

به نام خدا



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده برق و کامپیوتر



ریاضی مهندسی

تمرین کامپیوتری شماره ۱

آرمان رستمی

۸۱۰۱۹۶۶۰۹

## فروردین ماه ۱۳۹۸

### فهرست

عنوان	شماره صفحه
<a href="#"><u>چکیده</u></a>	۳
<a href="#"><u>بخش ۱</u></a>	۴
<a href="#"><u>بخش ۲</u></a>	۶
<a href="#"><u>بخش ۳</u></a>	۸
<a href="#"><u>بخش ۴</u></a>	۹
<a href="#"><u>پیوست ۱: روند اجرای برنامه</u></a>	۱۰

## چکیده

در این تمرین به بررسی یکی از روش های اندازه گیری ضربان قلب به نام تحلیل سیگنال های فتوپلتیسموگرام<sup>۱</sup> پرداخته می شود. در این روش با تاباندن نور به بافت، میزان نور بازتاب شده با توجه به میزان فشار خون تغییر می کند. نرخ این تغییرات متناسب با نرخ ضربان قلب می باشد.

برای بدست آوردن فرکانس این تغییرات از تبدیل سریع فوریه<sup>۲</sup> استفاده می کنیم. تبدیل فوریه، یک سیگنال در حوزه زمان را به حوزه فرکانس انتقال می دهد. پس از آن برای بدست آوردن فرکانس مورد نظر و همچنین حذف سیگنال های زائد<sup>۳</sup> بهره می بریم.

در این تمرین ابتدا فایلی تصویری شامل چند ثانیه تصویر از دوربین گوشی در حالی که فلش گوشی روشن بوده ایجاد می کنیم و سپس به کمک نرم افزار متلب<sup>۴</sup> آن را پردازش می کنیم.

---

<sup>۱</sup>Photoplethysmogram (PPG)

<sup>۲</sup>Fast Fourier transform (FFT)

<sup>۳</sup>Noise

<sup>۴</sup>Matlab

### خواندن تصاویر از فیلم ضبط شده

#### خواندن فیلم گرفته شده از بافت

به کمک دستور VideoReader فیلم را می خوانیم. آرگومان ورودی این دستور آدرس فیلم می باشد که در صورتی که فقط نام فیلم نوشته شود متناظر با پوشه ای که کد در آن است می باشد. خروجی این دستور یک آبجکت می باشد. برای بدست آوردن تعداد فریم های از رابطه زیر استفاده می کنیم:

$$\text{Number of Frames} = \text{video.Duration} * \text{video.FrameRate}$$

#### بدست آوردن میزان رنگ قرمز در هر فریم

برای بدست آوردن این مقدار از دستور readFrame استفاده می کنیم. آرگومان ورودی این دستور آبجکت بدست آمده از دستور VideoReader می باشد. خروجی این دستور یک ماتریس<sup>۱</sup> ۳ بعدی می باشد که به ترتیب در برگیرنده میزان رنگ های قرمز، سبز و آبی در هر پیکسل فریم می باشد. رنگ قرمز هر پیکسل در هر فریم را به این روش استخراج می کنیم و در آرایه ای ذخیره می کنیم.

#### بدست آوردن میزان میانگین رنگ قرمز در هر فریم

در مرحله قبل میزان رنگ قرمز هر پیکسل در هر فریم را بدست آوردیم. برای بدست آوردن میانگین این رنگ ها در هر فریم میانگین هر ستون و سپس میانگین این نتایج را بدست می آوریم و یک عدد مربوط به میانگین رنگ قرمز در آن فریم بدست می آید. سپس آن ها را در یک آرایه ذخیره میکنیم. به طور دقیق تر کد متلب برای بدست آوردن این میانگین در فریم i ام به شرح زیر می باشد:

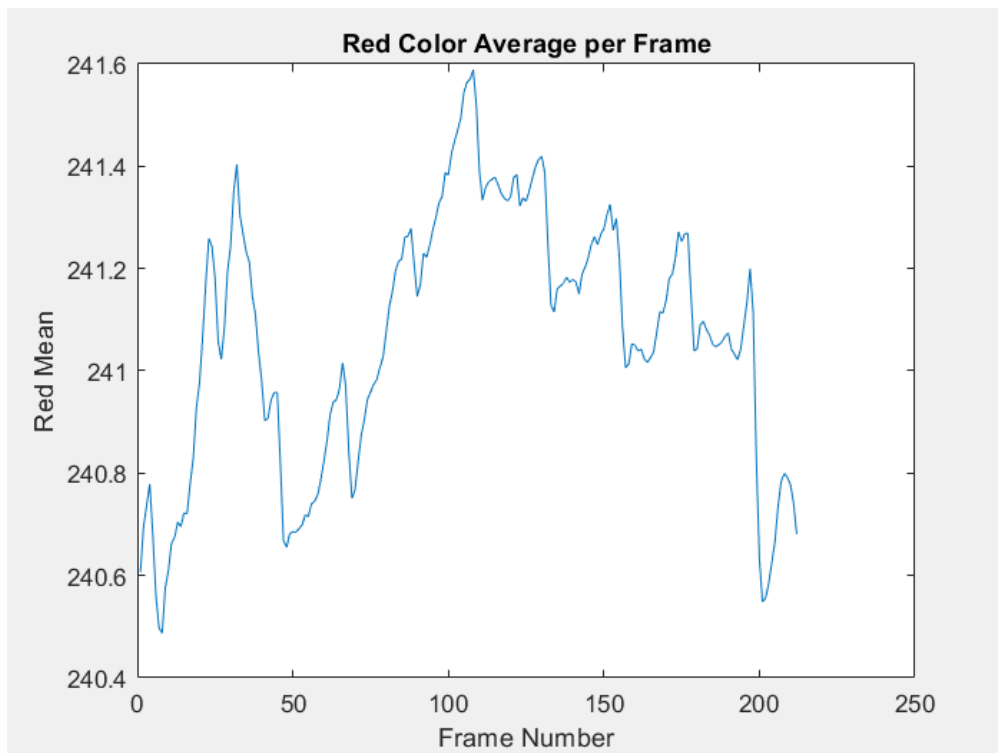
```
redMean(i) = mean(mean(redFrames(:, :, i))));
```

---

<sup>1</sup>Matrix

### رسم نمودار میانگین رنگ قرمز در هر فریم

برای این کار از دستور plot استفاده می کنیم که آرگومان ورودی آن آرایه میانگین های رنگ قرمز می باشد. نتیجه حاصله در شکل ۱ قابل رویت می باشد.



شکل ۱- میانگین رنگ قرمز در هر فریم فیلم

## تحلیل فوریه

بدست آوردن فرکانس نمونه برداری<sup>۱</sup>

فرکانس نمونه برداری عبارت است از تعداد نقاط نمونه برداری شده در واحد زمان یا مکان. در این جا فرکانس نمونه برداری برابر با تعداد فریم ها در هر ثانیه می باشد. برای بدست آوردن آن از آجکت video استفاده می کنیم. فرکانس نمونه برداری برابرست با `video.FrameRate`

## بدست آوردن ضرایب فوریه هر فرکانس

برای بدست آوردن ضرایب فوریه هر فرکانس از دستور `fft` استفاده می کنیم. آرگومان ورودی این تابع در این جا مقادیر میانگین قرمزی در هر فریم می باشد که در بخش قبل بدست آمد. نکته حائز اهمیت در این جا این است که این دستور ابتدا ضرایب فوریه محدوده فرکانس  $[0, \frac{Fs}{2}]$  و سپس ضریب مربوط به محدوده فرکانس  $(0, \frac{-Fs}{2}]$  را بر می گرداند. به عبارت دیگر تقریباً نصفه اول خروجی باز گردانده شده از این تابع مربوط به محدوده فرکانس  $[0, \frac{Fs}{2}]$  و بقیه مربوط به محدوده فرکانس  $(\frac{-Fs}{2}, 0]$  می باشد. فرکانس کاربردی در این جا مربوط به محدوده فرکانسی  $[0, \frac{Fs}{2}]$  می باشد چون به کمک آن می توان قله سیگنال فرکانسی و در نتیجه فرکانس سیگنال را یافت.

## بدست آوردن ضرایب فوریه متناظر با مقدار ضربان قلب در بازه [50, 220] در دقیقه

فرکانس متناظر با هر ضربان قلب متناظر است با  $\frac{1}{60}$  هرتز. یعنی داریم:

$$1 \text{ BPM} = \frac{1}{60} \text{ Hz}$$

پس باید ضرایب فوریه متناظر در محدوده فرکانسی  $[\frac{50}{60}, \frac{220}{60}]$  که تقریباً معادل با محدوده فرکانسی [0.8333, 3.6667] می باشد را یافت.

