

به نام خدا

تمرین شماره ۳

درس کنترل سیستم های عصبی عضلانی

تهیه کننده: علیرضا امیری

شماره دانشجویی: ۴۰۲۰۲۴۱۴

استاد درس: دکتر دلربایی

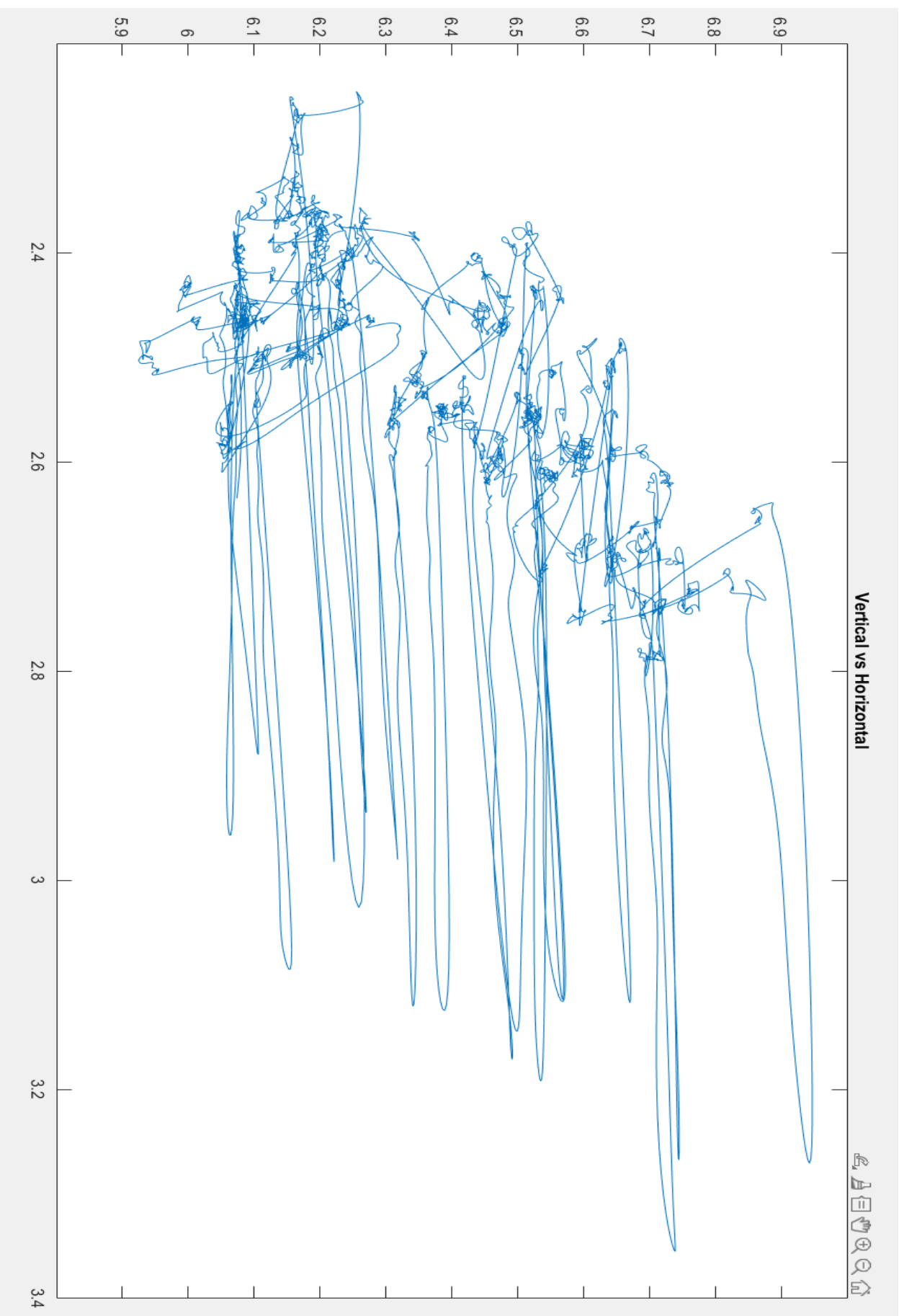
زمستان ۱۴۰۲

برای انجام این تمرین، ابتدا داده ها را به وسیله ی دستور **load** به محیط متلب وارد می کنیم.

