Powerline noise elimination

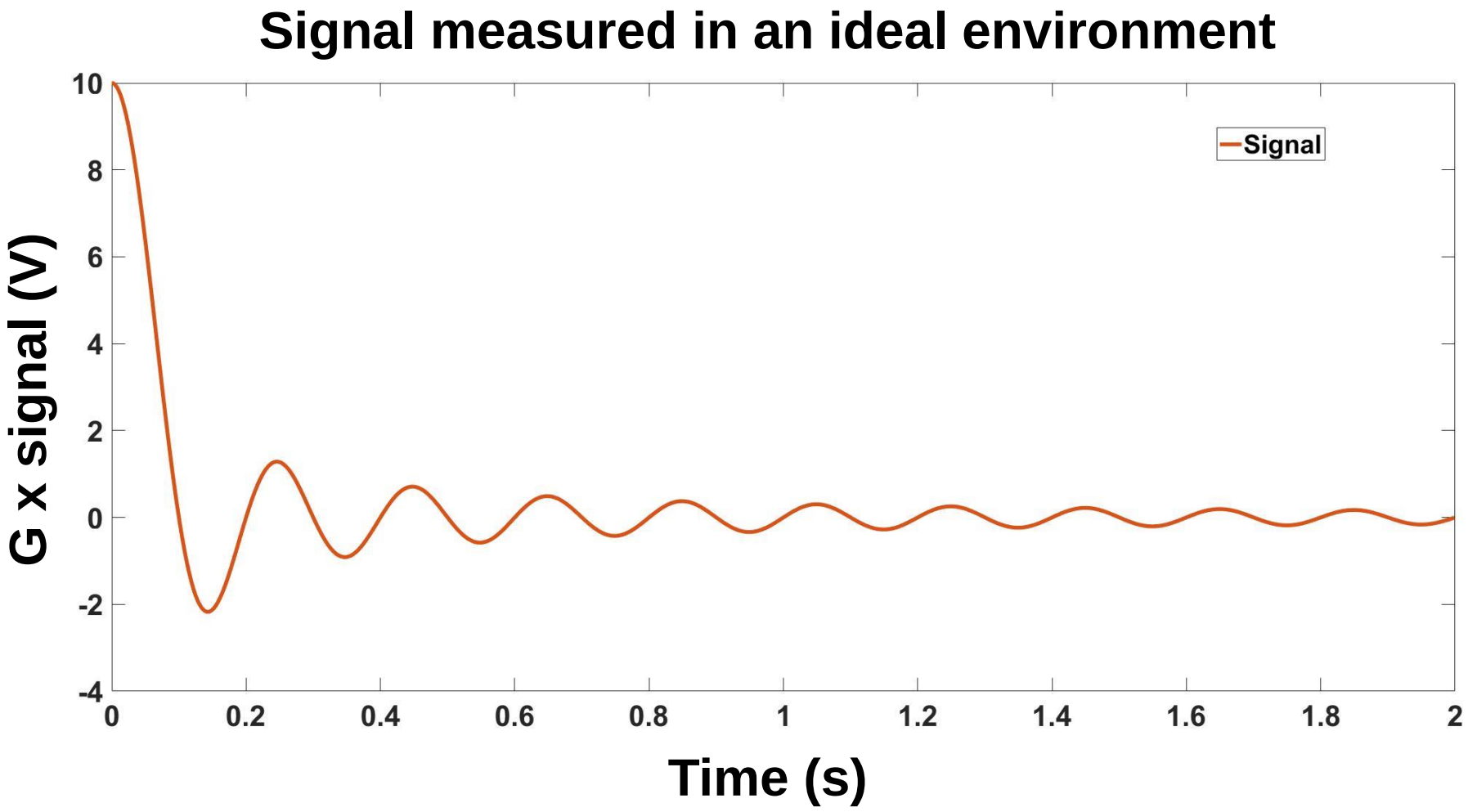
MATLAB tutorial series (Part 2.2)

Pouyan Ebrahimbabaie

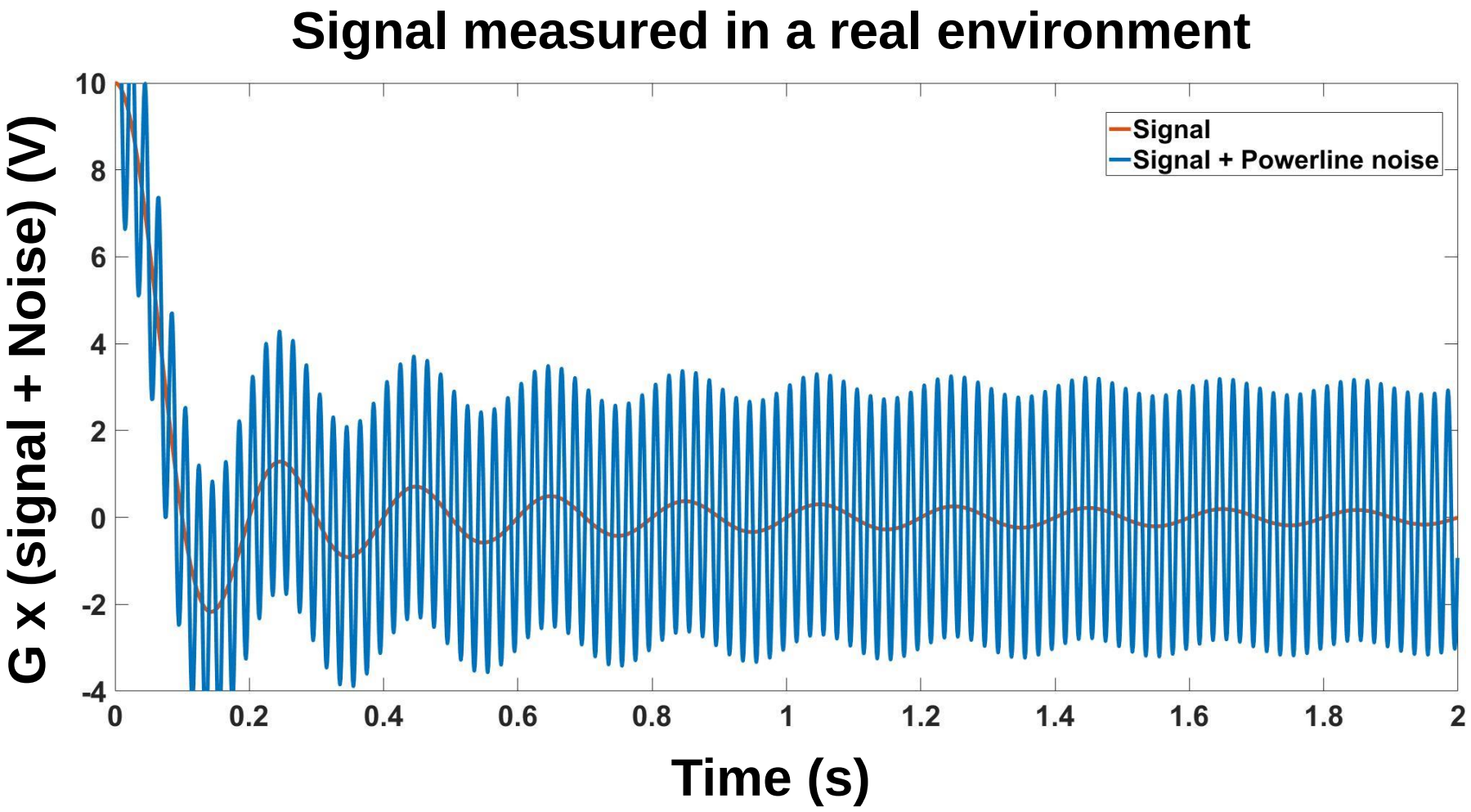
Laboratory for Signal and Image Exploitation (INTELSIG)
Dept. of Electrical Engineering and Computer Science
University of Liège
Liège, Belgium