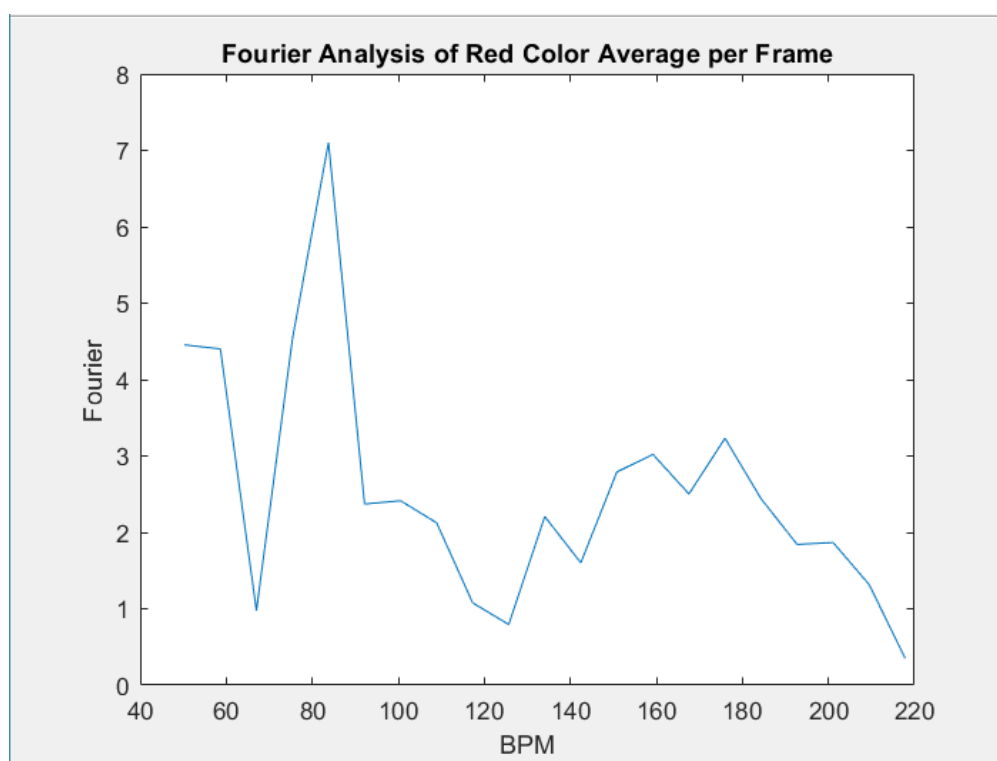
برای این کار ابتدا فرکانس نمونه گیری به اندازه تعداد فریم ها تقسیم بندی شده و سپس به یک عدد در فریم مورد نظر متناظر شده است. چون فرکانس مورد نظر در بازه  $[0, \frac{Fs}{2}]$  می باشد در این بازه به یافتن فرکانس های محدوده  $[\frac{50}{60}, \frac{220}{60}]$  می پردازیم. برای این کار از دستور `find` استفاده می

<sup>۱</sup>Sampling Frequency (Fs)

کنیم. به عنوان آرگومان ورودی، فرکانس های در بازه  $[0, \frac{F_s}{2}]$  را به تابع می دهیم. خروجی این دستور تعدادی index می باشد که متناظر با index فرکانس های در بازه مورد نظر است. به ازای این index ها ضرایب فوریه متناظر را نیز داریم.

### رسم نمودار ضرایب فوریه متناظر با ضربان قلب در بازه [50, 220] در دقیقه

می دانیم هر BPM معادل با  $\frac{1}{6}$  هرتز می باشد. پس برای رسم ضرایب فوریه متناظر با ضربان قلب، فرکانس های بدست آمده را در ۶۰ ضرب می کنیم و به عنوان محور افقی نمایش می دهیم. نکته حائز اهمیت در این جا این است که دستور fft ضرایب فوریه را به صورت عدد مختلط<sup>۱</sup> خروجی می دهد. از دستور abs برای بدست آوردن مقادیر حقیقی<sup>۲</sup> این ضرایب کمک می گیریم. به کمک دستور plot نمودار مورد نظر را رسم می کنیم که نتایج حاصله در شکل ۲ موجود می باشد.



