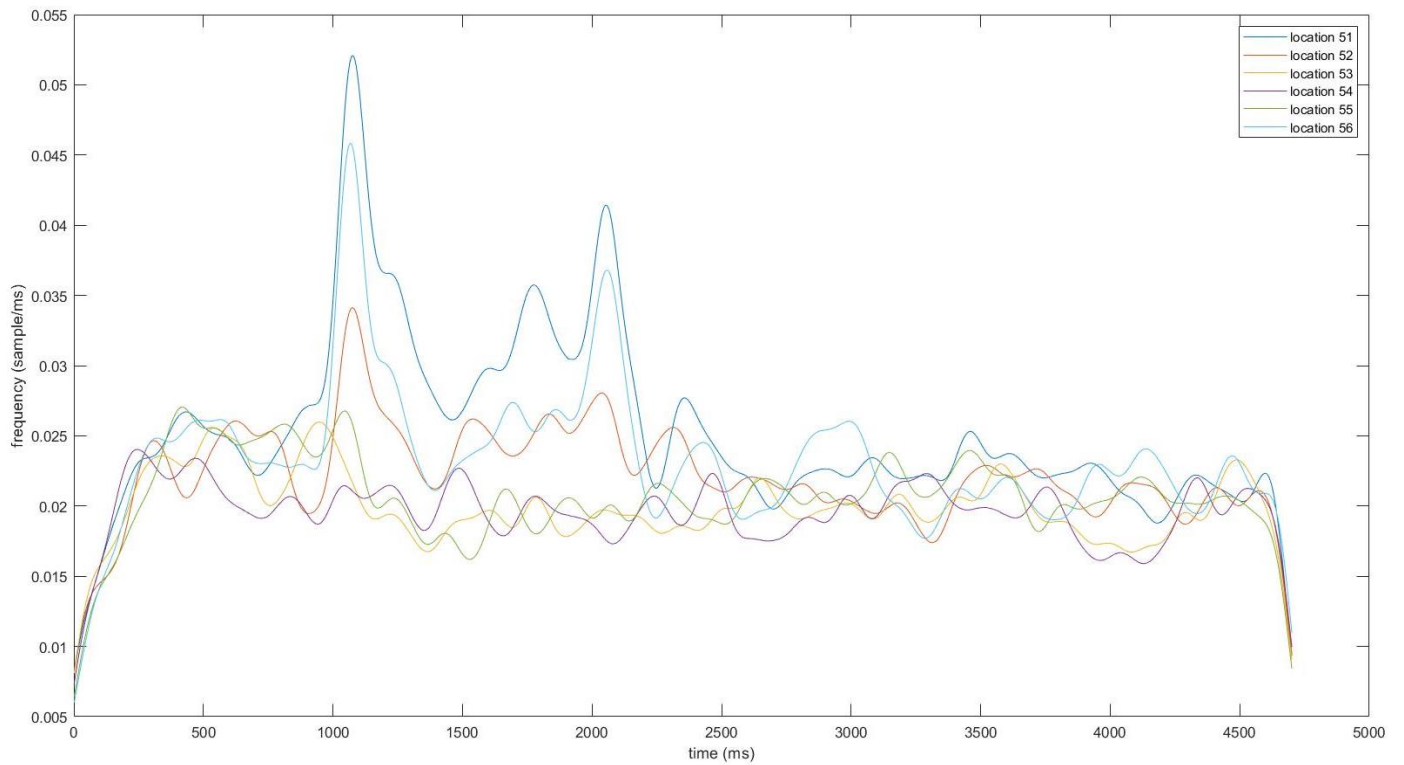
% psth
figure ;
subplot(211)
[ph_in,hh,rr] = psth(yek_in, 10, 1000, 73, 4701) ;
title('psth IN')
xlabel('time (ms)')
ylabel('spike summation')

