



ارتعاشات پیشا و کاربرد آن در مهندسی زلزله

در نرم افزار های

MatLab و SeismoSignal

جلسه چهارم: پردازش و محاسبه
پارامترهای شتابنگاشت در MatLab
(ویرایش ۲)



مقدمه ای بر نرم افزار MatLab

2

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

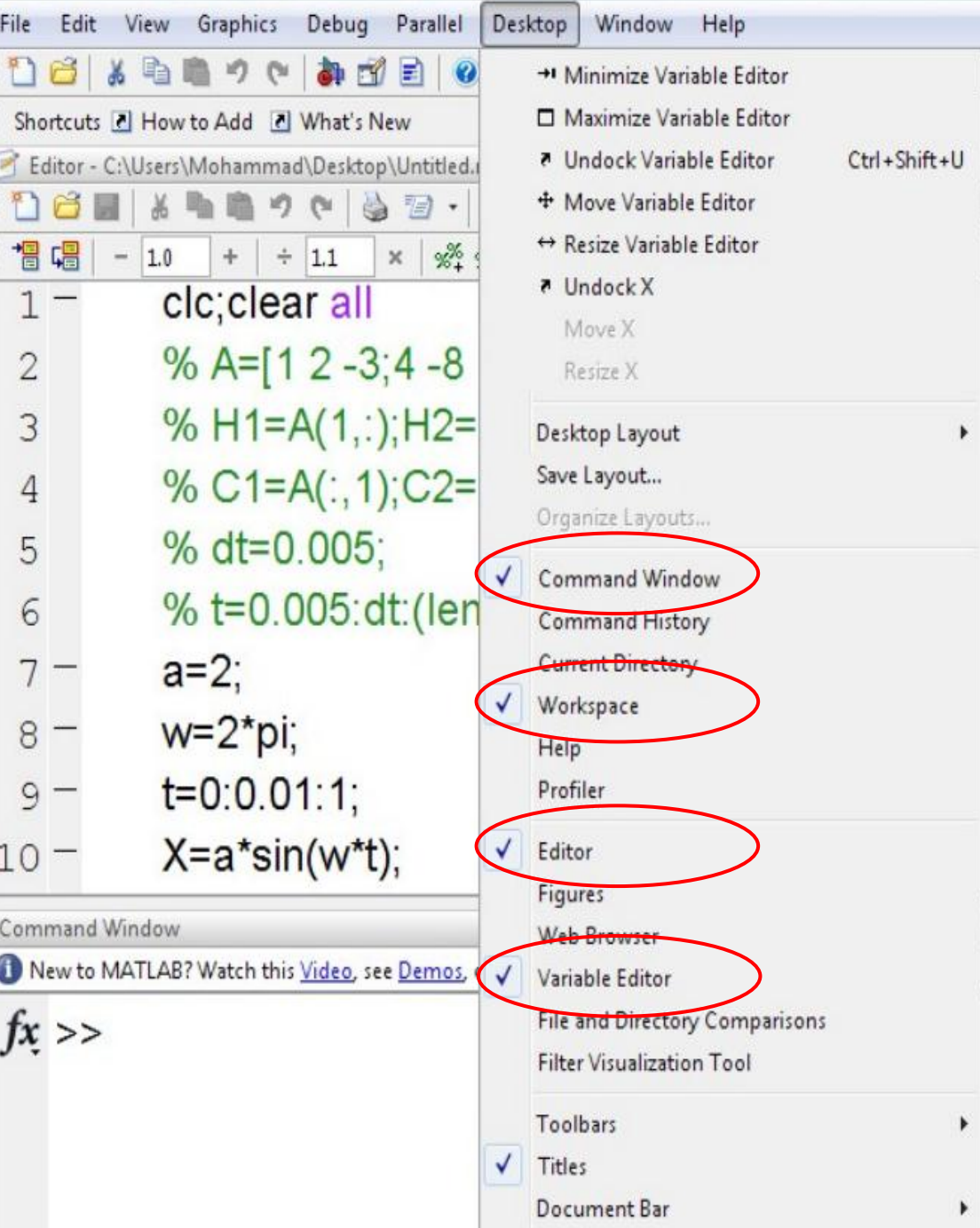
□ **MATLAB** یک محیط نرم افزاری برای انجام محاسبات عددی و یک زبان برنامه نویسی نسل چهارم است. واژه متلب هم به معنی محیط محاسبات رقمی و هم به معنی خود زبان برنامه نویسی مربوطه است که از ترکیب دو واژه **MATrix** و **LABoratory** ایجاد شده است. این نام حاکی از رویکرد ماتریس محور برنامه است، که در آن حتی اعداد منفرد هم به عنوان ماتریس در نظر گرفته می شوند. کار کردن با ماتریس ها در متلب بسیار ساده است. در حقیقت تمام داده ها در متلب به شکل یک ماتریس ذخیره می شوند. برای مثال یک عدد (اسکالر) به شکل یک ماتریس 1×1 ذخیره می شود. یک رشته مانند «Whale is the biggest animal» به شکل ماتریسی با یک سطر و چندین ستون (که تعداد ستون ها به تعداد کاراکترهاست) ذخیره می شود. حتی یک تصویر به شکل یک ماتریس سه بعدی ذخیره می گردد که بُعد اول و دوم آن برای تعیین مختصات نقاط و بُعد سوم آن برای تعیین رنگ نقاط استفاده می شود. فایل های صوتی نیز در متلب به شکل ماتریس های تک ستون (بردارهای ستونی) ذخیره می شوند. بنابراین جای تعجب نیست که متلب مخفف عبارت آزمایشگاه ماتریس باشد.



مقدمه ای بر نرم افزار MatLab

□ شرکت سازنده متلب شرکت Mathworks نام دارد. این شرکت در سال ۱۹۸۴ بنیان نهاده شد و هم اکنون دارای بیش از ۲۰۰۰ نفر پرسنل است. دفتر مرکزی این شرکت در ایالت ماساچوست آمریکا قرار دارد. برای دسترسی به سایت این شرکت می توان به آدرس <http://www.mathworks.com> مراجعه کرد. متأسفانه این آدرس برای ایرانیان بسته است و برای وارد شدن به سایت این شرکت باید از برنامه های تغییر IP استفاده کرد. هسته متلب برای سرعت و کارایی بالا به زبان سی نوشته شده است ولی رابط گرافیکی آن به زبان جاوا پیاده سازی گشته است.

جانمایی پنجره ها



□ برای کنترل کدهای نوشته شده بهتر است چهار پنجره **Workspace**, **Command Window**, **Editor** و **Variable Editor** به صورت همزمان در داخل محیط اصلی نرم افزار موجود باشد. برای این منظور از منوی **Desktop** برای روشن کردن این پنجره ها استفاده می شود.



جانمایی پنجره ها در MatLab (پیشنهادی)



5

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

The image shows the MATLAB 7.8.0 (R2009a) interface with several windows and annotations. The main window is the Editor, which contains a script with the following code:

```
1 clc;clear all
2 % A=[1 2 -3;4 -8 14];
3 % H1=A(1,:);H2=A(2,:);
4 % C1=A(:,1);C2=A(:,2);C3=A(:,3);
5 % dt=0.005;
6 % t=0.005:dt:(length(H1)*dt);
7 a=2;
8 w=2*pi;
9 t=0:0.01:1;
10 X=a*sin(w*t);
```

The Command Window is at the bottom left, showing the prompt `>>`. The Workspace window is at the top right, showing the variables `X`, `Y`, `a`, `t`, and `w`. The Variable Editor window is at the bottom right, showing the data for variable `X`.