شکل ۲- ضرایب فوریه متناظر با ضربان قلب در بازه [50, 220] در دقیقه

<sup>۱</sup>Complex Number

<sup>۲</sup>Real

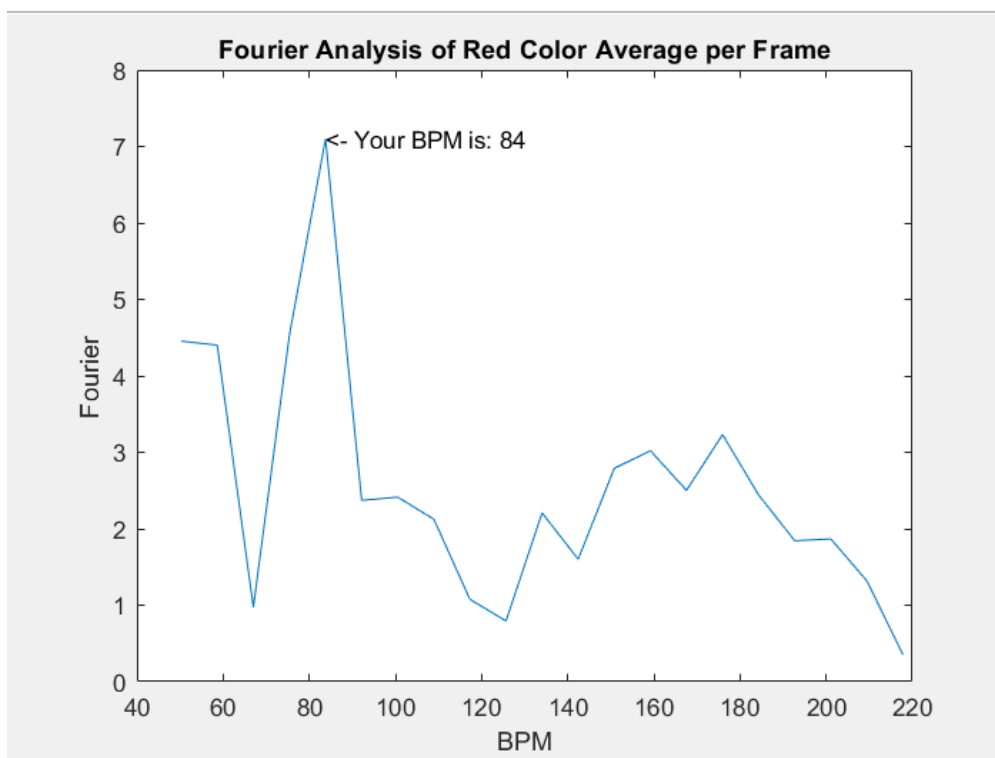
## نمایش ضربان قلب

## بدست آوردن ضربان قلب

با استفاده از دستور `max` اندیس متناظر با بیشترین مقدار در ضرایب فوریه در محدوده `[50, 220]` در دقیقه را بدست می آوریم. مقدار متناظر با این اندیس برابر با ضربان قلب می باشد.

## مشخص کردن ضربان قلب در نمودار ضربان قلب و فرکانس

با استفاده از دستور `text` نقطه ای متناظر با ضربان قلب بدست آمده را مشخص می کنیم. نتایج حاصله در شکل ۳ قابل مشاهده می باشد.