subplot(212)
[ph_out,hho,rro] = psth(yek_out, 10, 1000, 64, 4701) ;
ylim([0 160])
title('psth OUT')
xlabel('time (ms)')
ylabel('spike summation')
```


2.2.1. File name :

2.b

2.2.2. Figures :



2.2.3. Explanation :

در این سوال پس از جدا کردن location های مختلف سراغ محاسبه و رسم firing rate رفتیم :

در تصویر اول یک کرنل بصورت گوسی با میانگین 0 و واریانس 0.05 تعریف کردیم و آن را با دستور conv روی trial ها اثر دادیم. سعی کردم که شکل ایجاد شده بصورت smooth نمایش داده شود و تغییرات کوچک را نادیده بگیرد. با این حال فیلتر استفاده شده علی نیست، به همین علت در مرحله بعد فیلتر دیگری اعمال کردیم تا این مشکل حل شود.

در تصویر بعدی یک فیلتر گوسی علی یا به عبارتی halfnormal طراحی کردم و مانند قبل آن را روی همه trial ها کانوالو کردم. سپس میانگین آن را برای همه location ها رسم کردم. شیفیت این حالت نسبت به حالت قبلی آنچنان مشهود نیست زیرا طول فیلتر نسبت به طول سیگنال کوچک است ولی با بررسی مقادیر دو سیگنال فیلتر شده در زمان یکسان می بینیم که سیگنال فیلتر شده با گوسی علی نسبت به سیگنال فیلتر شده با گوسی، یک شیفیت زمانی دارد.

2.2.4. Codes :

```
%% initialize
locations = squeeze(Event.mgs.codes(:,5)) ;
neuron = spike{4,1} ;

c51 = 0; c52 = 0; c53 = 0;
c54 = 0; c55 = 0; c56 = 0;

for i = 1:422
    if locations(i,1) == 51
        c51 = c51 + 1 ;
        l51(c51,:) = neuron.mgs(i,:) ;
    elseif locations(i,1) == 52
        c52 = c52 + 1 ;
        l52(c52,:) = neuron.mgs(i,:) ;
    elseif locations(i,1) == 53
        c53 = c53 + 1 ;
        l53(c53,:) = neuron.mgs(i,:) ;
    elseif locations(i,1) == 54
        c54 = c54 + 1 ;
        l54(c54,:) = neuron.mgs(i,:) ;
    elseif locations(i,1) == 55
        c55 = c55 + 1 ;
        l55(c55,:) = neuron.mgs(i,:) ;
    elseif locations(i,1) == 56
        c56 = c56 + 1 ;
        l56(c56,:) = neuron.mgs(i,:) ;
    end
end

%% gaussian filter

Y = unique(Event.mgs.codes(:,5));

y_index = cell([1 numel(Y)]);
for i=1:numel(Y)
    y_index{i} = find(Event.mgs.codes(:,5) == Y(i));
end

y_spike = cell([1 numel(Y)]);
for i=1:numel(Y)
    r = y_index{i};
    for j = 1:numel(y_index{i})
        z1(j,:) = spike{4, 1}.mgs(r(j),:);
    end
    y_spike{i} = z1;
    clear z1
    clear r
end

for i=1:numel(Y)
    r = y_spike{i};
    for j=1:size(r,1)
        sigma = 0.05;
        edge = (-3*sigma:.001:3*sigma);
        kernel = normpdf(edge,0,sigma);
        kernel = kernel*.001;
        A6(j,:)=conv(r(j,:),kernel,'same');
    end
    y_SDF{i} = A6;
end
```

```

figure ;
plot(mean(y_SDF{1}),'DisplayName','location 51'); hold on
plot(mean(y_SDF{2}),'DisplayName','location 52'); hold on
plot(mean(y_SDF{3}),'DisplayName','location 53'); hold on
plot(mean(y_SDF{4}),'DisplayName','location 54'); hold on
plot(mean(y_SDF{5}),'DisplayName','location 55'); hold on
plot(mean(y_SDF{6}),'DisplayName','location 56'); hold on
xlabel('time (ms)');
ylabel('frequency (sample/ms)');
legend ;
hold off

%% HalfNormal

for i=1:numel(Y)
    r = y_spike{i};
    for j=1:size(r,1)
        pd = makedist('HalfNormal','mu',0,'sigma',1000);
        x = 0:100 ;
        pdf_k = pdf(pd,x);
        A6(j,:)=conv(r(j,:),pdf_k,'same');
    end
    y_SDF{i} = A6*20;
    clear A6
    clear r
end