Applied digital signal processing (ELEN0071-1)
31 March 2021

Motivation

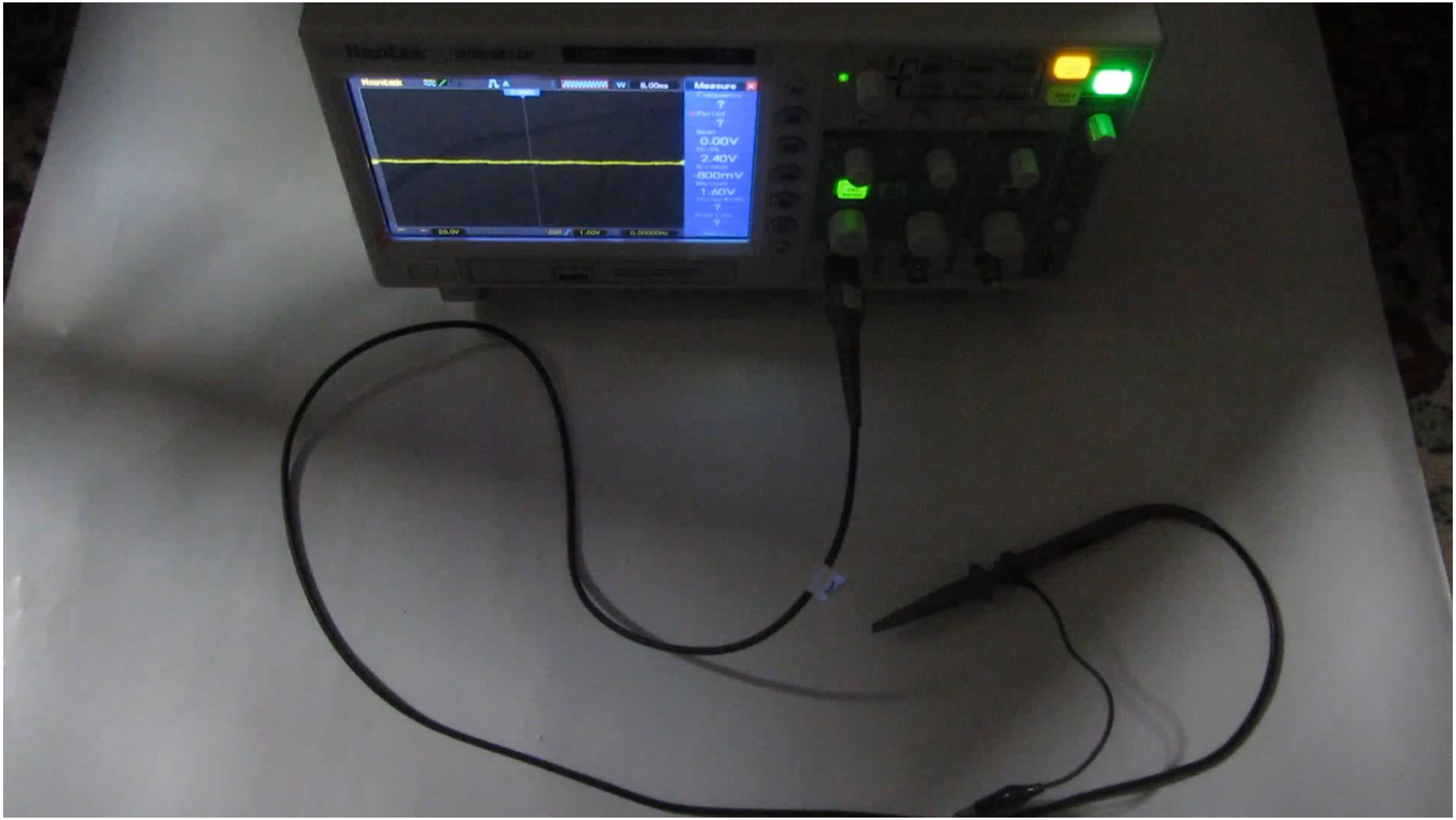

Signal measured in an ideal environment
G x signal (V)
Time (s)
Signal
10
8
6
4
2
0
-2
-4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Motivation

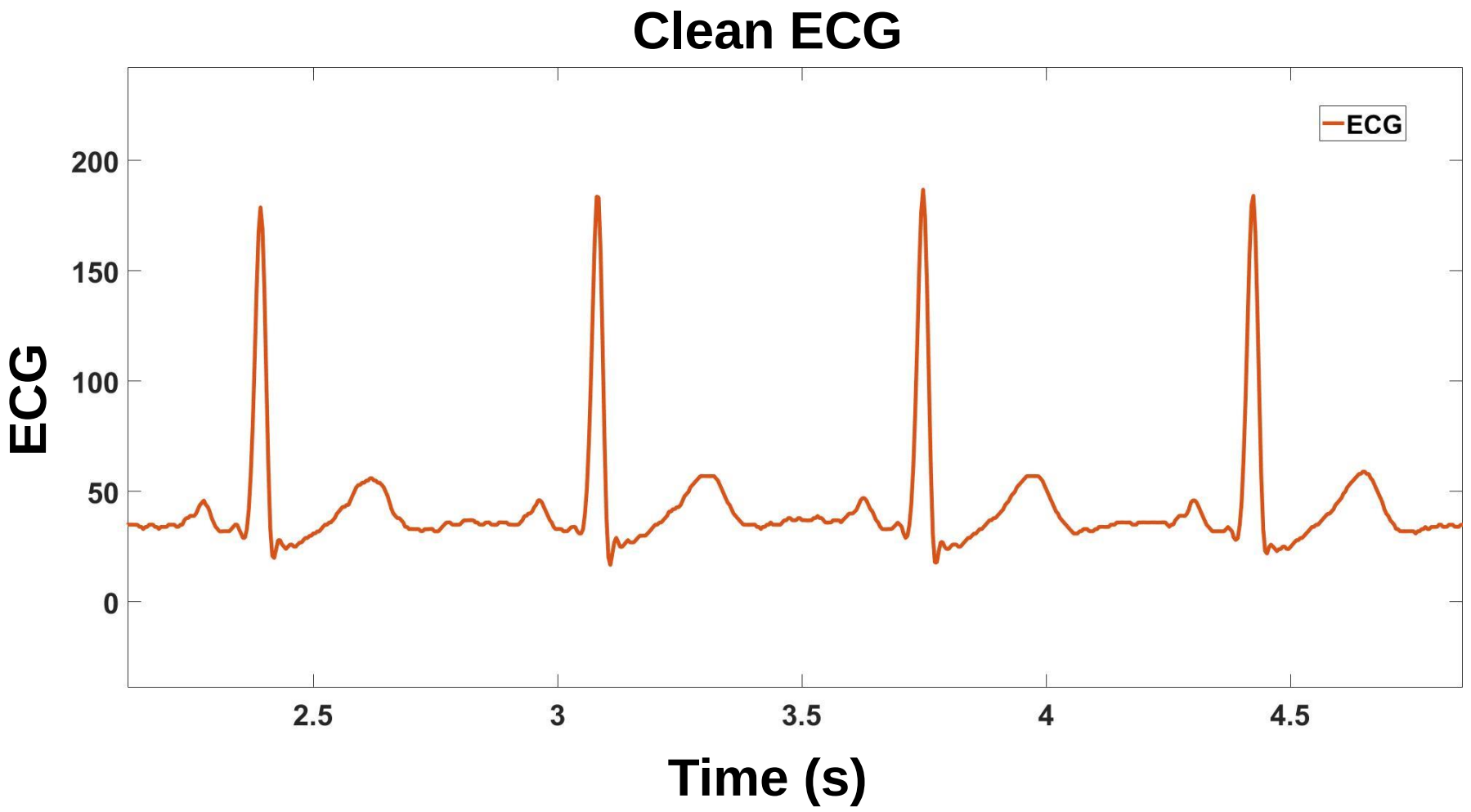

Signal measured in a real environment
G x (signal + Noise) (V)
Time (s)
Signal
Signal + Powerline noise
10
8
6
4
2
0
-2
-4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Human body as an antenna

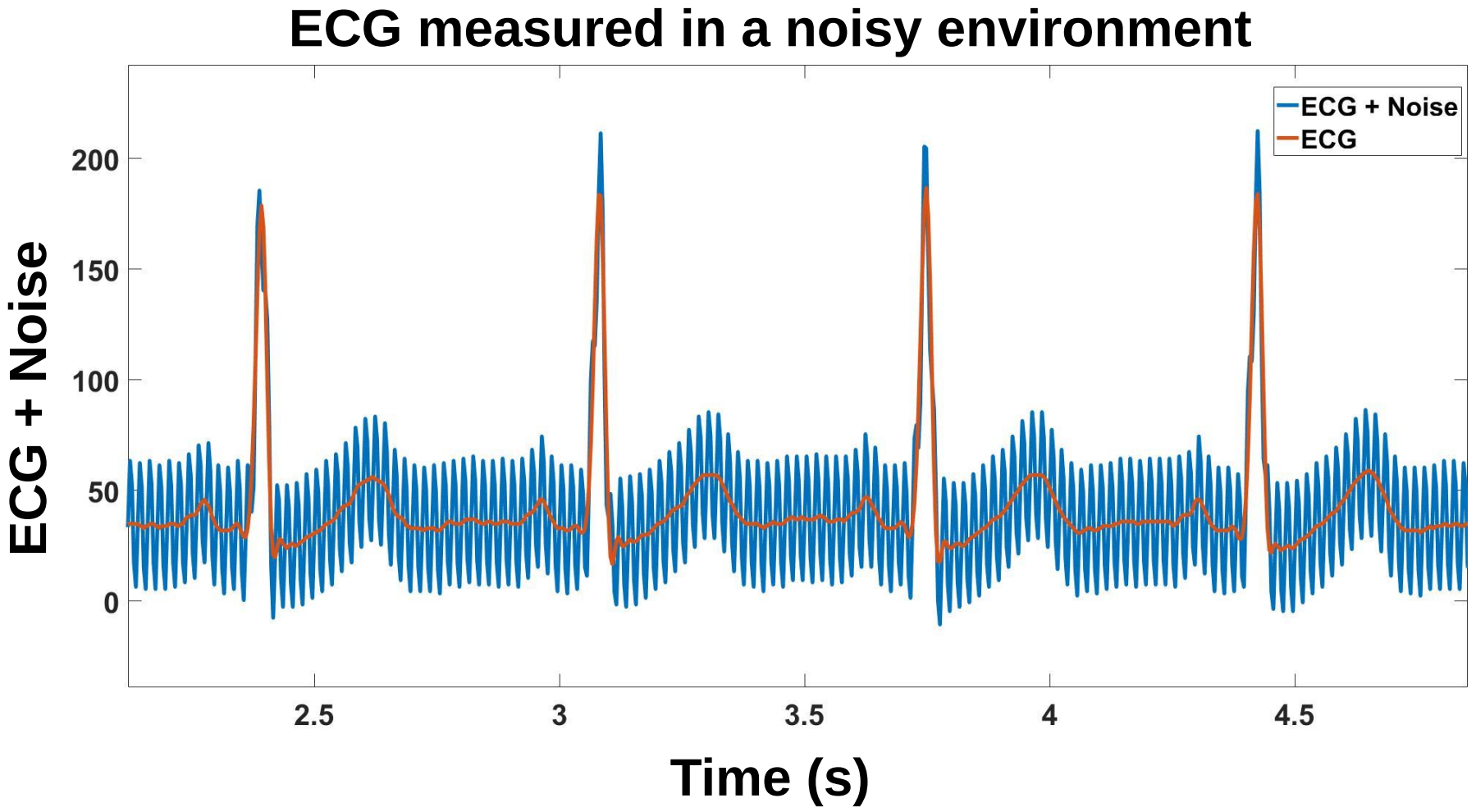
50Hz noise exists !