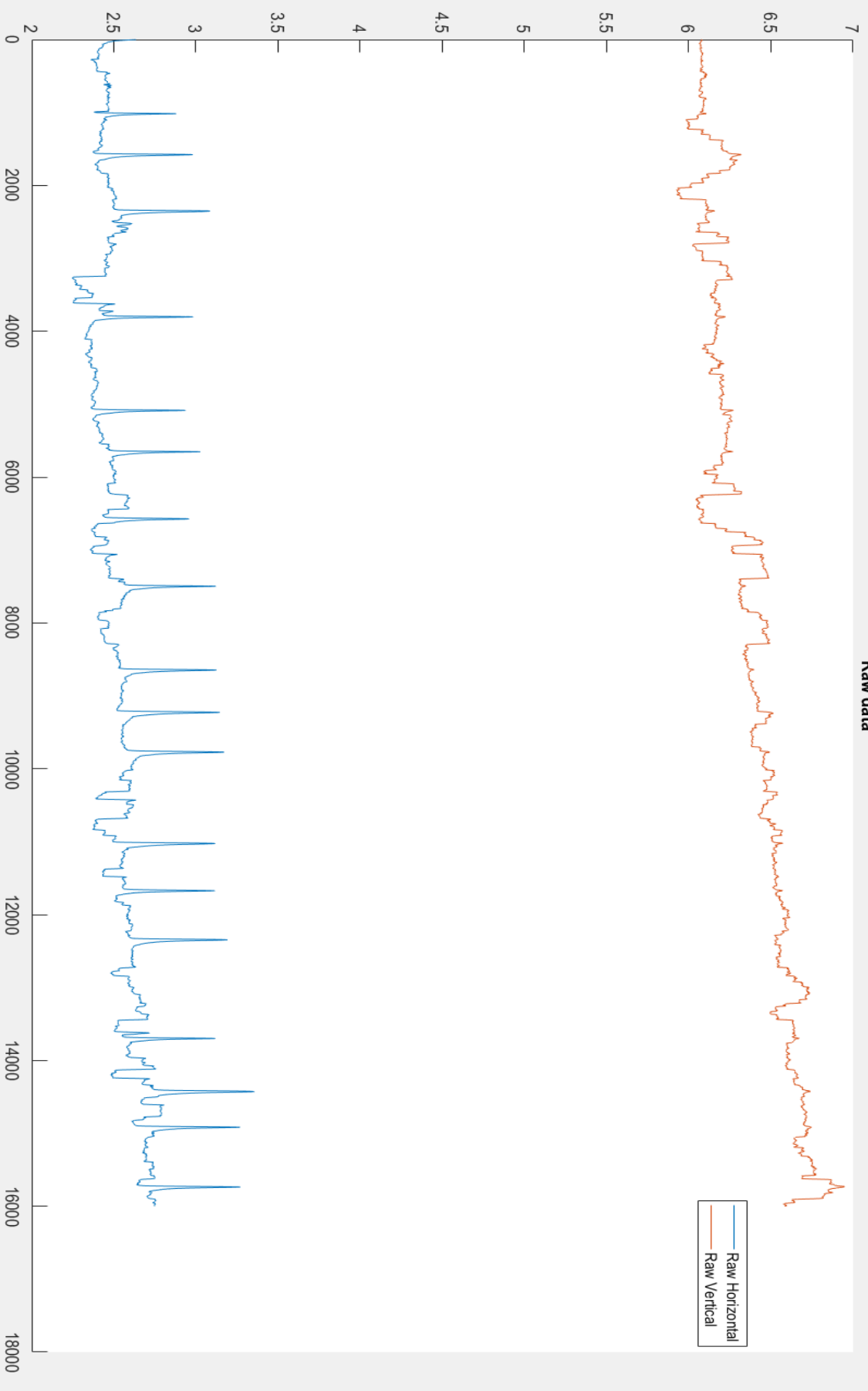
داده های موجود، مربوط به موقعیت چشم در دو راستای افقی و عمودی می باشد. بنابراین ابتدا

این داده ها را نمایش می دهیم.

```
Raw_Horizontal = data.B;  
Raw_Vertical = data.A;  
  
figure(1)  
hold on  
plot(1:length(Raw_Horizontal) , Raw_Horizontal);  
plot(1:length(Raw_Vertical) , Raw_Vertical);  
title('Raw data')  
legend('Raw Horizontal', 'Raw Vertical')  
  
figure(2)  
plot(Raw_Horizontal , Raw_Vertical)  
title('Vertical vs Horizontal')
```



Raw data



می‌دانیم در دیتا‌های الکترواکولوگرافی، پرش‌هایی به دلیل پلک زدن در سیگنال عمودی به وجود می‌آیند. با بررسی نمودار فوق، مشخص می‌شود که ستون اول داده‌ها مربوط به سیگنال عمودی و ستون دوم مربوط به سیگنال افقی می‌باشد.

پاسخ سوال ۱:

در این بخش، آفست و ترند داده‌های خام افقی و عمودی که در بخش قبل مشخص شد را به وسیله‌ی دستور `detrend` حذف می‌کنیم و نمودار داده‌ها را قبل و بعد از پردازش رسم می‌کنیم.

```
% Detrend data and plot Signals  
% Question 1
```

```
Detrended_Raw_Horizontal = detrend(Raw_Horizontal);  
Detrended_Raw_Vertical = detrend(Raw_Vertical);
```

```
figure(2)  
title('Raw signals vs Detrended Signals')  
hold on  
plot(1:length(Raw_Horizontal) , Raw_Horizontal);  
plot(1:length(Raw_Vertical) , Raw_Vertical);  
plot(1:length(Detrended_Raw_Horizontal) , Detrended_Raw_Horizontal);  
plot(1:length(Detrended_Raw_Vertical), Detrended_Raw_Vertical);  
legend('Raw Horizontal', 'Raw Vertical' , 'Detrended Raw Horizontal' , 'Detrended Raw  
Vertical' , 'Location','best')
```

```
>> mean(Detrended_Raw_Horizontal)
```

```
ans =
```