Annotations (red circles) highlight the following areas:

- The Editor toolbar (top left).
- The Command Window (bottom left).
- The Workspace window (top right).
- The Variable Editor window (bottom right).

Labels for the windows are placed over the corresponding areas:

- Command Window
- Editor
- Workspace
- Variable Editor



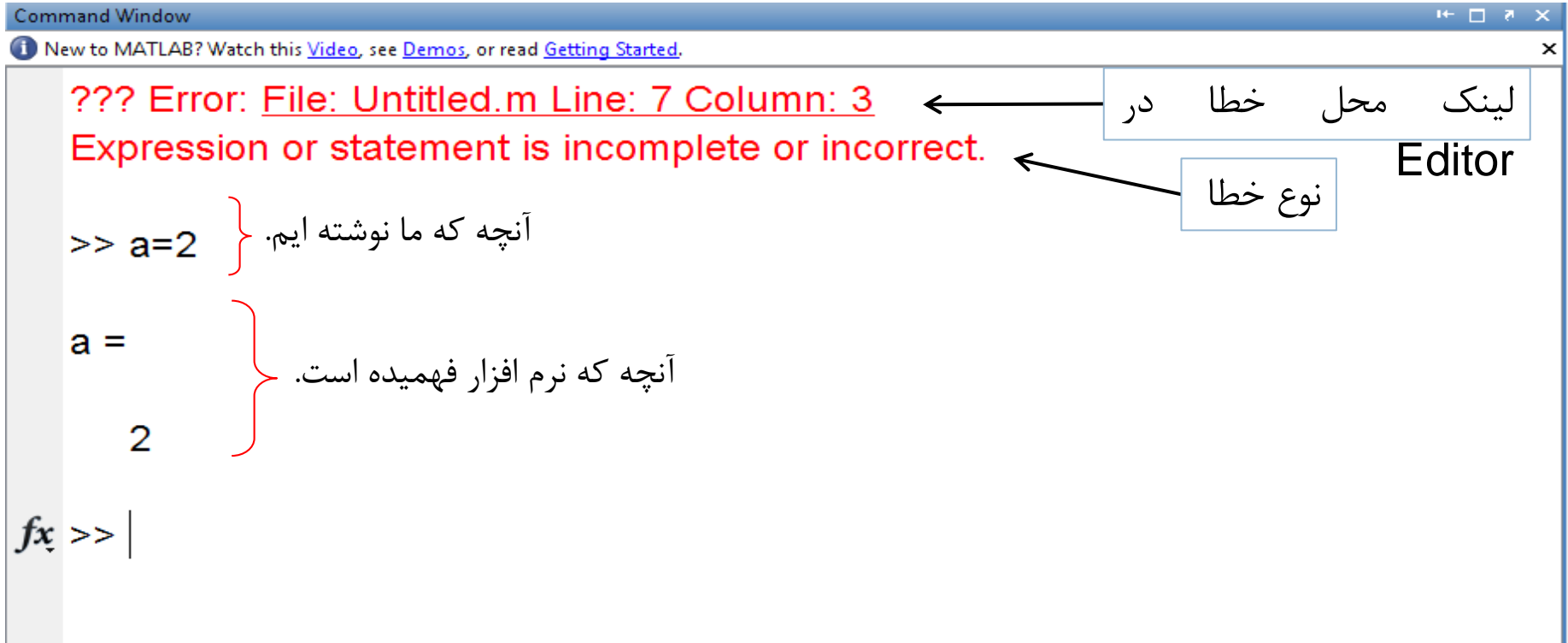
محیط Command Window



6

beheshtikhah@gmail.com

محمد حسین بهشتی خواه



با نوشتن دستورات در این پنجره، این دستورات با زدن **Enter** اجرا می شود. کاربرد دیگر این پنجره، نمایش **Error** (خطا) های موجود در دستورات نوشته شده در این پنجره یا پنجره **editor** است. یک نوع از این خطا ها در شکل نشان داده شده است. با کلیک بر روی لینک موجود در پیام، **Cursor** پنجره **Editor**، به روی خط حاوی خطا منتقل می شود. به این طریق می توان با بازبینی آن خط، خطا را اصلاح کرد. باید توجه داشت، از آنجایی که کدهای نوشته شده در **Editor** به یکدیگر مرتبط هستند، خطا در یک خط می تواند ناشی از خطا در خطوط قبلی آن باشد.



محیط Editor

7

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

The screenshot shows the MATLAB Editor window with a script file named 'Untitled.m'. The script contains the following code:

```
1 clc; clear all
2 a=2;
3 w=2*pi;
4 t=0:0.01:1;
5 X=a*sin(w*t);
6 Y=trapz(t,X);
7 plot(t,X);
8 % rasme nemoodar
```

Annotations and arrows point to specific parts of the code and the editor interface:

- An arrow points from the 'Run' button (a green play icon) in the toolbar to a box containing the text: **کلید اجرای برنامه (F5)**.
- An arrow points from the first line of code, `clc; clear all`, to a box containing the text: **کد نوشته شده شامل دستورات MatLab**.
- An arrow points from the comment line `% rasme nemoodar` to a box containing the text: **عبارت نوشته شده بعد از علامت % خوانده نمی شود، که معمولاً برای توضیح کدها نوشته می شود.**
- An arrow points from the right margin of the editor to a box containing the text: **رنگ سبز، نشانگر عدم وجود اشکال ویرایشی و رنگ قرمز، نشانگر وجود اشکال ویرایشی در متن است، که در این صورت (رنگ قرمز)، کد اجرا نخواهد شد.**

پرکاربردترین پنجره نرم افزار **MatLab**، همین پنجره است. در این پنجره دستورات لازم نوشته شده و با زدن کلید **F5** اجرا می شود. به طور کلی در این پنجره، همانند نرم افزار **Word** می توان هر متنی را نوشت (حتی فارسی). ولی برای اجرای کد، این متون تنها باید شامل دستورات **MatLab** باشد.



محیط Workspace



8

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

Workspace				
Stack: Base				
Name	Value	Min	Max	
X	<1x101 double>	-2	2	
Y	2.6563e-17	2.6563e...	2.6563...	
a	2	2	2	
t	<1x101 double>	0	1	
w	6.2832	6.2832	6.2832	

در صورتی که یک کد نوشته شده در **Commad Window** یا **Editor**، اجرا شود، نام، مقدار و نوع تمامی متغیرهای موجود در آن کد، در پنجره **Workspace** نمایش داده می شود. با دوبار کلیک روی نام متغیرها می توان مقادیر آن ها را با جزئیات بیشتر در پنجره **Variable Editor** مشاهده کرد.



محیط Variable Editor

9

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

این پنجره به خودی خود وجود نداشته و با کلیک بر روی هریک از متغیرهای موجود در پنجره Workspace باز می شود، که محیطی شبیه نرم افزار Excel دارد. می توان داده ها را از نرم افزار Excel نیز به این محیط وارد کرد.



