

۱- هدف از این سوال، مدیریت یک شکل با استفاده از کدنویسی در یک فایل متنی است.

فرض کنید سیگنالهای  $x(t)$  و  $y(t)$  به صورت زیر تعریف شده باشند:

$$x(t) = e^{-\frac{t}{2}}u(t) \quad , \quad y(t) = t^2 e^{-\frac{t}{2}}u(t)$$

الف- سیگنالهای فوق را در بازه  $0 \leq t \leq 10$  با دقت زمانی 0.01 در یک دستگاه مختصات نمایش دهید و محورهای افقی و

عمودی را به ترتیب Time و Amplitude نامگذاری کنید.

ب- توضیح دهید با اجرای کدهای زیر بعد از رسم دو منحنی با استفاده از دستور plot چه تغییری در شکل ایجاد می‌شود؟

```
[~, hobj, ~, ~]=legend({'Input','Output'},'FontSize',12,'Location','Northeast');  
hl = findobj(hobj,'type','line');  
set(hl,'LineWidth',1.5);  
ht = findobj(hobj,'type','text')  
set(ht,'FontSize',14);
```

ج- با الهام از کدهای فوق، نوع فونت legend را به times، اندازه فونت را به ۱۴، موقعیت آن را در مرکز شکل، ضخامت خطها را

۲ و اسم آنها را signal1 و signal2 تغییر دهید.

د- مجموعه کدهای زیر، چه تغییری در شکل ایجاد میکند؟

```

h0 = gca;
set(h0,'xtick',[0 5 10],'ytick',.5*(0:6))
set(h0,'ylim',[0, 2.3])
set(h0,'fontsize',12,'fontname','times','fontweight','bold')

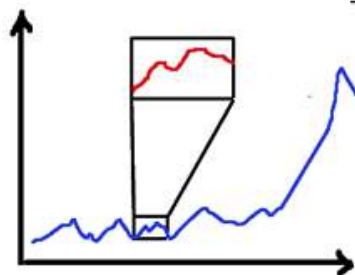
set(get(h0,'xlabel'),'string','Time (s)','fontsize',16,'fontname','helvetka','fontweight','normal')
set(get(h0,'ylabel'),'string','Amplitude (V)','fontsize',16,'fontname','helvetka','fontweight','normal')
h1 = findobj(h0,'type','line');
col=eye(3);

for j=1:length(h1)
    set(h1(j),'color',col(j,:), 'linewidth',2)
end

```

ه- با تغییر در کدهای فوق، رنگ منحنیها را بترتیب سیاه و سبز یا ضخامت ۱ و ۱٫۵ قرار دهید. ضمناً شماره فونت اعداد روی محورها را ۱۴ قرار دهید.

و- در ادامه به شکل اصلی، میخواهیم بخشی از شکل را به صورت بزرگنمایی شده اضافه کنیم (مانند مثال زیر). برای منحنیهای مساله، بازه زمانی (1,3) به صورت بزرگنمایی شده به شکل اضافه کنید. این کار را با کمک کدهای متنی انجام دهید.



- ۲- سیگنال ECG پیوست را بخوانید. سیگنال فوق با نرخ ۳۶۰ هرتز نمونه‌برداری شده است.
- الف- سیگنال را رسم کنید بگونه‌ای که محور افقی، زمان بر حسب ثانیه باشد. مشاهده میشود که سیگنال با دو پدیده نویز سفید و جابجایی خط پایه، تخریب شده است. در ادامه، هدف این است که سیگنال را بهبود دهیم.
- ب- یک فیلتر پایین گذر باترورث مرتبه ۱۲ با تابع butter و فرکانس قطع ۳۰ هرتز ( $wn=0.17$ ) طراحی کنید و پاسخ فرکانسی فیلتر را رسم کنید. (از فانکشن butter میتوانید استفاده کنید).
- ج- فیلتر فوق را روی سیگنال ECG اعمال کنید و نتیجه آن را در پنجره‌ای به طول ۴ ثانیه نمایش دهید. در شکل سیگنال اصلی و سیگنال فیلتر شده را همزمان نمایش دهید. بعلاوه برای اجرای فیلتر از دستور filter استفاده کنید و نتیجه آن را با دستور filtfilt مقایسه کنید. (مجموعاً دو شکل که هر کدام دو منحنی را نشان میدهد).

و- سیگنال  $z[n]$  را که به صورت  $z[n] = y[n] * y[-n]$  تعریف شده، محاسبه و و در بازه زمانی (0,12) رسم کنید.

ز- یک روش ساده برای محاسبه تقریبی نرخ ضربان قلب، تعیین موقعیت زمانی اولین قله بعد از لحظه صفر در سیگنال  $z[n]$  است. برنامه ای بنویسید که با کمک  $z[n]$  نرخ ضربان قلب را بصورت پیوسته باتأخیر حداکثر ۴ ثانیه گزارش کند. در این حالت لازم است  $y[n]$  را در پنجره های زمانی پیوسته به طول ۴ ثانیه در محاسبه  $z[n]$  دخالت دهید (یعنی سیگنال را باید در بلوکهایی به طول ۴ ثانیه در نظر بگیرید و برای آن کانولوشن را مانند آنچه در بند قبل دیدید انجام دهید و سپس بلوک ۴ ثانیه ای بعدی و ...).

۳- یکی از کاربردهای تبدیل فوریه، فشردگی سیگنال است. در این مثال، فشردگی سیگنال صوتی را بررسی می کنیم.

الف- سیگنال music پیوست، به فرمت wav است که یک فرمت ابتدایی برای سیگنالهای صوتی است. آن را با کمک تابع `audioread` (یا در نسخه های قدیمیتر متلب با تابع `wavread`) بخوانید. داده خوانده شده، شامل دو داده صوتی برای پخش استریو است که ما همان داده اول را مورد مطالعه قرار میدهیم. داده اول را بر حسب زمان رسم کنید.

ب- داده را اجرا (`play`) کنید و گوش دهید. (از دستورات `audioplayer` و `play` میتوانید استفاده کنید).

ج- تبدیل فوریه داده فوق را با کمک دستور `fft` حساب کنید و اندازه آن را نمایش دهید.

د- تنها به تعداد ۳۰۰۰۰ نمونه اول تبدیل فوریه فوق را نگهدارید و بقیه را دور بریزید. حال با داشتن این ۳۰۰۰۰ نمونه، سیگنال صوتی را بازیابی کنید (با دستور `ifft` و توجه کنید که چون سیگنال حقیقی است، تقارن هرمیتی تبدیل فوریه، به شما اجازه میدهد ۳۰۰۰۰ نمونه آخر تبدیل فوریه را هم حساب کنید). سیگنال بازیابی شده را در متلب اجرا کنید و گوش دهید (توجه: برای اجرا، ابتدا لازم است با دستور `audiowrite` داده را به فرمت wav ذخیره کنید و سپس با `audioread` و `play` اجرا کنید).

ه- بند د را با ۶۰۰۰۰ نمونه تکرار کنید و تفاوت کیفیت صوت را مقایسه کنید.

و- بجای تبدیل فوریه، از تبدیل کسینوسی با دستور `dct` استفاده کنید و مراحل "د" و "ه" را تکرار کنید. در این حالت و برای مقایسه باید به ترتیب ۶۰۰۰۰ و ۱۲۰۰۰۰ نمونه اول تبدیل فوریه کسینوسی را حفظ کنید.