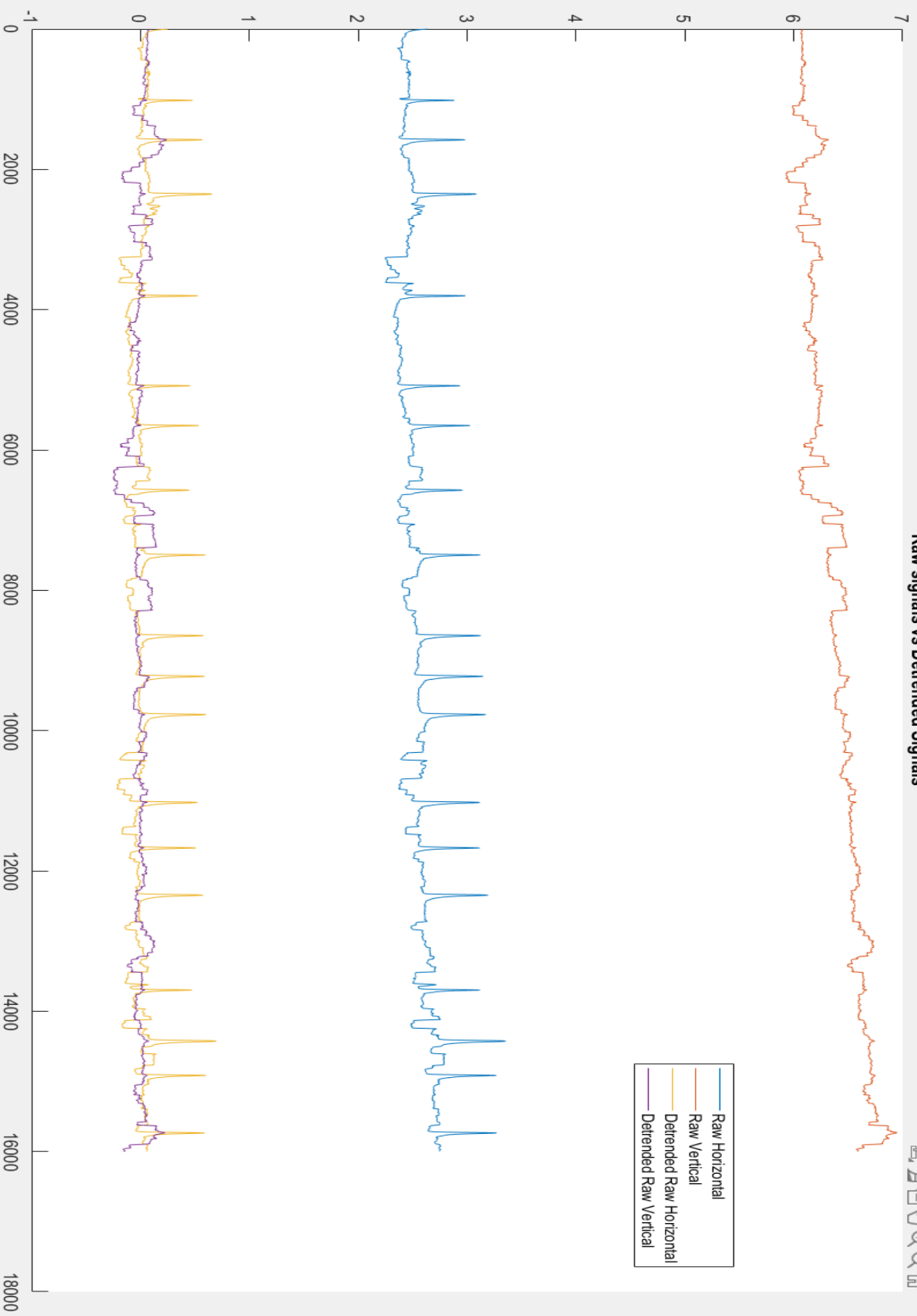
```
-1.3380e-16
```

```
>> mean(Detrended_Raw_Vertical)
```

```
ans =
```

```
-1.4376e-15
```

Raw signals vs Detrended Signals



پس از اجرای این دستور، ترند و آفست از سیگنال ها حذف می شود و همانطور که در تصویر نمایش داده شده است، میانگین داده های پردازش شده برابر با صفر می شود.