خواندن و نمایش خروجی رکورد خام در MatLab



11

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- `clc`
محیط **Command Window** را پاک می کند تا پیغام های خطای مربوط به کد جدید (در صورت وجود) از پیغام های قبلی جدا شود.
- `clear all`
تمامی متغیر های موجود در محیط **Workspace** را پاک می کند، تا کد جدید از متغیرهای قبلی استفاده نکند.
- `close all`
تمامی شکل (**Figure**) های رسم شده از کدهای قبلی را می بندد، تا شکل های کد جدید روی شکل های قبلی رسم نشود.
- `fid = fopen('J:\Guilan University\SSA2\2749-1.V1');`
فایل خروجی دستگاه شتابنگار با فرمت **V1** را از آدرس **J:\Guilan University\SSA2** باز کرده و در متغیر **fid** ذخیره می کند. (به وجود 'دقت کنید!')
- `a = textscan(fid,'%f',12046,'headerlines',27);`
فایل خروجی شتابنگار را که در متغیر **fid** ذخیره شده بود را به صورت سطر به سطر (همانند قانون نام گذاری درایه ماتریس ها) می خواند. در این خواندن، از ۲۷ خط اول (**headerlines**) صرفنظر شده و ۱۲۰۴۶ عدد اعشاری (**%f**) در متغیر **a** ذخیره می شود.



خواندن و نمایش خروجی رکورد خام در MatLab



12

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- `A=a{1};`
- متغیر `a` دارای فرمت `Cell` است (مخصوص `MatLab`) است، که با این دستور تبدیل به فرمت عادی شده و در متغیر `A` (`Acceleration`) ذخیره می شود.
- `dt=0.005; t=0:dt:(length(A)-1)*0.005;`
- به منظور رسم شتاب، لازم است محور زمان نیز مشخص باشد. برای این منظور بردار `t` را با گام زمانی (`dt`) `0.005` و برابر با طول بردار `A` می سازیم.
- علامت `;` (`semicolon`) باعث جلوگیری از چاپ متغیر `t` در محیط `Command Window` می شود.
- `Figure(1); plot(t,A,'-g','LineWidth',2);`
- نمودار شتاب در مقابل زمان با دستور `plot` ساخته می شود. این نمودار به ضخامت (`Linewidth`) `2` و رنگ سبز (`g`) و به صورت خط ممتد (`-`) خواهد بود که در `figure(1)` نمایش داده خواهد شد.
- `print(Figure(1),'-dbmp','unfiltered','-r200');`
- نمودار رسم شده با فرمت `bmp` و با نام `unfiltered` در آدرس `Current Directory` ذخیره می شود.
- `save ('RawData.mat', 'A','t', 'dt');`
- متغیرهای `A` و `t` (در `Workspace` ببینید!) به منظور استفاده در کدهای دیگر در فایل به نام `RawData` با فرمت `mat` (مخصوص `MatLab`) در آدرس `Current Directory` ذخیره می شود. برای بازخوانی این متغیر از دستور `load` استفاده خواهد شد.



خواندن و نمایش خروجی SeismoSignal در MatLab



13

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- در صورتی که بخواهیم فایل خروجی نرم افزار SeismoSignal را در MatLab بازخوانی کنیم. متغیر A به صورت زیر در MatLab ایجاد می شود.
- یادآوری: فایل خروجی SeismoSignal معمولاً در دو ستون (زمان و شتاب) و با فرمت dat (از نوع متنی) ذخیره می شود، که خط اول فایل، نام و واحد هر ستون را نشان می دهد. بنابراین در نرم افزار MatLab باید از خط اول (توضیحات) صرفنظر کرد.
- اگر فرض کنیم نام فایل خروجی Seismo.dat باشد، خواهیم داشت:
- `fid = fopen('Seismo.dat');`
- `a = textscan(fid,'%f',20000000,'headerlines',1);`
- خط اول فایل، خوانده نشده و تمام اعداد، همانند قانون نام گذاری درایه ماتریس ها، خوانده شده و در a ذخیره می شود.
- `a=a{1};`
- `A = a(2:2:length(a))';`
- `t = a(1:2:length(a))';`
- از آنجایی که a شامل زمان و شتاب است، لازم است این دو از یکدیگر جدا شود. در ماتریس سطری a، درایه اول زمان و درایه دوم شتاب است، که به همین صورت تا آخر تکرار می شود (درایه سوم زمان و درایه چهارم شتاب است). لذا برای A درایه دوم انتخاب شده و با گام ۲ تا آخر خوانده می شود. برای t درایه اول انتخاب شده و با گام ۲ تا آخر خوانده می شود. به این ترتیب ماتریس a تبدیل به دو ماتریس A و t می شود. علامت ' (quote یا کُتیشن) برای محاسبه ترانپو (transpose) (تغییر جای سطر و ستون) ماتریس استفاده می شود.