شکل ۳- ضربان قلب بدست آمده



## نمایش سیگنال PPG بدون نویز

## حذف نویز از سیگنال

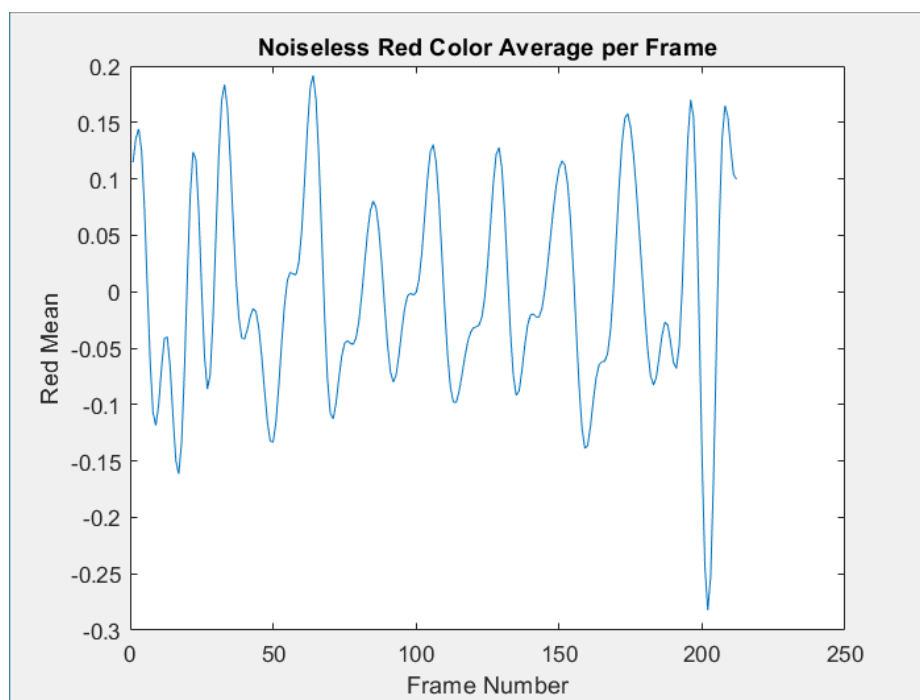
برای حذف نویز از سیگنال در این تمرین، مقادیر فرکانس های در بازه مربوط به ضربان قلب  $[50, 220]$  در دقیقه را هم در محدوده فرکانس  $[0, \frac{F_S}{2}]$  و هم در محدوده  $(0, \frac{-F_S}{2}]$  نگه می داریم و بقیه مقادیر را صفر می کنیم. نکته قابل توجه در این جا قرینه بودن محل قرار گیری فرکانس در بازه مورد نظر در هر بازه  $(0, \frac{-F_S}{2}]$  نسبت به بازه  $[0, \frac{F_S}{2}]$  می باشد.

## بدست آوردن سیگنال اصلی بدون نویز

برای بدست آوردن سیگنال اصلی در حالتی که نویز آن حذف شده است از دستور `ifft` بهره می بریم که تبدیل فوریه معکوس<sup>۱</sup> تابع مورد نظر را نتیجه می دهد.

## رسم سیگنال اصلی بدون نویز

در قسمت قبل سیگنال اصلی بدون نویز را بدست آوردیم. به کمک دستور `plot` نمودار متناظر با این سیگنال را رسم می کنیم که نتایج در شکل ۴ مشخص می باشد.



شکل ۴- سیگنال اصلی بدون نویز

<sup>۱</sup>Inverse Laplace Transform

## پیوست ۱: روند اجرای برنامه

برای اجرای برنامه این تمرین شما نیز به برنامه متلب دارید. برای اجرای این برنامه در پوشه ای که فایل P1.m موجود است فایلی تصویری با فرمت mp4 و نام A.mp4 که فیلمی چند ثانیه ای از تابش نور فلش گوشی به بافت است قرار دهید. فایل P1.m را توسط برنامه متلب باز کنید. برای اجرای کد ۳ راه موجود است:

(۱) کلیک بر گزینه Run موجود در تب<sup>۱</sup> Editor

(۲) نوشتن دستور P1 در پنجره دستور<sup>۲</sup>

(۳) استفاده از میانبر<sup>۳</sup> F5

برای اجرای برنامه بخش امتیازی نیاز به برنامه [IP Webcam](#) می باشد. پس از نصب این برنامه تنظیمات زیر را در برنامه اجرا کنید:

- IP Webcam Settings >> Video preferences -> Video resolution -> 640x480
- IP Webcam Settings >> Video preferences -> FPS Limit -> 20

هات اسپات گوشی خود را روشن کنید و سپس از بخش Service Control روی گزینه Start Server کلیک کنید. با انتخاب منو "Actions..." در برنامه و انتخاب گزینه "Toggle LED" فلش گوشی خود را روشن کنید. سپس با اجرای برنامه P1Bonus.m به اجرای برنامه متلب طبق دستورات بالا اقدام فرمایید. دست خود را به مدت تقریباً ۹ ثانیه در مقابل فلش گوشی نگه دارید تا نتایج پدیدار شوند.

---

<sup>۱</sup>Tab

<sup>۲</sup>Command Window

<sup>۳</sup>Shortcut