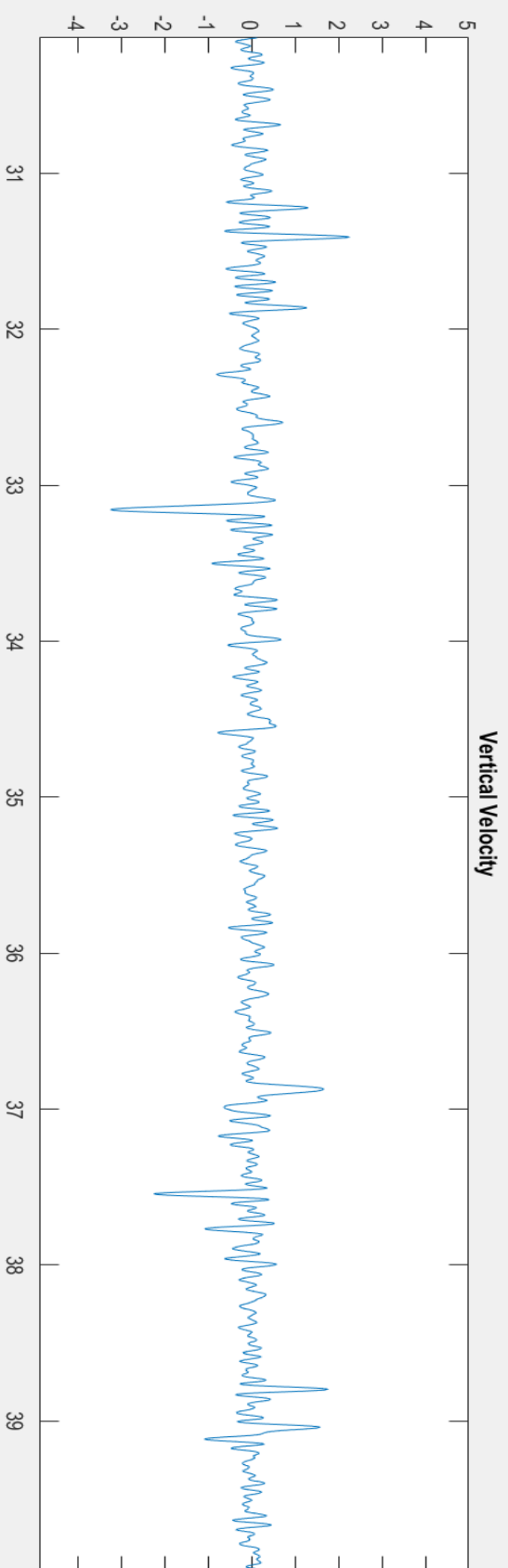
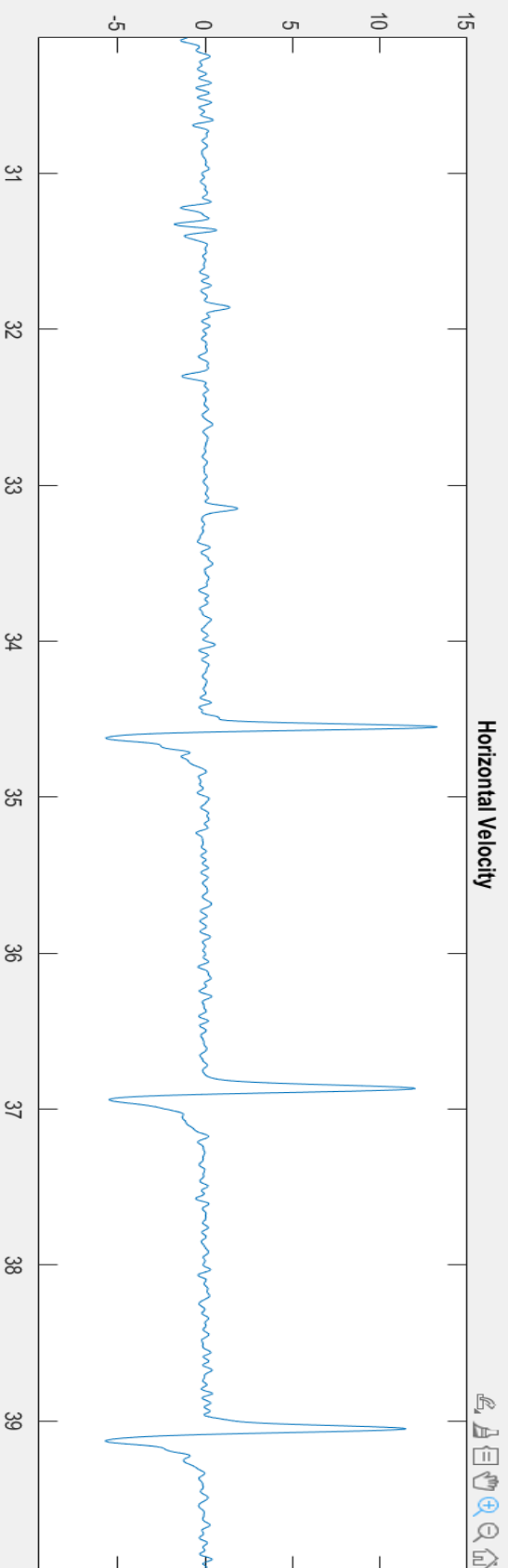
پاسخ سوال ۲:

برای تشخیص پلک زدن، تثبیت و جهش، علاوه بر موقعیت چشم، به داده های مربوط به سرعت چشم نیز نیاز داریم. بنابراین ابتدا با محاسبه ی مشتق زمانی از داده های موقعیت، سرعت حرکت چشم در دو راستای افقی و عمودی به دست می آید

```
%assuming sample rate is 250 Hz
time = (1:length(Raw_Horizontal))/250;
velH = diff(Detrended_Raw_Horizontal)./diff(time);
velV = diff(Detrended_Raw_Vertical)./diff(time);
figure(4)
subplot(2,1,1)
plot(time , [velH , 0])
title('Horizontal Velocity')
subplot(2,1,2)
plot(time , [velV , 0])
title('Vertical Velocity')
```

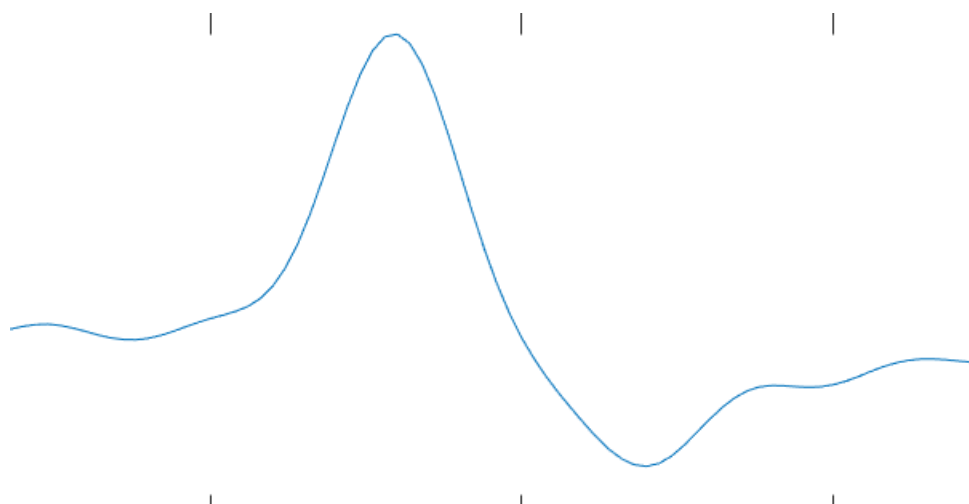
همچنین، مقدار سرعت برآیند، به صورت زیر محاسبه می شود.

```
Velocity = sqrt(velH.^2 + velV.^2);
```