یادآوری: فیلتر

14

beheshtikhah@gmail.com

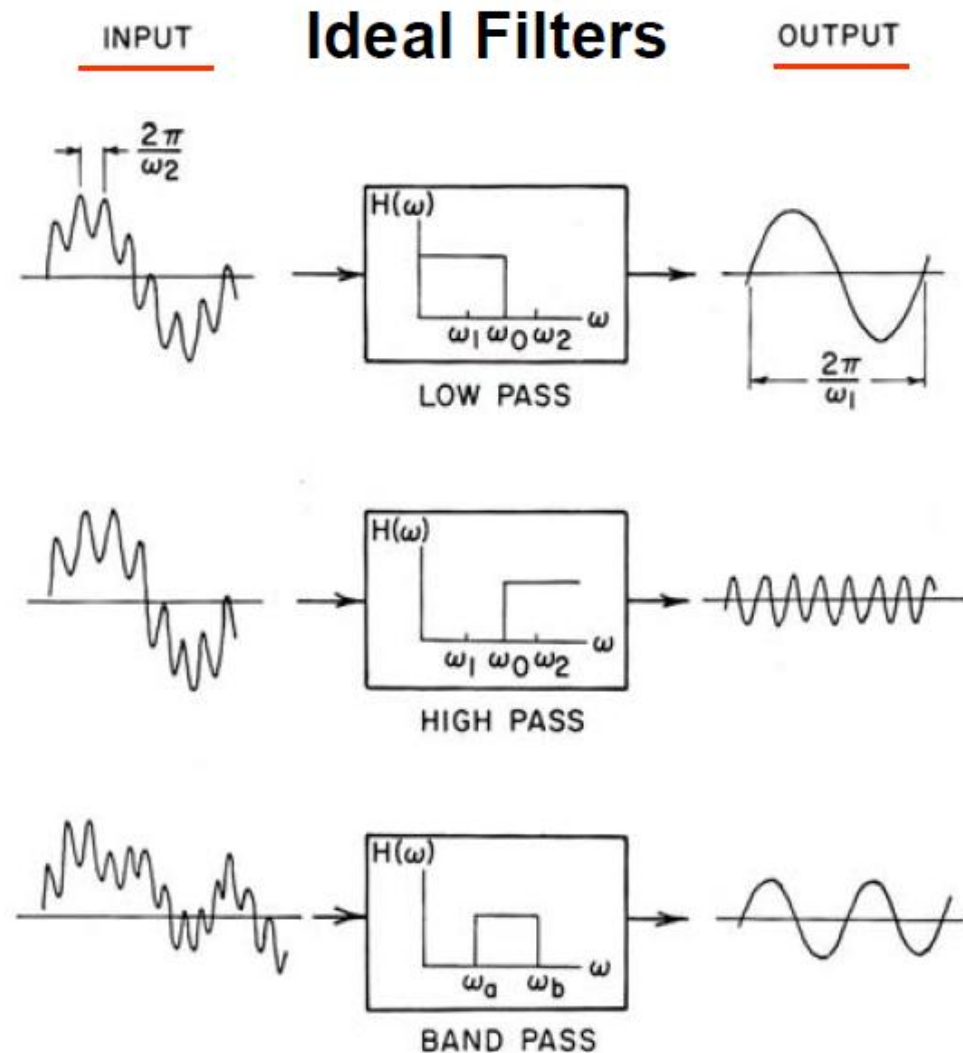
محمد حسین بهشتی خواه

- فیلترینگ، یعنی حذف مولفه هایی از شتابنگاشت که دارای فرکانس مشخصی هستند. این مولفه ها، بسته به کاربرد شتابنگاشت، می تواند نویز های محیطی یا حتی مولفه هایی از خود زلزله باشد. مثلاً در صورتی که بخواهیم این شتابنگاشت ها را به یک مدل عددی اعمال کنیم، بسته به اندازه مش بندی نیاز به حذف فرکانس هایی از زلزله داریم.
- برای مقاصد مهندسی، عمل فیلترینگ معمولاً با نرم افزار SeismoSignal انجام می گیرد. این کار در نرم افزار MatLab نیز با استفاده از توابع آماده، قابل انجام است که در ادامه بحث می شود.
- بعد از خواندن رکورد خام دستگاه SSA-2 در نرم افزار MatLab، عمل فیلترینگ (filtering) روی این شتابنگاشت انجام می شود.

☐ fdesign.lowpass

☐ fdesign.highpass

☐ fdesign.bandpass





طراحی فیلتر در MatLab



16

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- ❑ Filter design is the process of creating the filter coefficients to meet specific filtering requirements. Filter implementation involves choosing and applying a particular filter structure to those coefficients. Only after both design and implementation have been performed can data be filtered.
- ❑ All of the filter design functions operate with normalized frequencies, so they do not require the system sampling rate as an extra input argument.
- ❑ Nyquist frequency (folding frequency), defined as half the sampling frequency.

Harry Nyquist



Harry Nyquist (1889-1976)



فیلتر ButterWorth در MatLab



17

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- ❑ `clc;clear all;`
- ❑ `load RawData.mat`
- ❑ فایل RawData بازخوانی می شود. در این صورت متغیر های `t` و `A` و `dt` در Workspace نمایش داده خواهد شد. توجه شود که کد حاوی این دستور باید در کنار فایل RawData.mat ذخیره شده باشد.
- ❑ `h = fdesign.bandpass ('n,fc1,fc2',2,2,10,200) % design`
- ❑ ضرایب (Coefficients) یک فیلتر میانگذر (bandpass) وارد می شود. با توجه به این دستور، درجه فیلتر (n) ۲، فرکانس بالاگذر (fc1) ۲ هرتز، فرکانس پایین گذر (fc2) ۱۰ هرتز اعمال می شود. از آنجایی که بهتر است فیلتر تا فرکانس نایکوئیست رسم شود، نرخ نمونه برداری ۲۰۰ نیز می تواند در کنار ضرایب فیلتر وارد شود. در نهایت، این مقادیر در متغیر `h` به عنوان ضرایب یک فیلتر میان گذر ذخیره خواهد شد.
- ❑ `%h = fdesign.bandstop ('n,fc1,fc2',2,2,10,200) % design`
- ❑ یک فیلتر bandstop با مشخصات قبلی.
- ❑ `%h = fdesign.lowpass ('n,fc',2,10,200) % design`
- ❑ یک فیلتر پایین گذر تنها به یک فرکانس (پایین گذر) نیاز خواهد داشت.
- ❑ `%h = fdesign.highpass ('n,fc',2,2,200) % design`
- ❑ یک فیلتر بالا گذر تنها به یک فرکانس (بالا گذر) نیاز خواهد داشت.
- ❑ `Hd = design(h, 'butter'); % implementation`
- ❑ ضرایب فیلتر `h`، به نوع باترورث (Butterworth) اختصاص داده شده و این مجموعه در متغیر `Hd` ذخیره می شود.



فیلتر ButterWorth در MatLab



18

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- ❑ `outButt = filter(Hd,A); % outButt = out put % filtration`
❑ حال، فیلتر Hd به شتابنگاشت A (که قبلاً با دستور load به Workspace وارد شده) اعمال شده و نتیجه آن، یعنی شتابنگاشت فیلتر شده، در متغیر outButt ذخیره می شود.
- ❑ `figure(1); plot(t,outButt,'--b','LineWidth',2);grid on;`
❑ نمودار شتابنگاشت فیلتر شده در مقابل زمان در شکل ۱ (`figure(1)`) به صورت خط چین آبی رنگ رسم می شود. برای روشن کردن گرید بندی از `grid on` استفاده شده است.
- ❑ `legend('filtered',1);`
❑ راهنما (`legend`)ی نمودار در قسمت بالا سمت راست (محل شماره ۱) به نام `filtered` به نمودار اضافه می شود.
- ❑ `title('Accel');`
❑ عنوان (`title`) نمودار به نام `Accel` به نمودار اضافه می شود.
- ❑ `xlabel('Time (Sec)'); ylabel('Accel (m/s^2)'); grid on;`
❑ به ترتیب، برچسب محور افقی و قائم نمودار به نام های `Time (Sec)` و `Accel (m/s^2)` به نمودار اضافه می شود.
- ❑ `xlim([13 20]);`
❑ به منظور نمایش بزرگتر (`zoom`) محور x را محدود به زمان ۱۳ تا ۲۰ ثانیه می کنیم.
- ❑ `print(figure(1),'-dbmp','filtered','-r200');`
- ❑ `save ('RawandFilteredData.mat','A','t','outButt','dt');`