Electrocardiogram (ECG)



Electrocardiogram (ECG)



Powerline noise elimination

- **Step 1: plot the signal**
  - **find the sampling period (Ts)**
  - **find the ending time (Tmax)**
  - **find the length of the signal (N)**

Powerline noise elimination

- **Step 1: plot the signal**
  - **find the sampling period (Ts)**
  - **find the ending time (Tmax)**
  - **find the length of the signal (N)**
- **Step 2: plot one-sided magnitude spectrum**
  - **use fft to plot magnitude spectrum**

Powerline noise elimination

- **Step 1: plot the signal**
  - **find the sampling period (Ts)**
  - **find the ending time (Tmax)**
  - **find the length of the signal (N)**
- **Step 2: plot one-sided magnitude spectrum**
  - **use fft to plot magnitude spectrum**
- **Step 3: identify the noise frequencies**

Powerline noise elimination

- **Step 1: plot the signal**
  - **find the sampling period (Ts)**
  - **find the ending time (Tmax)**
  - **find the length of the signal (N)**
- **Step 2: plot one-sided magnitude spectrum**
  - **use fft to plot magnitude spectrum**
- **Step 3: identify the noise frequencies**
- **Step 4: filter out undesired frequency components**
  - **here we used Notch filter (i.e. Notching)**

Powerline noise elimination

- **Step 1: plot the signal**
  - **find the sampling period (Ts)**
  - **find the ending time (Tmax)**
  - **find the length of the signal (N)**
- **Step 2: plot one-sided magnitude spectrum**
  - **use fft to plot magnitude spectrum**
- **Step 3: identify the noise frequencies**
- **Step 4: filter out undesired frequency components**
  - **here we used Notch filter (i.e. Notching)**
- **Step 5: plot the results**
  - **Plot the original and filtered signals together**

Step 1: plot the signal

```
% Load ecg signal from .mat file
load('TNS_2_1_2018_example1.mat','ecg')
```

Step 1: plot the signal

```
% Load ecg signal from .mat file
load('TNS_2_1_2018_example1.mat','ecg')
% Fs is given
Fs=250;
```

Step 1: plot the signal

```
% Load ecg signal from .mat file
load('TNS_2_1_2018_example1.mat','ecg')
% Fs is given
Fs=250;
% Ts sampling period
Ts=1/Fs;
% Length of the signal
N=length(ecg);
```

Step 1: plot the signal

```
% Load ecg signal from .mat file
load('TNS_2_1_2018_example1.mat','ecg')
% Fs is given
Fs=250;
% Ts sampling period
Ts=1/Fs;
% Length of the signal
N=length(ecg);
% Ending time
Tmax=(N-1)*Ts;
% Time vector
t=0:Ts:Tmax;
```

Step 1: plot the signal

```
% Plot the original signal
figure(1)
plot(t,ecg)
xlabel('Time (s)')
title('ECG + noise')
```

Step 2: plot one-sided magnitude spectrum

```
% Compute fft
ECG=fft(ecg);
```

Step 2: plot one-sided magnitude spectrum

```
% Compute fft
ECG=fft(ecg);
% Take abs and scale it
ECG2=abs(ECG/N);
```

Step 2: plot one-sided magnitude spectrum

```
% Compute fft
ECG=fft(ecg);
% Take abs and scale it
ECG2=abs(ECG/N);
% Pick the first half
ECG1=ECG2(1:N/2+1);
```

Step 2: plot one-sided magnitude spectrum

```
% Compute fft
ECG=fft(ecg);
% Take abs and scale it
ECG2=abs(ECG/N);
% Pick the first half
ECG1=ECG2(1:N/2+1);
% Multiply by 2 (except the DC part), to compenseate
% the removed side from the spectrum.
ECG1(2:end-1) = 2*ECG1(2:end-1);
```

Step 2: plot one-sided magnitude spectrum

```
% Compute fft
ECG=fft(ecg);
% Take abs and scale it
ECG2=abs(ECG/N);
% Pick the first half
ECG1=ECG2(1:N/2+1);
% Multiply by 2 (except the DC part), to compenseate
% the removed side from the spectrum.
ECG1(2:end-1) = 2*ECG1(2:end-1);
% Frequency range
F = Fs*(0:(N/2))/N;
```

Step 2: plot one-sided magnitude spectrum

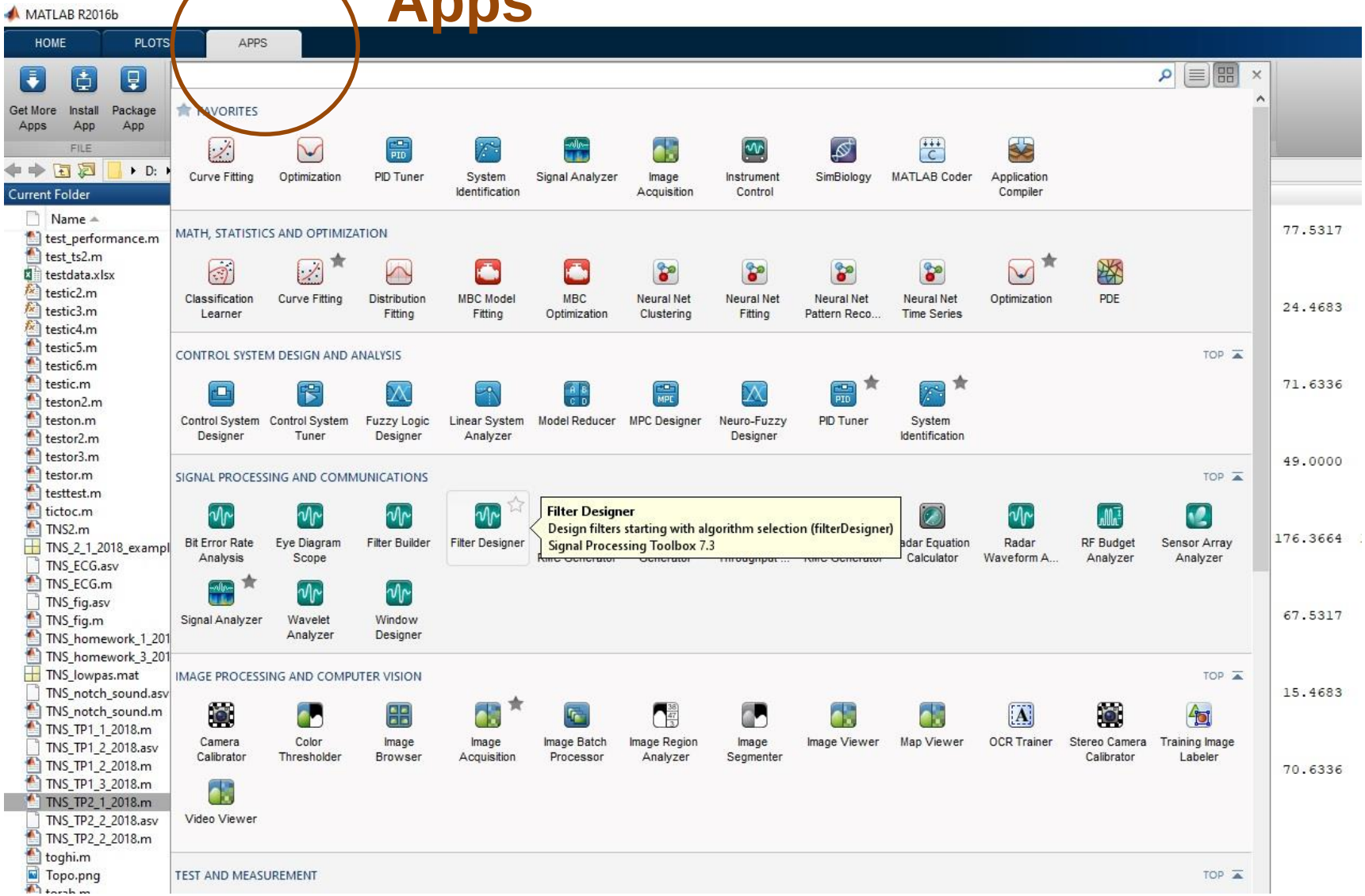
```
% Plot single-sided spectrum
figure(2)
plot(F,ECG1,'LineWidth',2.5)
title('Single-Sided Amplitude Spectrum')
xlabel('f (Hz)');
```