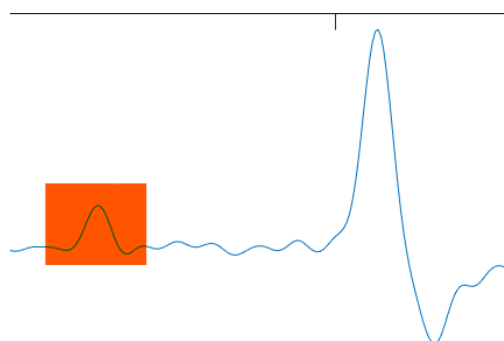


حال، برای تشخیص موارد خواسته شده، نمودار های سرعت و موقعیت را در کنار یکدیگر رسم می کنیم.

تشخیص پلک زدن: در لحظه ی پلک زدن، سرعت چشم در راستای عمودی نوسانی ناگهانی دارد که ابتدا به یک پیک مثبت رسیده، سپس تا مقداری منفی کم شده و در نهایت به صفر نزدیک می شود.

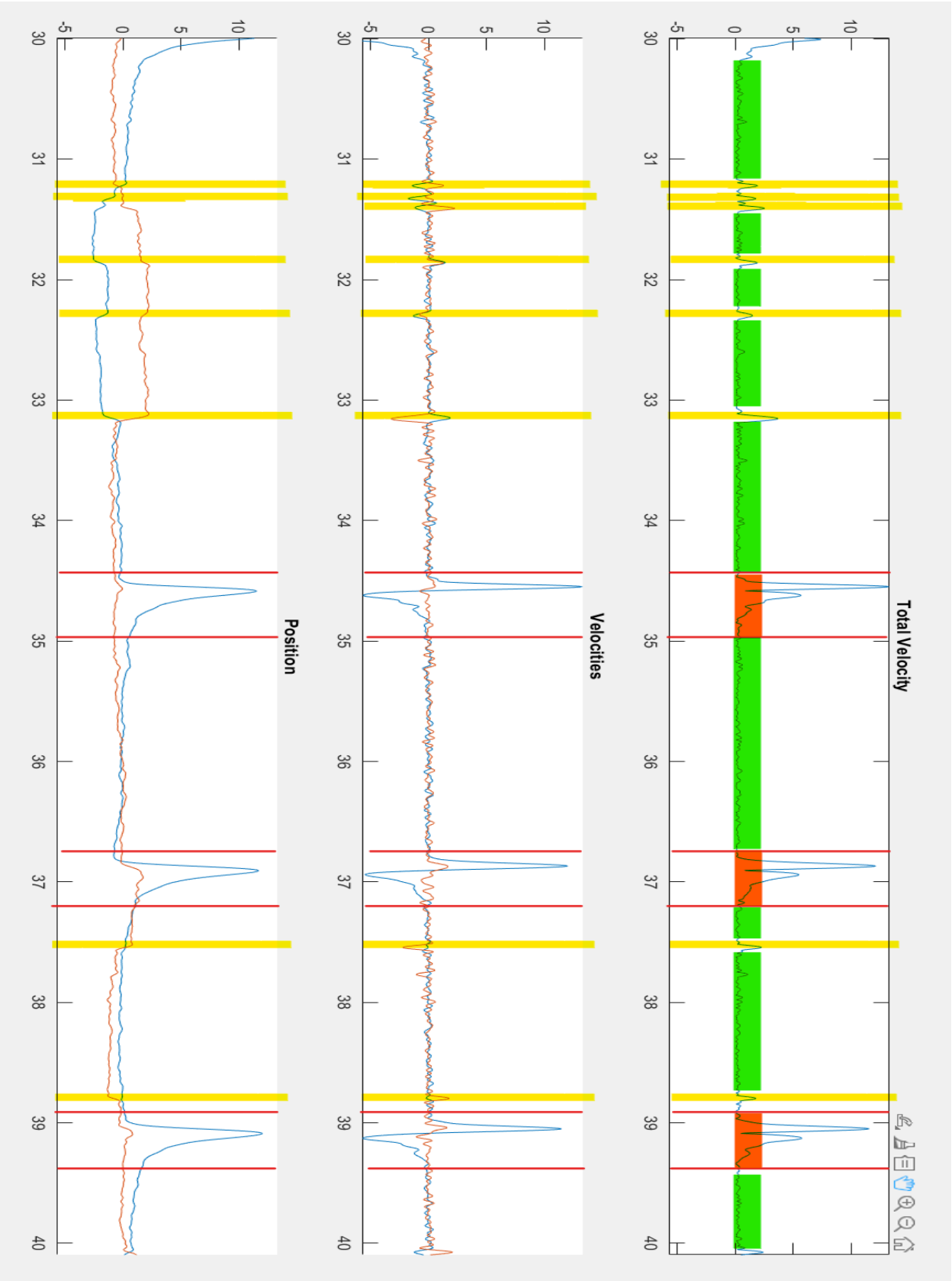


تشخیص جهش: جهش های چشم، به صورت نوسان هایی با دامنه های کمتر از پلک زدن بر روی نمودار سرعت قابل مشاهده است.