FV Tool



19

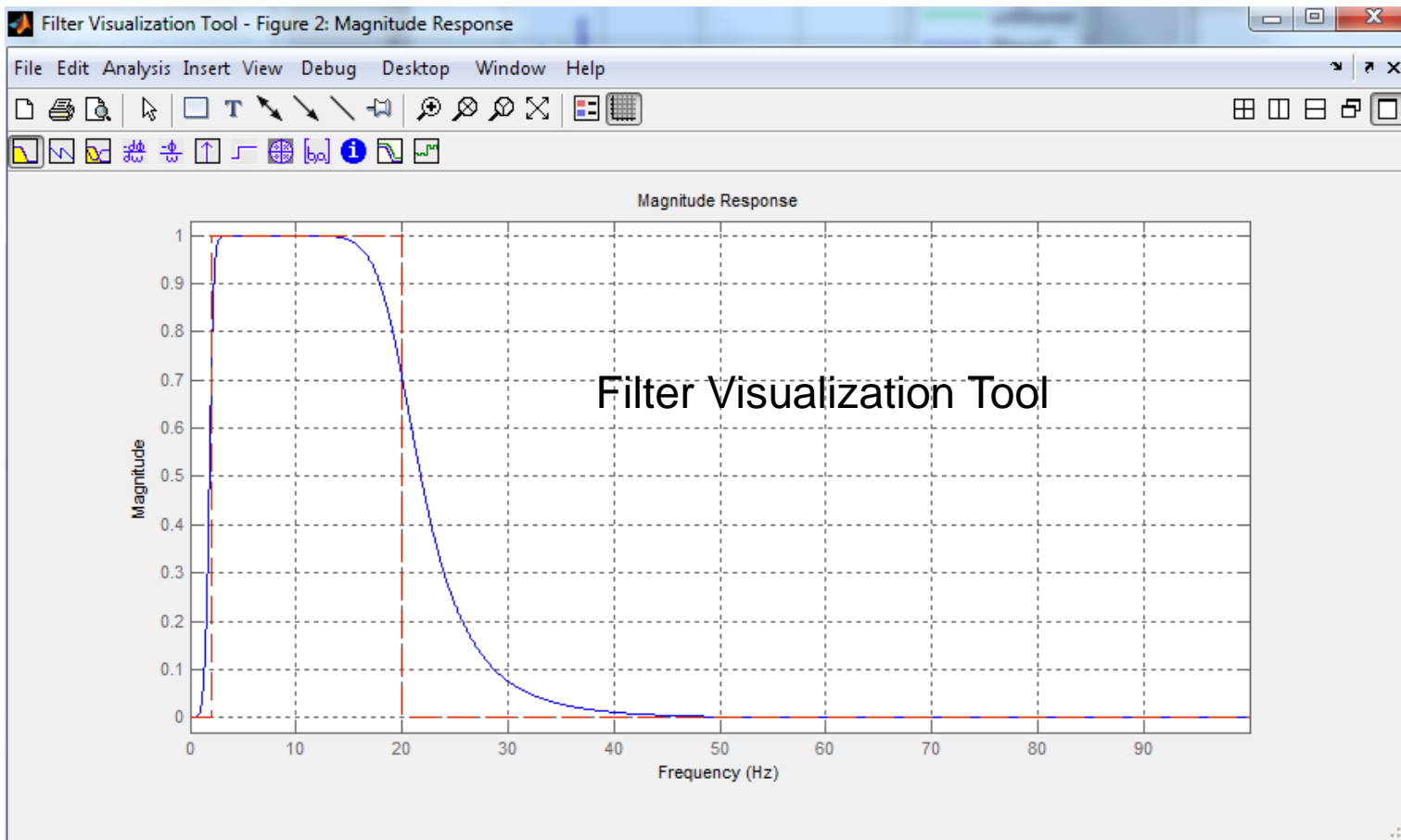
beheshtikhah@gmail.com

محمد حسین بهشتی خواه

❑ `fvtool(Hd,'MagnitudeDisplay','Magnitude');`

❑ برای نمایش فیلتر طراحی شده، از این دستور استفاده می کنیم.

❑ `print(gcf,'-dbmp','magnitude of the filter','-r200');`





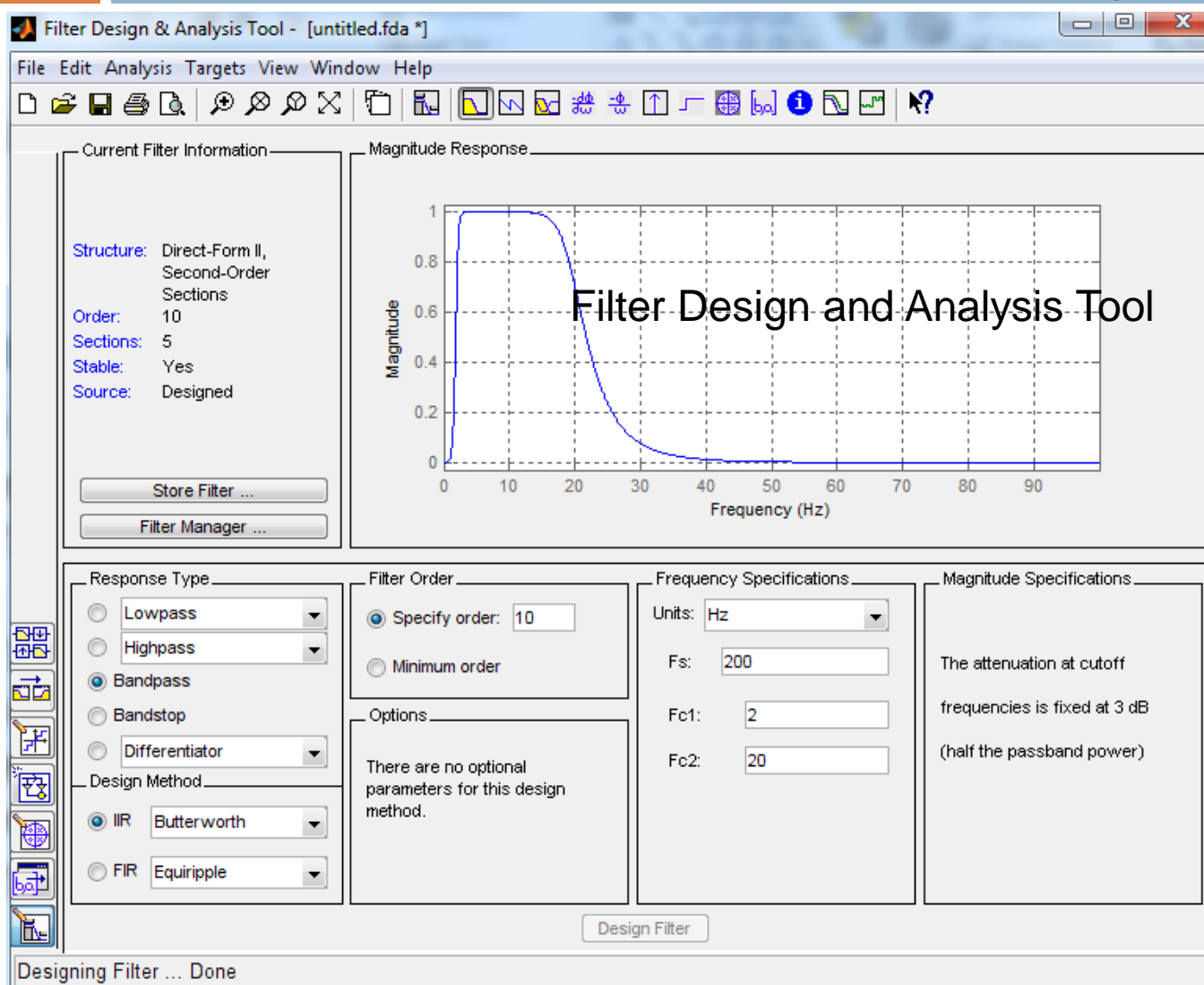
FDA Tool



20

beheshtikhah@gmail.com

محمد حسین بهشتی خواه



□ محیط گرافیکی برای طراحی فیلتر با دستور fdatool

□ رسم نمودار تا Nyquist frequency (در اینجا ۱۰۰ هرتز)