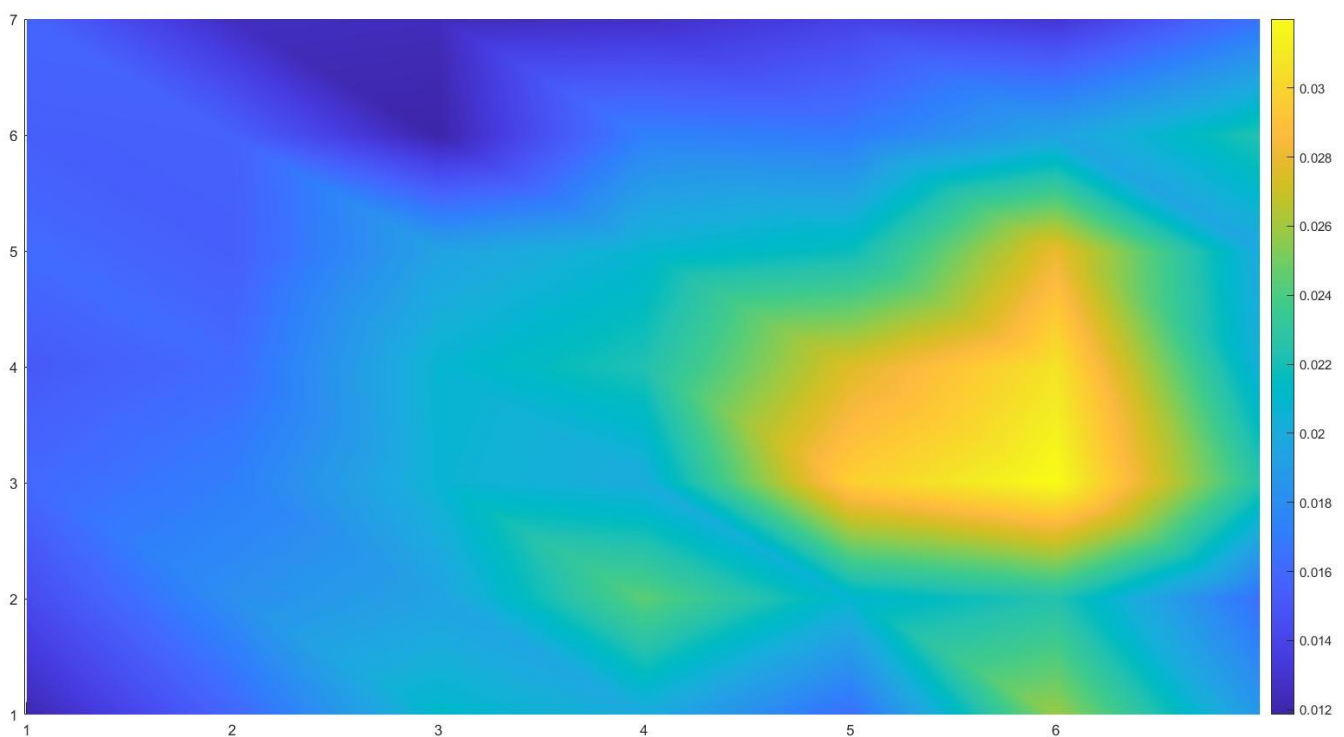
figure ;
plot(smooth(mean(y_SDF{1}),'DisplayName','location 51'); hold on
plot(smooth(mean(y_SDF{2}),'DisplayName','location 52'); hold on
plot(smooth(mean(y_SDF{3}),'DisplayName','location 53'); hold on
plot(smooth(mean(y_SDF{4}),'DisplayName','location 54'); hold on
plot(smooth(mean(y_SDF{5}),'DisplayName','location 55'); hold on
plot(smooth(mean(y_SDF{6}),'DisplayName','location 56'); hold on
xlabel('time (ms)');
ylabel('frequency (sample/ms)');
legend ;
hold off

```

3.1.1. File name :

3.a

3.1.2. Figures :



3.1.3. Explanation :

در این سوال ابتدا از قسمت event برای هر پروب trial ها را بدست آوردیم. سپس همه آن ها را در یک ماتریس جمع کرده و میانگین firing rate را محاسبه کردیم. سپس بر حسب جایگاه آنها در تصویر 7 در 7 میانگین firing rate ها را در یک ماتریس 7 در 7 به صورت متناظر قرار دادیم و نمودار نشان داده شده به عنوان receptive field نرون مورد بررسی به دست آمد.

3.1.4. Codes :