تشخیص تثبیت: فواصل بین جهش ها و یا بین جهش و پلک زدن، ناحیه‌ی تثبیت در نظر گرفته می‌شود. در این نقاط، سرعت چشم تقریباً برابر با صفر بوده و نوساناتی با دامنه های کم حول صفر دارند.

در نمودار صفحه‌ی بعد، بازه‌ی زمانی ۳۰ تا ۴۰ ثانیه در نظر گرفته شده و حرکت پلک زدن به رنگ قرمز، جهش به رنگ زرد و تثبیت به رنگ سبز مشخص شده است.



پاسخ سوال ۳:

برای محاسبه‌ی فرکانس های پلک زدن، جهش و تثبیت، باید تعداد وقوع هر یک را به دست آوریم تا با تقسیم این تعداد بر زمان کل، فرکانس وقوع آن حرکت به دست آید.

برای این منظور، باید مقادیر آستانه‌ای برای تشخیص پلک زدن و جهش تعریف شود تا بتوانیم با شمردن تعداد دفعاتی که سیگنال از آن مقادیر آستانه بالاتر رفته است، تعداد وقوع حرکت را به دست آوریم. برای این منظور، تابعی به صورت زیر تعریف می‌شود که نقاطی که سیگنال خط آستانه را رو به بالا قطع می‌کند (Cross Over) را به دست می‌آورد.

```
% function to find zero crossings
function [Zx] = find_zc(x, y, threshold)
    y = y - threshold;
    zci = @(data) find(diff(sign(data)) > 0); % function: returns indices of +ZCs
    ix = zci(y); % find indices of + zero crossings of x
    ZeroX = @(x0, y0, x1, y1) x0 - (y0.*(x0 - x1))./(y0 - y1); % Interpolated x value
    for Zero-Crossing
        Zx = ZeroX(x(ix), y(ix), x(ix+1), y(ix+1));
    end
```

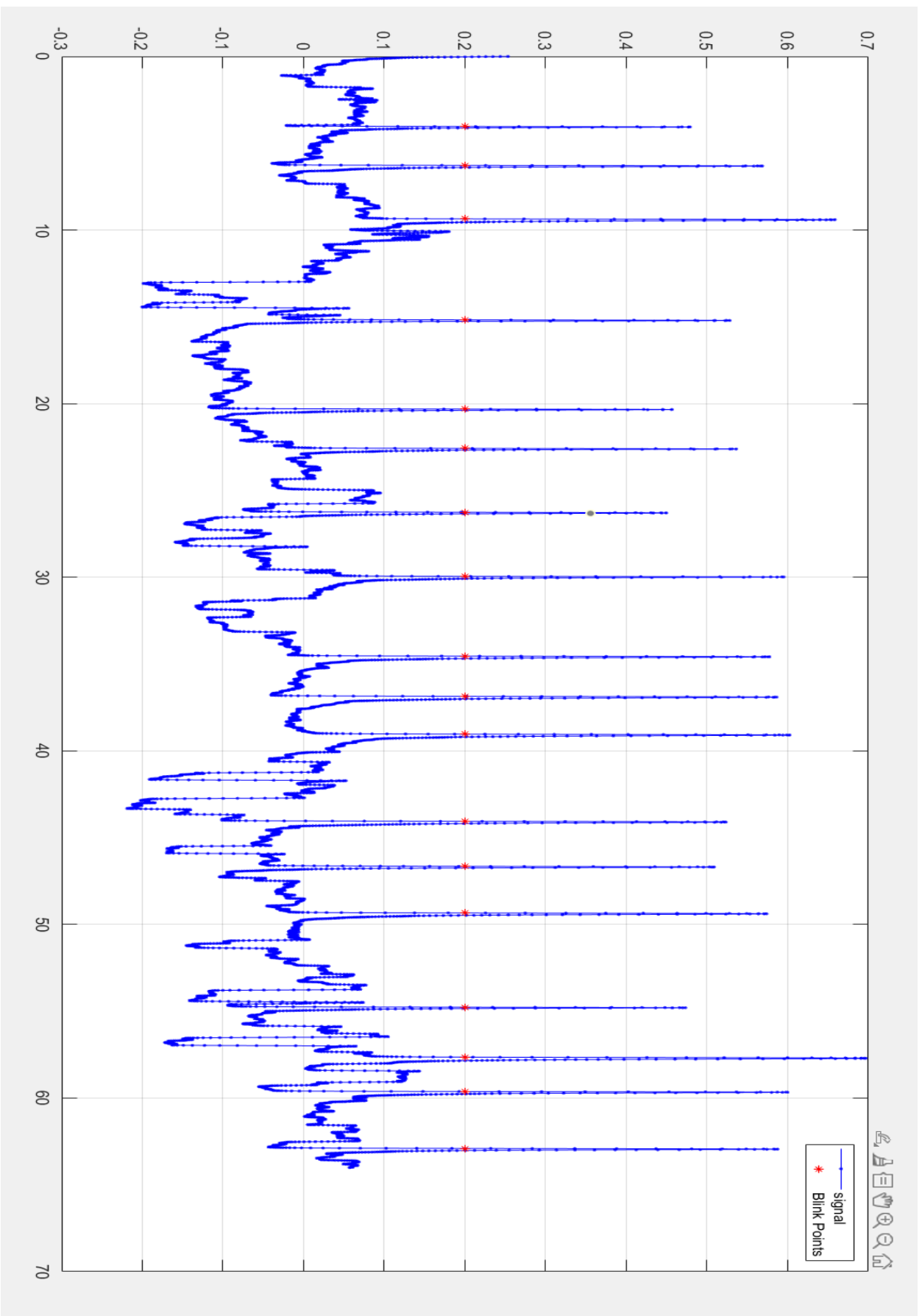
حال با در اختیار داشتن این تابع، اقدام به محاسبه‌ی تعداد دفعات وقوع حرکت های چشم می‌کنیم:

فرکانس پلک زدن: برای محاسبه‌ی این متغیر، بر روی سیگنال موقعیت عمودی مقدار آستانه‌ای برابر با ۰.۲ قرار می‌دهیم و تابع را بر روی آن اجرا می‌کنیم. پس از اجرای دستور، نقاط برخورد در متغیر Blinks ذخیره می‌شود. سپس با محاسبه‌ی تعداد آن و تقسیم بر زمان کل، فرکانس پلک زدن به دست می‌آید.

```
% Calculating Blink, Fixation and Saccade frequency
% Question 3
% Frequency of Blinking
Blink_threshold = 0.2;
Blinks = find_zc(time, Detrended_Raw_Horizontal, threshold)
Blink_Frequency = numel(Blinks)/time(16001)
figure(6)
plot(time, Detrended_Raw_Horizontal, 'b.-', Blinks,
Blink_threshold*ones(size(Blinks)), '*r', 'linewidth', 0.5, 'markersize', 5);
grid on
legend('signal', 'Blink Points');
```

Blink_Frequency =

0.2812



فرکانس جهش: برای محاسبه‌ی این متغیر، بر روی سیگنال سرعت برابند، مقدار آستانه‌ای برابر

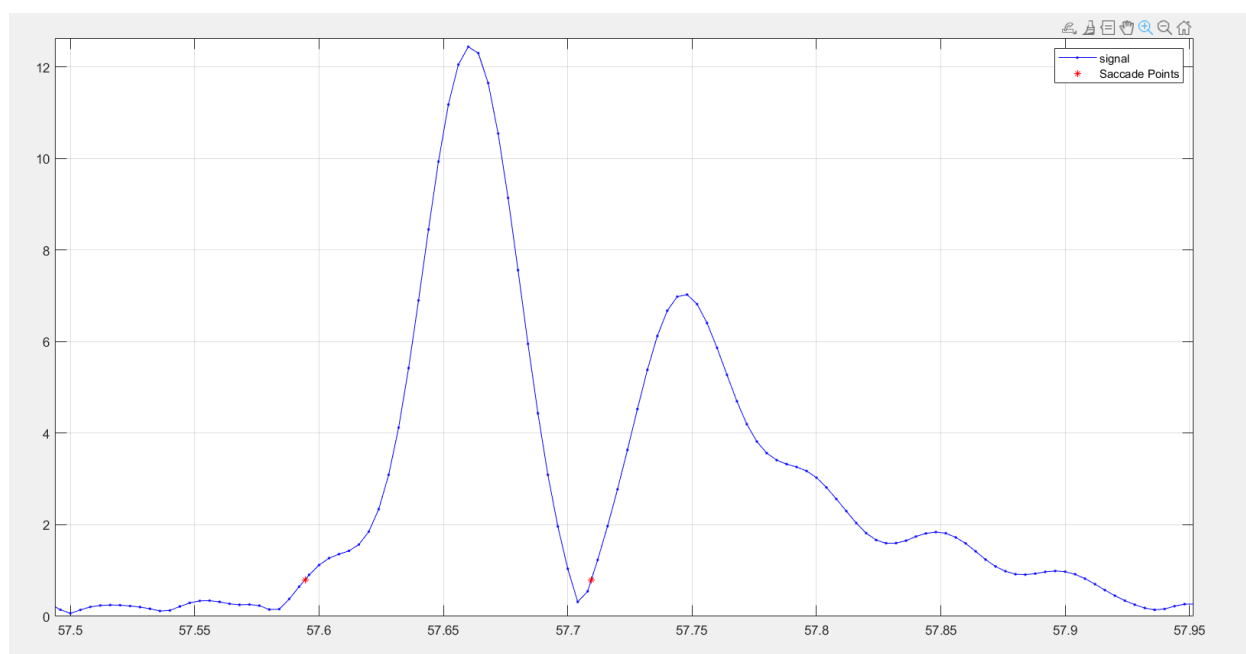
با ۰.۸ قرار می‌دهیم و تابع را بر روی آن اجرا می‌کنیم. پس از اجرای دستور، نقاط برخورد در

متغیر Saccades ذخیره می‌شود. نقاط ذخیره شده در این متغیر، تمام نقاط گذر سیگنال از

مقدار آستانه می‌باشد که این نقاط، شامل نقاط پلک زدن نیز می‌باشد. بنابراین، باید نقاط مربوط