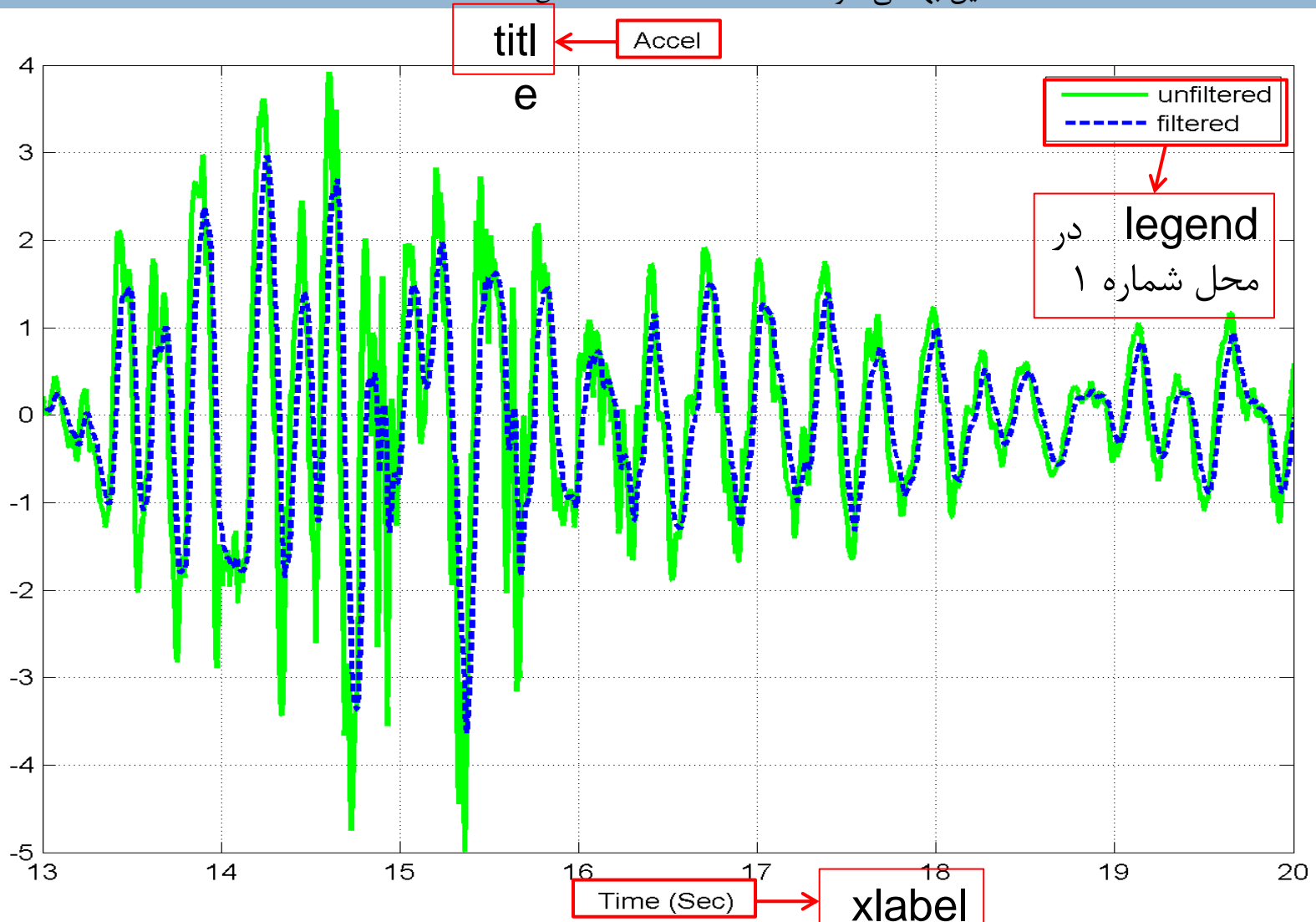


شتابنگاشت فیلتر شده با Band-Pass

21

beheshtikhah@gmail.com

محمد حسین بهشتی خواه





فیلتر پایین گذر Butterworth



22

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- `clc;clear all`
- `f=0:0.01:100;`
- `fn=20;` %Lowpass Frequency
- `n=4;` %Order
- `g=1./(1+(f./fn).^(2*n)).^0.5;` %Butterworth Lowpass Filter
- `figure(8);plot(f,g);grid on;`
- `xlabel('frequency (Hz)');ylabel('amplitude');`
- `print(figure(8),'-dbmp','Butterworth filter function','-r200');`

□ فیلتر Butterworth در نرم افزار MatLab و SeismoSignal معرفی شد. کد مربوط به نوع پایین گذر آن در نرم افزار متلب به منظور آشنایی بیشتر با این فیلتر به صورت بالا نوشته می شود.