Step 3: identify the noise frequencies

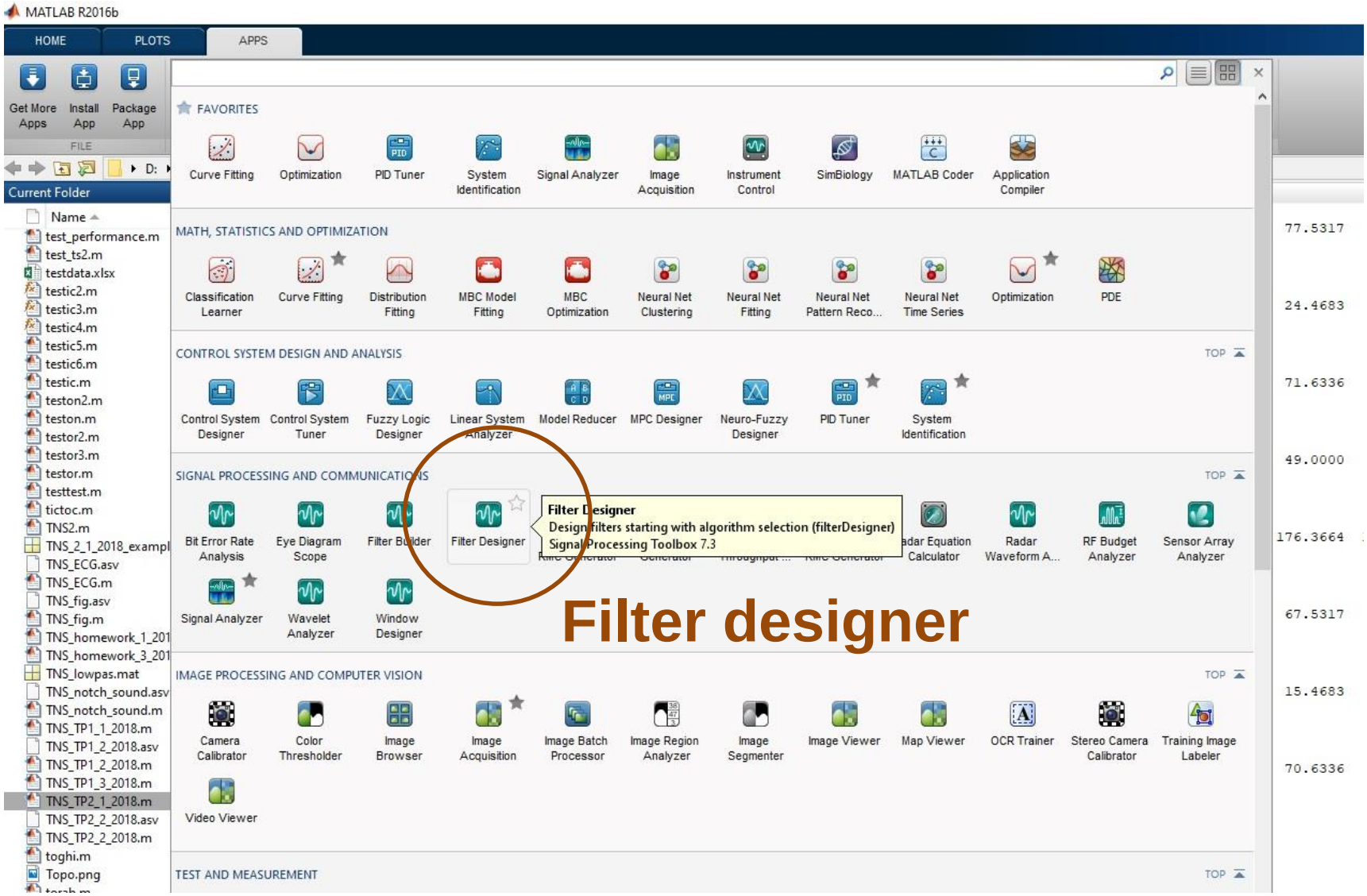
Just take a look to the single-sided magnitude spectrum and you will see …