به پلک زدن را از آن کم کنیم که با توجه به نمودار، به ازای هر مرتبه پلک زدن دو بار سیگنال از

مقدار آستانه بالاتر می‌رود.



```
% Frequency of Saccade
```

```
Saccade_threshold = 0.8;
```

```
Saccades = find_zc(time, [Velocity,0], Saccade_threshold)
```

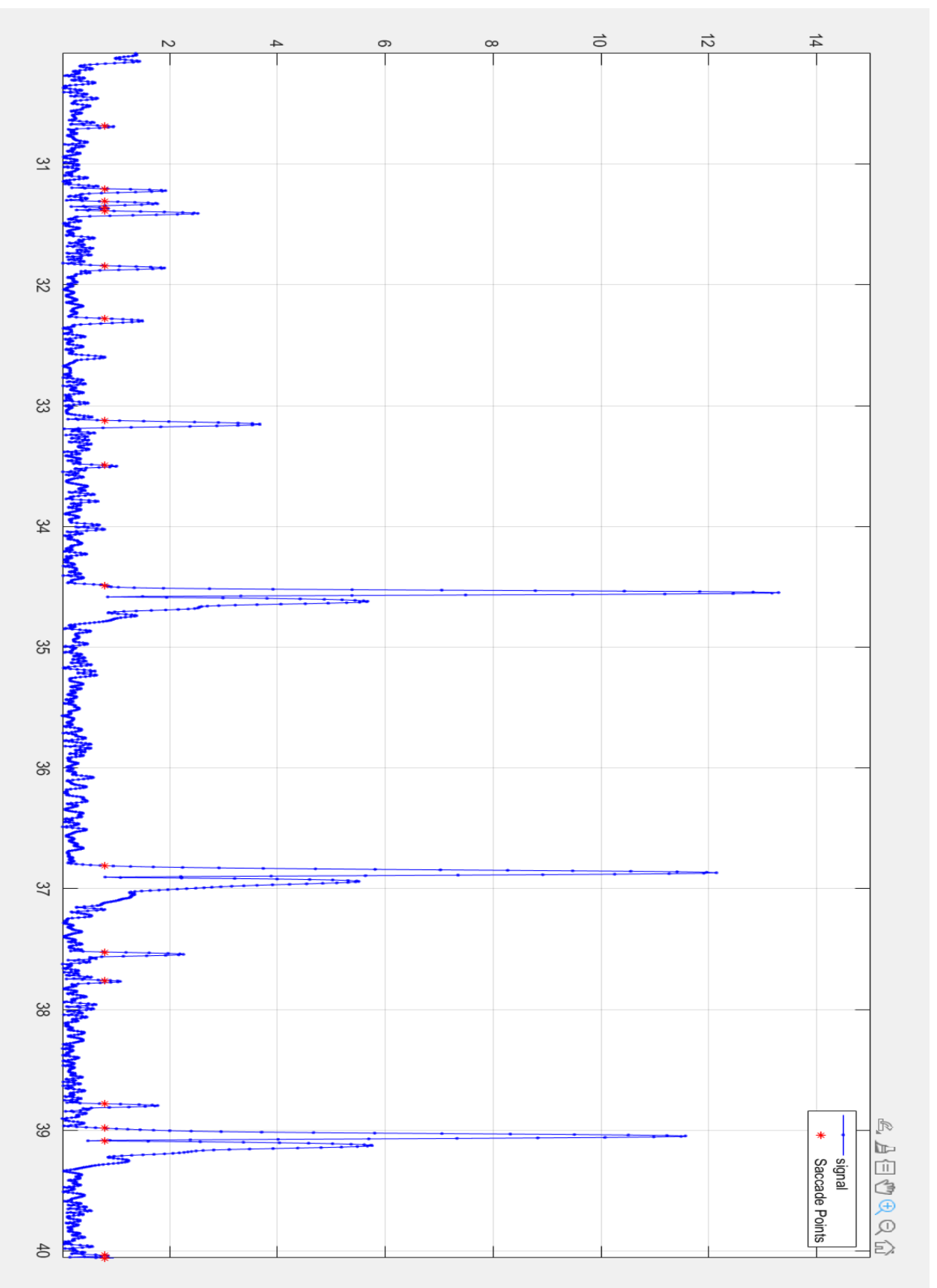
```
Saccade_Frequency = (numel(Saccades)-(numel(Blinks) * 2))/time(16001)
```

```
figure(7)
```

```
plot(time, [Velocity,0], 'b.-', Saccades, Saccade_threshold*ones(size(Saccades)),  
'*r', 'linewidth', 0.5, 'markersize', 5);
```

```
grid on
```

```
legend('signal', 'Saccade Points');
```



Saccade_Frequency =

2.8436

فرکانس تثبیت: محاسبه‌ی تعداد دفعات وقوع تثبیت، با در اختیار داشتن دفعات وقوع پلک زدن

و جهش ممکن است، چرا که هر فاصله‌ی بین دو جهش و یا یک جهش و پلک زدن، یک مرتبه

وقوع تثبیت محسوب می‌شود. بنابراین، تعداد وقوع تثبیت به صورت زیر محاسبه می‌شود:

Blink + Saccade + 1

در نهایت با تقسیم آین عدد بر زمان کل، فرکانس تثبیت به دست می‌آید.

%Frequency of Fixation

Fixation_Frequency = (numel(Saccades)-numel(Blinks)+1)/time(16001)

Fixation_Frequency =

3.1404

ممنون از توجه شما