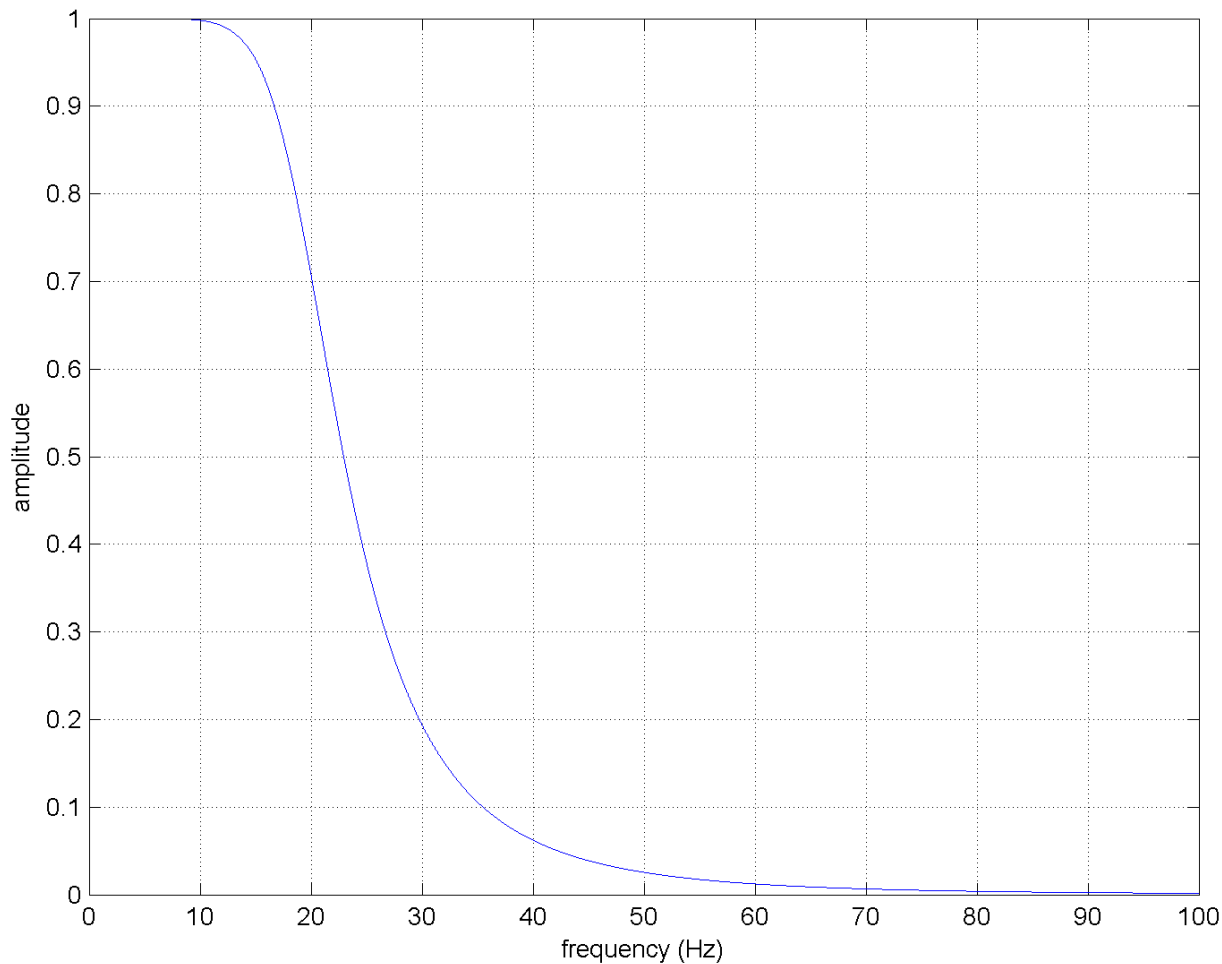
نمودار فیلتر پایین گذر Butterworth



23

beheshtikhah@gmail.com

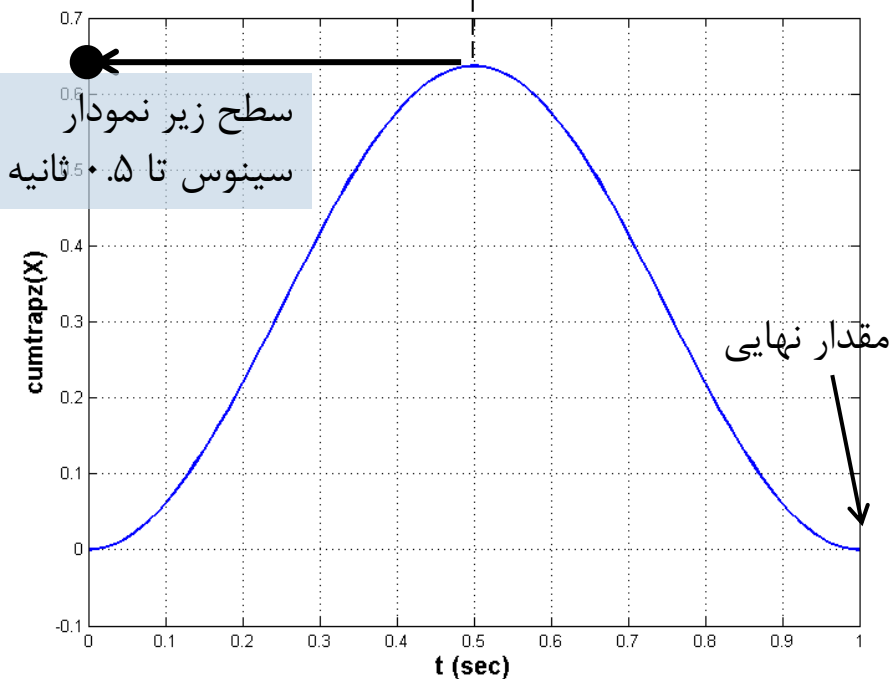
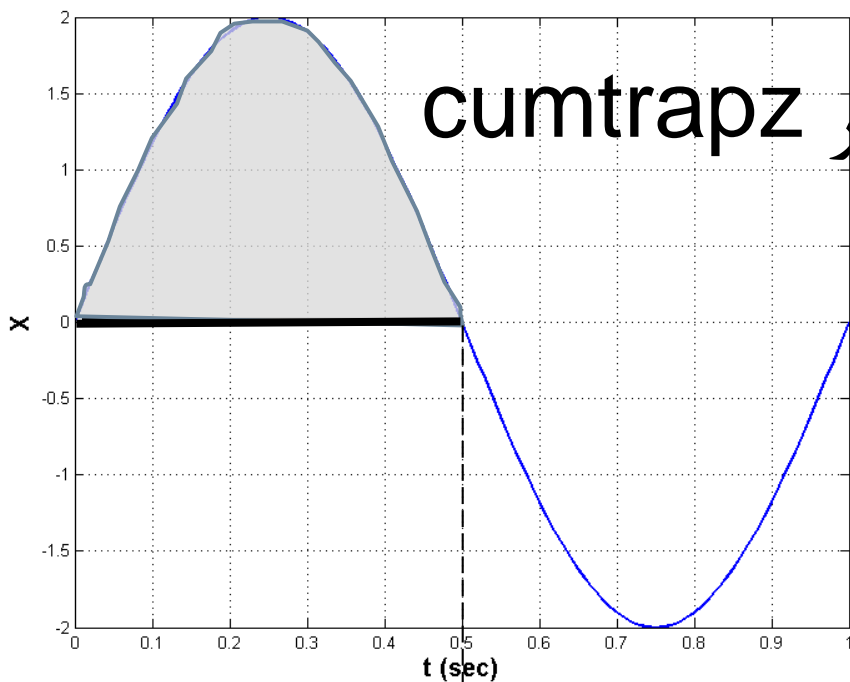
محمد حسین بهشتی خواه



□ رسم فیلتر با فرض
فرکانس نایکوئیست
۱۰۰ هرتز.

یادآوری: مفهوم دستور cumtrapz

1 محمد حسین بهشتی خواه



- دستور cumtrapz انتگرال تجمعی را به صورتی که در شکل زیر نشان داده، محاسبه می کند.
- مقدار نهایی صفر شده است، چون انتگرال یک دور کامل یک موج سینوسی صفر است.



محاسبه سرعت و جابه جایی از شتاب (با دستور cumtrapz)



25

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- `clc;clear all;close all`
- `load RawandFilteredData.mat`
- متغیرهای A و t و dt در Workspace ظاهر می شود.
- `V=cumtrapz(t,A);`
- سرعت، از سطح زیر نمودار شتاب به دست می آید.
- `D=cumtrapz(t,V);`
- جابه جایی، از سطح زیر نمودار سرعت به دست می آید.
- `figure(1);`
- `plot(t,V,'-g','LineWidth',2); grid on; xlabel('Time (Sec)'); ylabel('Velocity (m/s)');`
- `print(figure(1),'-dbmp','Vel','-r200');`
- `figure(2);`
- `plot(t,D,'-g','LineWidth',2); grid on; xlabel('Time (Sec)'); ylabel('Displacement (m)');`
- `print(figure(2),'-dbmp','Disp','-r200');`
- `save ('AllData.mat','A','outButt','t','V','D','dt');`