Notch filter for dummies

Apps

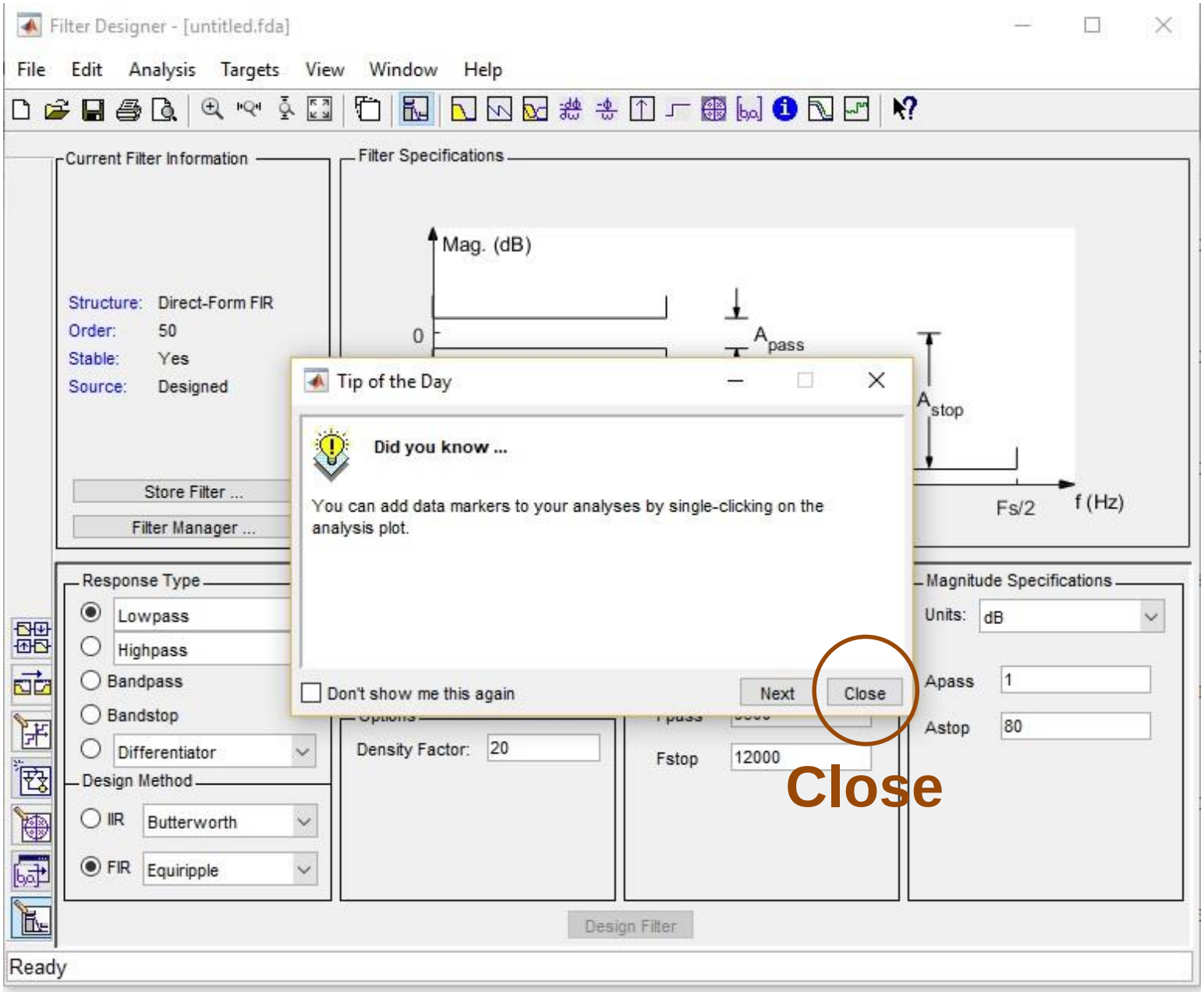


Notch filter for dummies



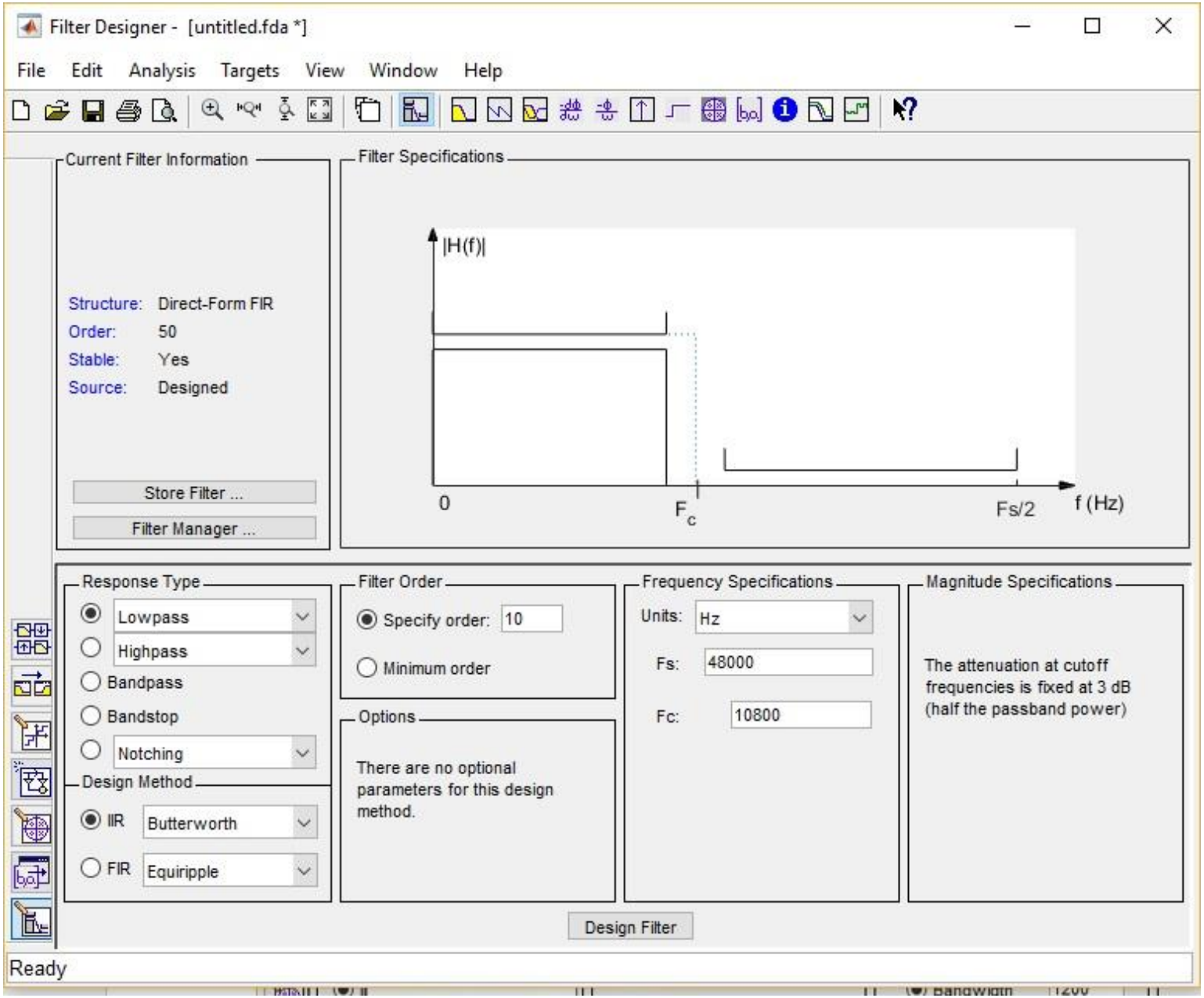
Filter designer

Notch filter for dummies