```
%% initialize
spikes = spike{4,1}.vodr ;
adrs = squeeze(Event.vodr.codes(:,10:17)) ;

delay = spikes(:,1:150) ;
prob{1,1} = spikes(:,151:351) ;
prob{1,2} = spikes(:,352:552) ;
prob{1,3} = spikes(:,565:765) ;
prob{1,4} = spikes(:,788:988) ;
prob{1,5} = spikes(:,990:1190) ;
prob{1,6} = spikes(:,1191:1391) ;
prob{1,7} = spikes(:,1400:1600) ;
prob{1,8} = spikes(:,1601:1801) ;

for k = 3:9
    for i = 1:7
        [r,c] = find(adrs == i + k*10) ;
        for m = 1:size(r,1)
            data(m,:) = prob{1,c(m,1)}(r(m,1),:) ;
        end
        answer(i,k-2) = sum(mean(data,1))/200 ;
    end
end

%% plot
figure ;
surface(answer)
shading interp
colorbar;
```

3.2.1. File name :

3.b

3.2.2. Figures :

[3.b\1.mp4](#)

3.2.3. Explanation :

در این سوال ابتدا به ازای هر trial یک ماتریس ساختیم. این ماتریس به گونه ای است که نشان می دهد در هر لحظه کدام پروب از پروب های صفحه 7 در 7 روشن شده است. به عبارتی هر trial را به یک ماتریس 7 در 7 در 1901 تبدیل کردیم که عدد آخر زمان می باشد. این کار نسبت به تاخیر ها و زمان هایی صورت گرفته که پروب ما روشن شده است، یعنی زمان های 200 میلی ثانیه ای را در نظر گرفتیم که در آن بازه یک پروب به نمایش در آمده است.

در مرحله بعد با توجه به ماتریس spike همه زمان هایی که اسپایک اتفاق افتاده و ما در آن ماتریس عدد یک را می بینیم در نظر گرفته و از 100 میلی ثانیه قبل از آن لحظه همه ماتریس های 7 در 7 متناظر با آن trial را برداشتیم و همه را در یک ماتریس قرار دادیم که به تعداد اسپایک ها سطر و به اندازه زمان در نظر گرفته شده یا همان 100، ستون دارد.

سپس همه لحظات متناظر را با هم میانگین گرفتیم و به یک ماتریس 7 در 7 در 100 رسیدیم و این ماتریس های میانگین گیری شده را پشت سر هم و با یک تاخیر نشان دادیم تا به صورت فیلم نمایش داده شود.

فیلم نهایی در پوشه مربوط به سوال موجود است. همچنین با کلیک بر روی لینک بالا نمایش داده می شود.

3.2.4. Codes :

```
%% stimulus matrix

for i = 1:size(Event.vodr.codes,1)
    stim{i} = zeros(7,7,1901) ;
end

for i = 1:size(Event.vodr.codes,1)
    for j = 10:17

        probe = Event.vodr.codes(i,j) ;
        col   = floor(probe/10) - 2 ;
        row   = mod(probe , (col + 2) * 10) ;

        stim{i}(row , col , (j-10)*200+200:(j-10)*200+400) = 1 ;

    end
end

%% Pre-STA
spike_counter = 1 ;
for i = 1:146
    trial_matrix = neuron.vodr(i,:) ;
    spikes       = find(trial_matrix == 1) ;

    for j = 1:length(spikes)

        index = spikes(1,j) ;
        if index >= 100

            pre_sta(spike_counter) = stim{i}(:, :, index-99:index) ;
            spike_counter = spike_counter + 1 ;

        end

    end
end

end

%% STA
STA_final = zeros(7,7,100) ;

for i = 1:100

    for j = 1:length(pre_sta)

        STA_final(:, :, i) = STA_final(:, :, i) + pre_sta{1,j}(:, :, i) ;

    end

end

STA_final = STA_final / length(pre_sta) ;

%% display
for i = 1:100

    surface(STA_final(:, :, i))
    colorbar
    shading interp
    pause(0.001)

end
```