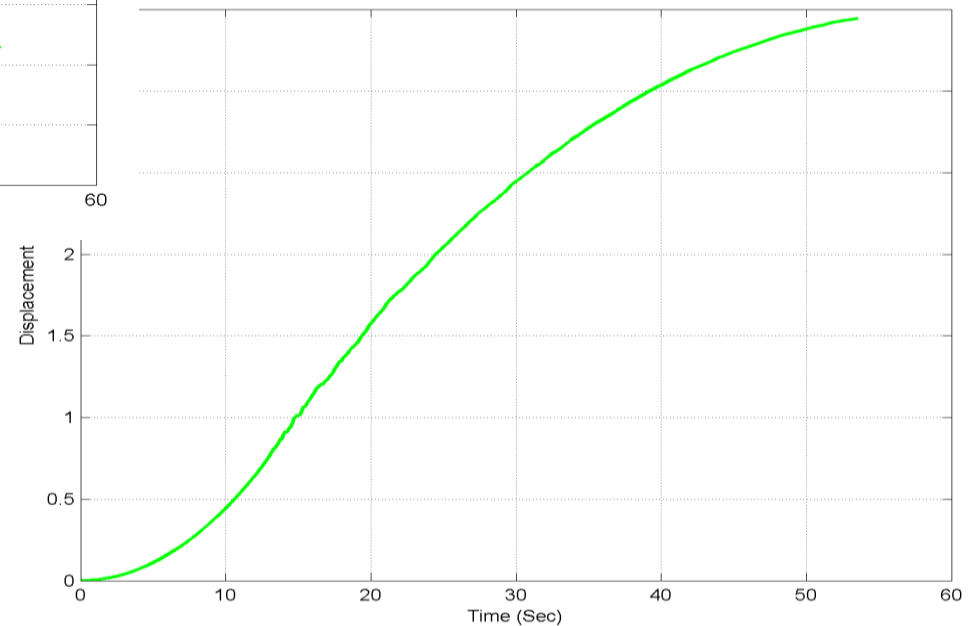
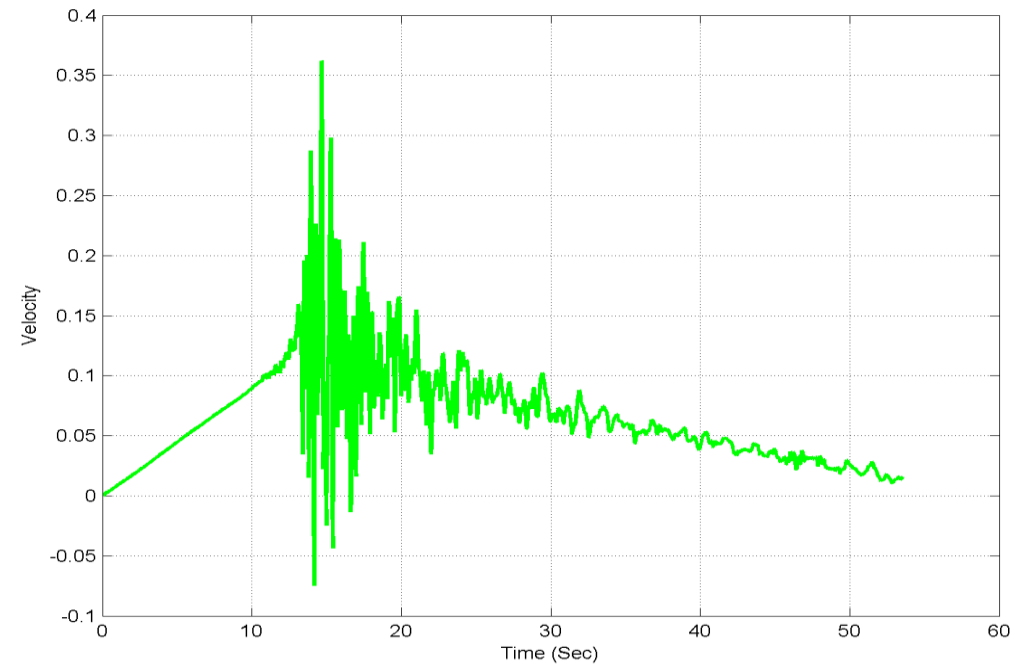
محاسبه سرعت و جابه جایی از شتاب



26

beheshtikhah@gmail.com

محمد حسین بهشتی خواه





یادآوری: مفهوم دستور find

27

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

دستور find شماره درایه هایی از یک ماتریس را که دارای مقدار مشخصی است را خروجی می دهد. کاربرد این دستور به صورت دستوراتی در Command Window نشان داده شده است.

Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> clear all; a=[1 -2 2 -7 6 8 2 4 8 0 -9 2];  
>> find(a==2)
```

ans =
3 7 12

```
>> length(ans)
```

ans =
3

fx >> |

شماره درایه هایی از ماتریس a که دارای مقدار ۲ است، چاپ می شود. بنابراین با دستور length می توان تعداد درایه های با مقدار ۲ را به دست آورد. در این صورت محیط Workspace به صورت زیر خواهد بود.

Workspace

Stack: Base

Name	Value	Min	Max
a	<1x12 double>	-9	8
ans	3	3	3



طول موثر (مدت دوام) Duration

28

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- ❑ `clc;clear all;close all`
- ❑ `load AllData.mat`
- ❑ `R=cumtrapz(t,A.^2)/max(cumtrapz(t,A.^2));`
- ❑ برای محاسبه شدت آریاس (نرمال شده به مقدار بیشینه آن) از توان دوم شتاب انتگرال گرفته شده و به مقدار بیشینه آن تقسیم می شود.
- ❑ `figure(1); plot(t,R); grid on; xlabel('Time (Sec)'); ylabel('RMS');`
- ❑ `print(figure(1),'-dbmp','Effec. Duration','-r200');`
- ❑ `ts=find(0.05<=R & R<=0.95);`
- ❑ با توجه به تعریف زمان موثر زلزله از روش Trifunac و Brady، ابتدا شماره درایه هایی از ماتریس R که مقداری بیشتر از ۰.۰۵ و کمتر از ۰.۹۵ دارند به دست آمده و در متغیر (ماتریس) ts ذخیره می شود.
- ❑ `Eff_Durat=(length(ts)+1)*dt;`
- ❑ سپس تعداد درایه های متغیر ts در گام زمانی ضرب می شود تا زمان (موثر) به دست آید. (جمع با یک اختیاری و در جهت اطمینان است)



شدت آریاس و مقادیر بیشینه

29

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

```
clc;clear all;close all
```

```
Load AllData.mat
```

```
AI=(pi/2/9.81)*trapz(t,A.^2)      %Arias Intensity
```

□ با توجه به تعریف شدت آریاس، انتگرالِ توانِ دومِ شتاب در $\pi/(2g)$ ضرب می شود. ($\pi=3.14$)

```
PGA=max(abs(A))
```

□ با توجه به تعریف PGA، ابتدا مقادیر شتابنگاشت با دستور abs مثبت شده و مقدار بیشینه آن در متغیر PGA ذخیره می شود.

```
PGV=max(abs(V))
```

```
PGD=max(abs(D))
```



مفهوم دستور linspace

30

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> s=linspace(-1,1,4)
```

```
s =
```

```
-1.0000 -0.3333 0.3333 1.0000
```

fx >> |

با توجه به این دستور، بردار S دارای ۴ درایه خواهد بود، به طوری که کمترین مقدار آن -۱ و بیشترین مقدار آن ۱ است و درایه های میانی به گونه ای محاسبه می شود تا فاصله بین تمام درایه ها برابر باشد.



FFT



31

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

- `clc;clear all;`
- `Load AllData.mat`
- `fft_A = fft(A);`

□ تبدیل فوریه ماتریس A در متغیر `fft_A` ذخیره می شود. مقادیر متغیر `fft_A` مختلط هستند (در `Workspace` ببینید!) که به صورت مزدوج یکدیگر هستند.

- `ampfft_A=abs(fft_A).*dt;`

□ از آنجایی که تبدیل فوریه یک بردار مختلط است، لازم است دامنه اعداد مختلط به دست آید. از طرف دیگر، چون این اعداد مختلط، مزدوج یکدیگر هستند، اعداد مربوط به دامنه آنها دو بار تکرار شده و در متغیر `ampfft_A` ذخیره می شود.

- `f=linspace(0,1,length(fft_A))/dt ;`

□ با دستور `linspace` محور افقی یعنی فرکانس به تعداد درایه های متغیر `ampfft_A` ساخته می شود.

- `figure(1); semilogx(f,ampfft_A);grid on; xlabel('freq (Hz)');ylabel('|FFT|');`

□ به جای `plot` از `semilogx` استفاده شده تا محور X (فرکانس) به صورت لگاریتمی رسم شود.

- `print(figure(1),'-dbmp','FFT','-r200');`



نمودار FFT



32

beheshtikhah@gmail.com

محمد حسین بهشتی خواه