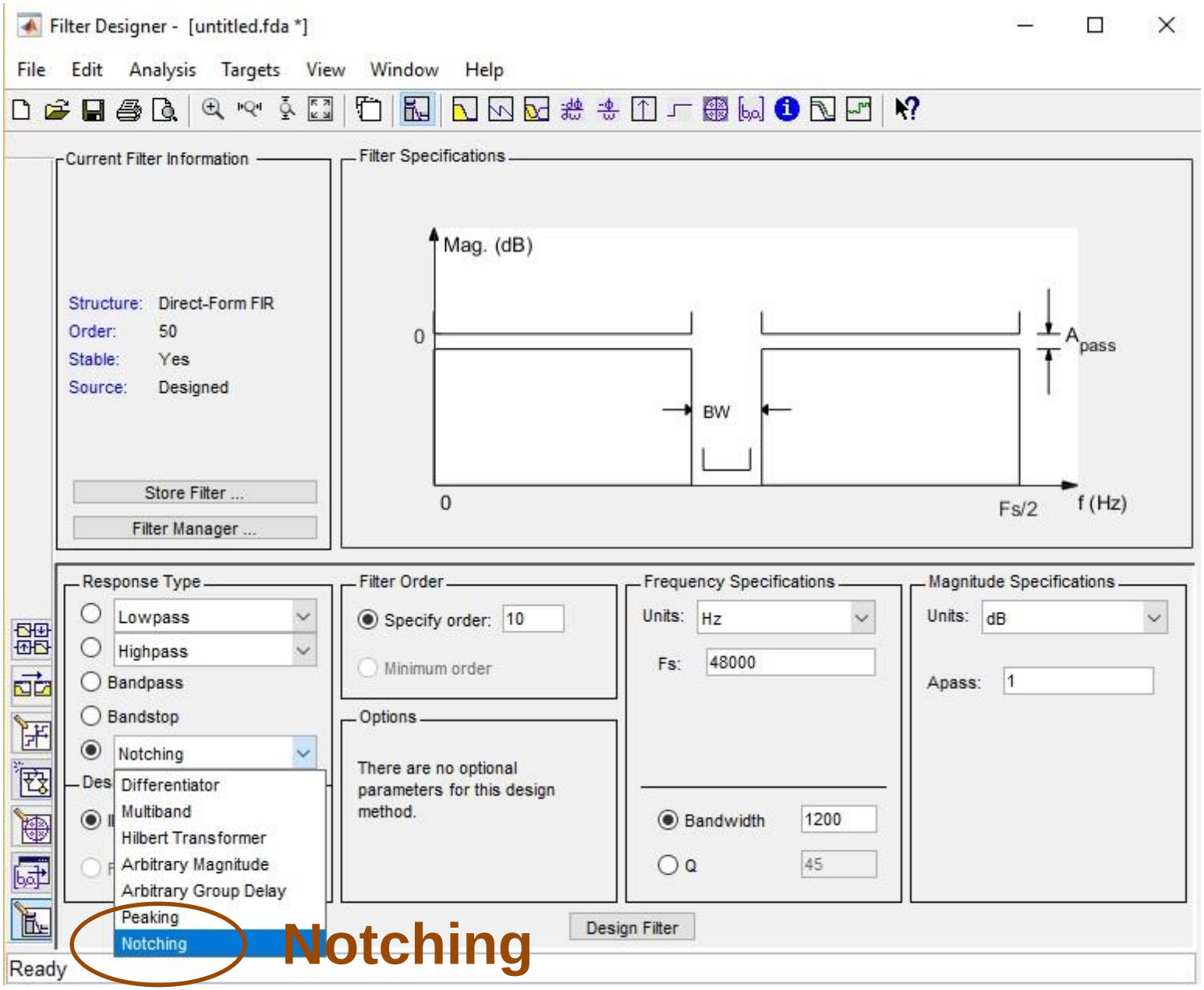


Close

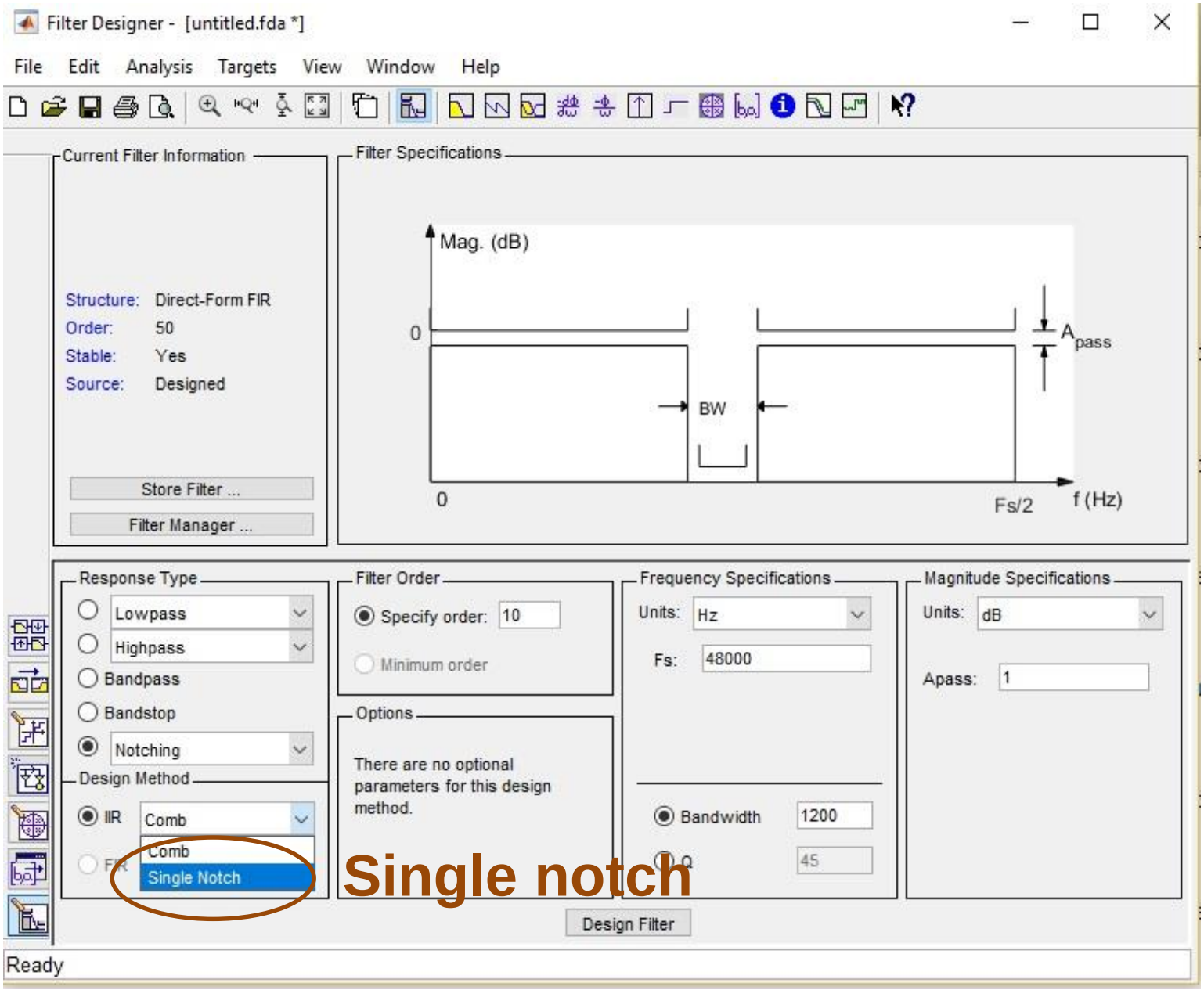
Notch filter for dummies



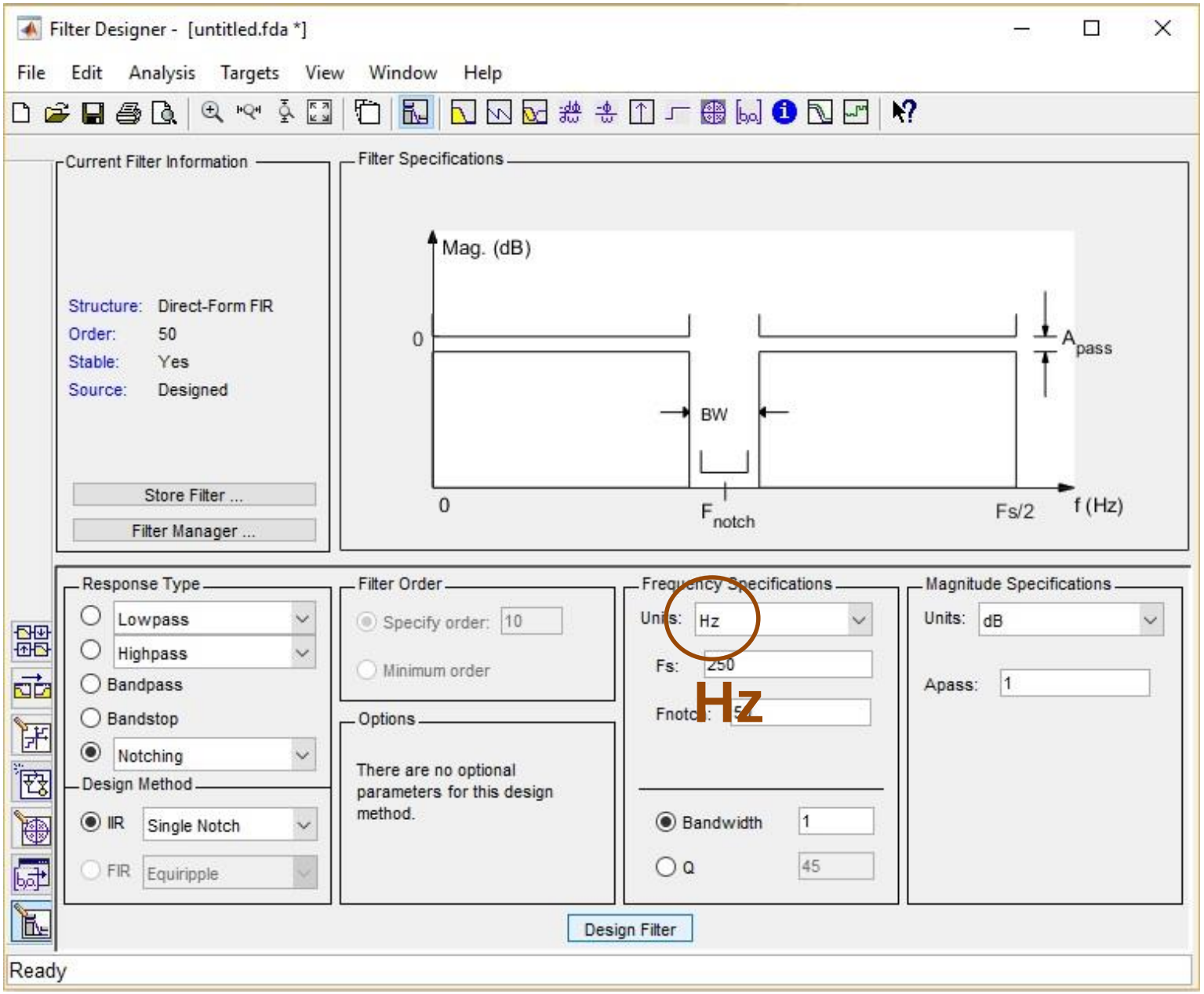
Notch filter for dummies



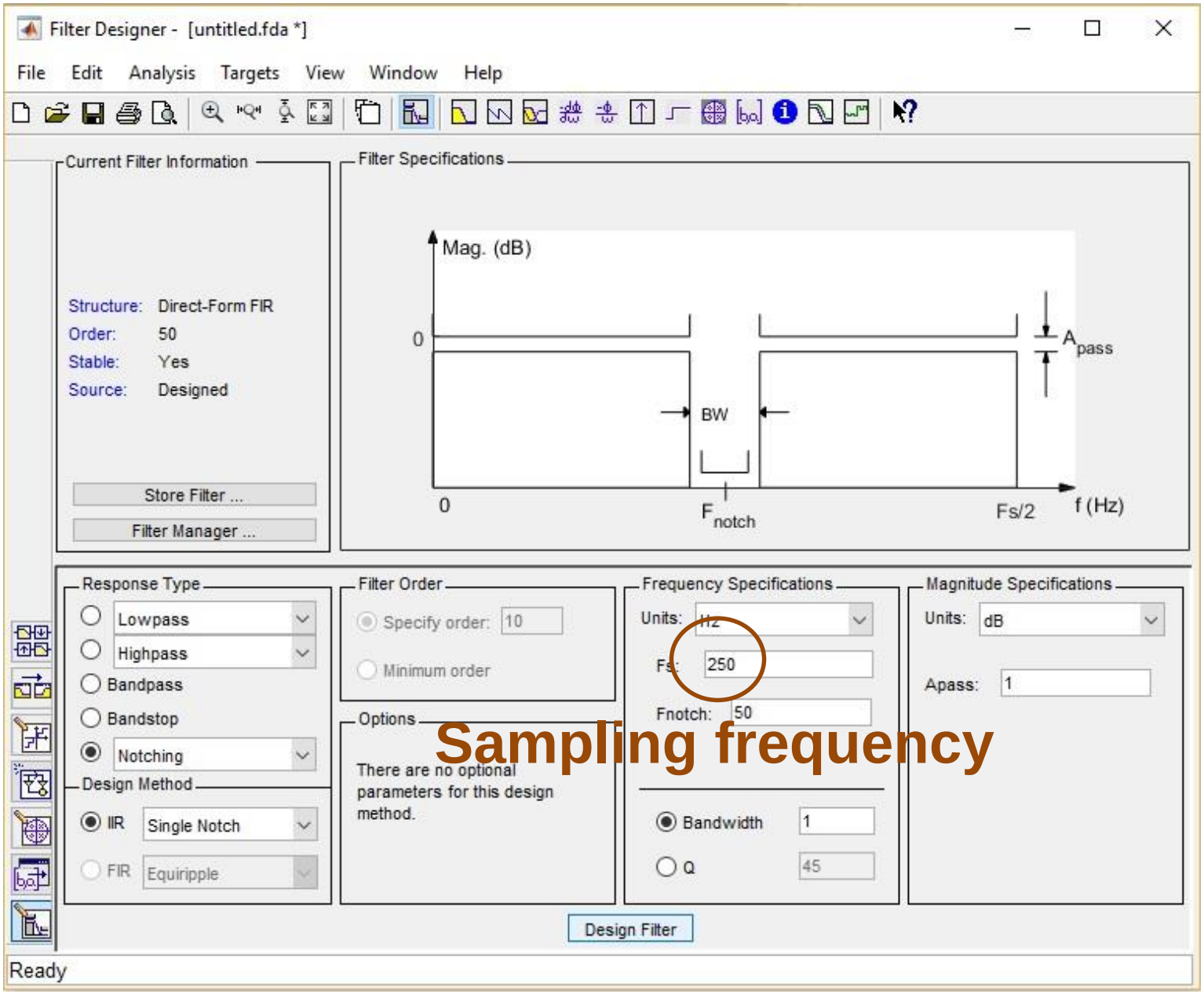
Notch filter for dummies



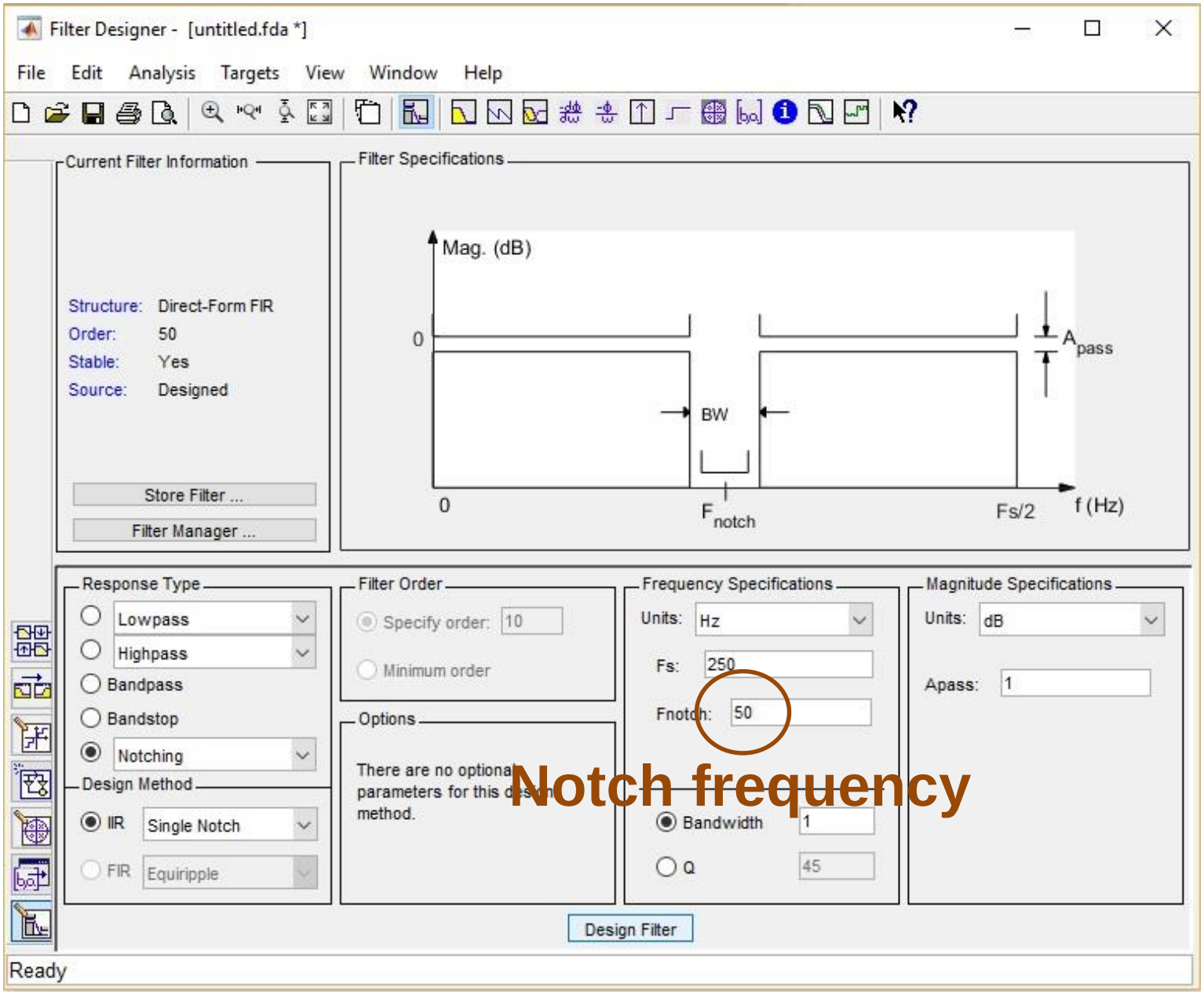
Notch filter for dummies



Notch filter for dummies

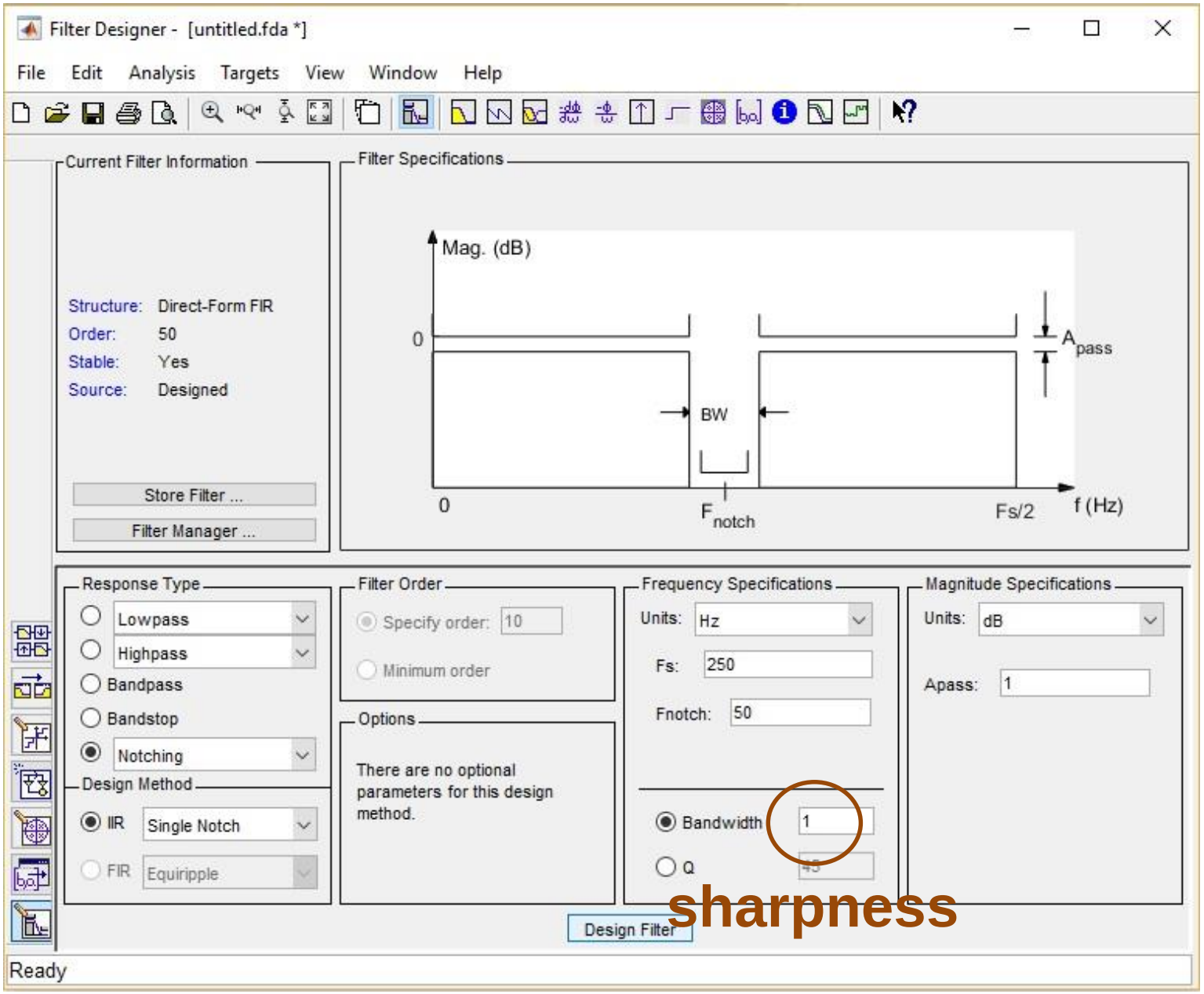


Notch filter for dummies

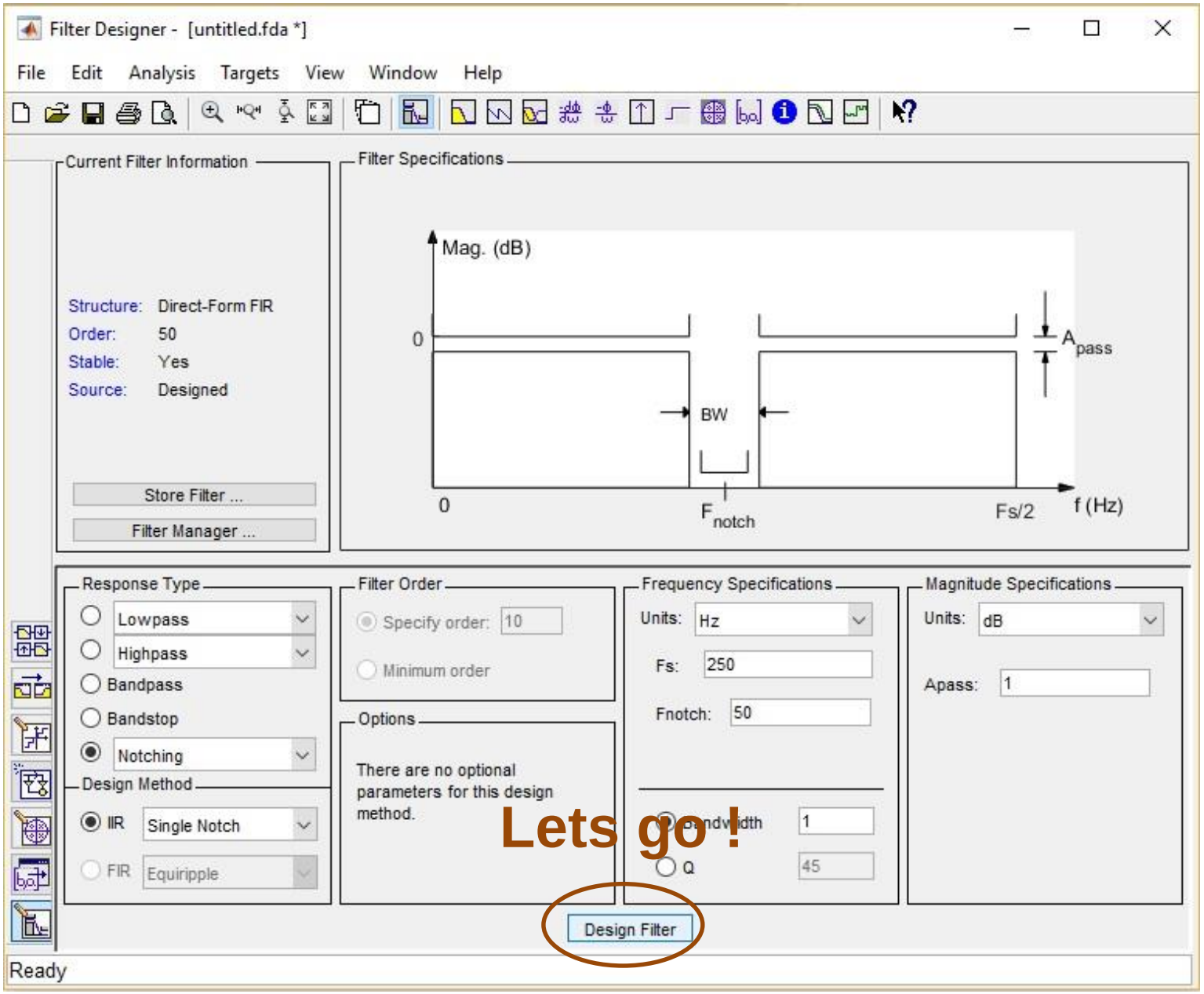


Notch frequency

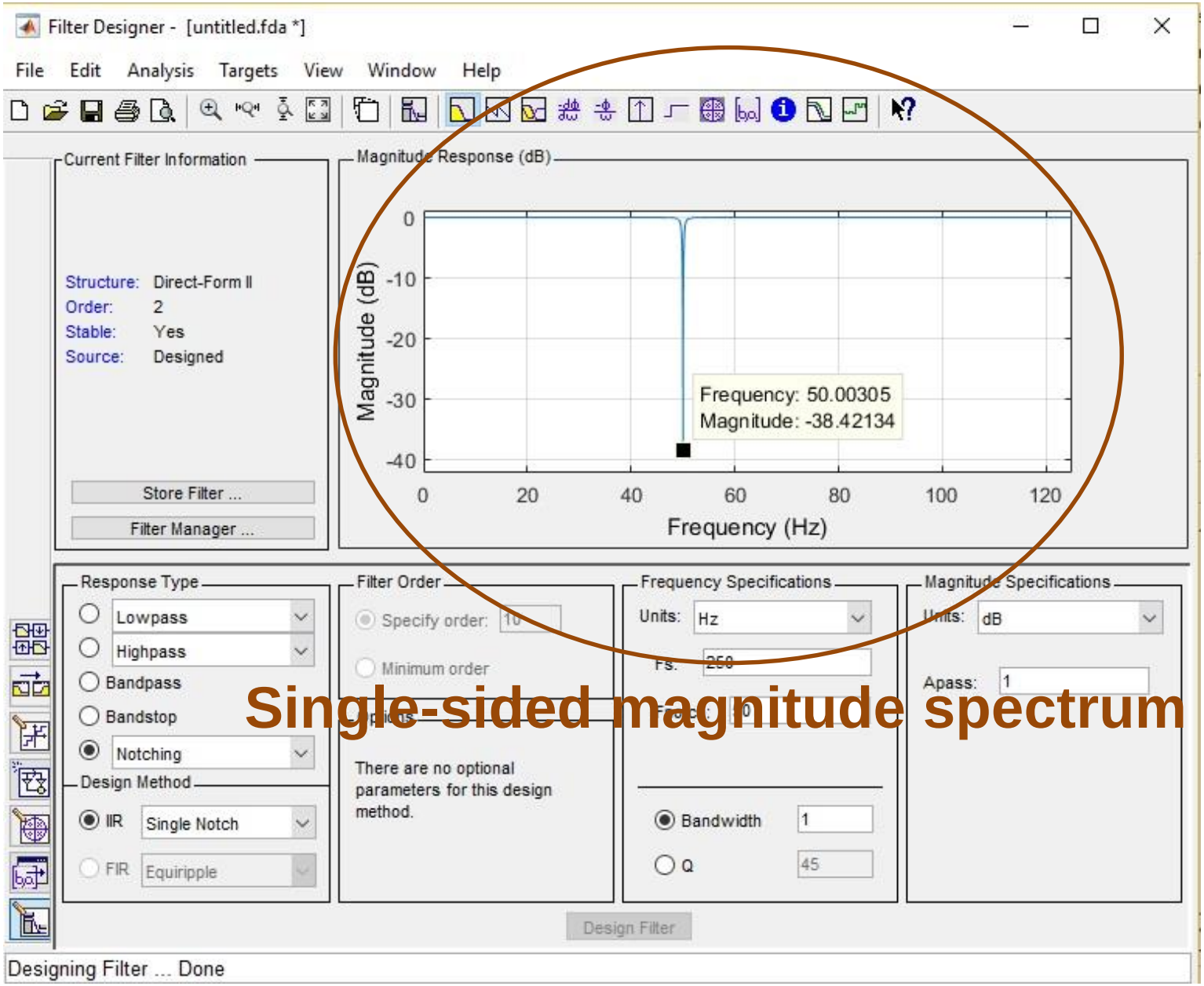
Notch filter for dummies



Notch filter for dummies

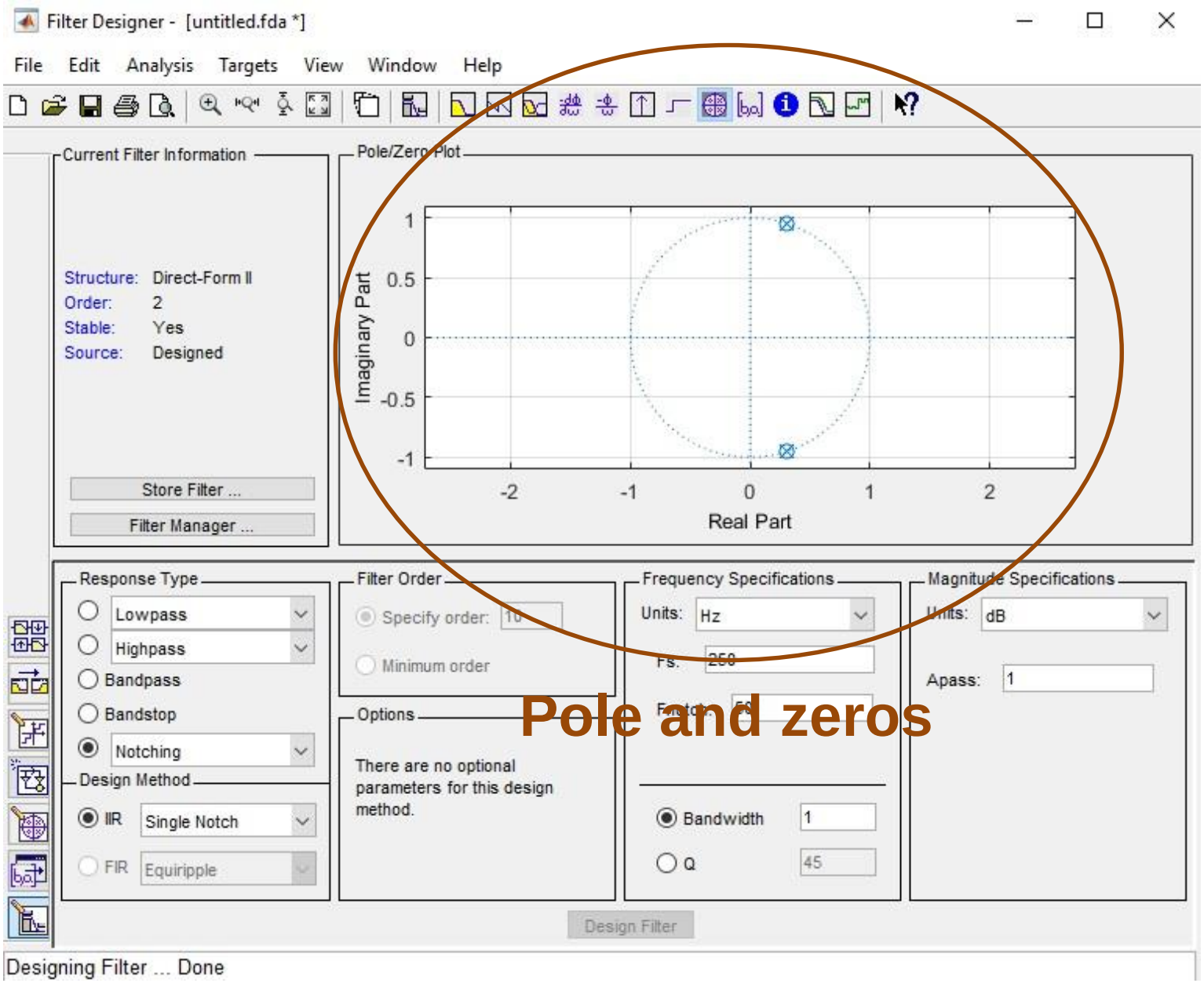


Notch filter for dummies



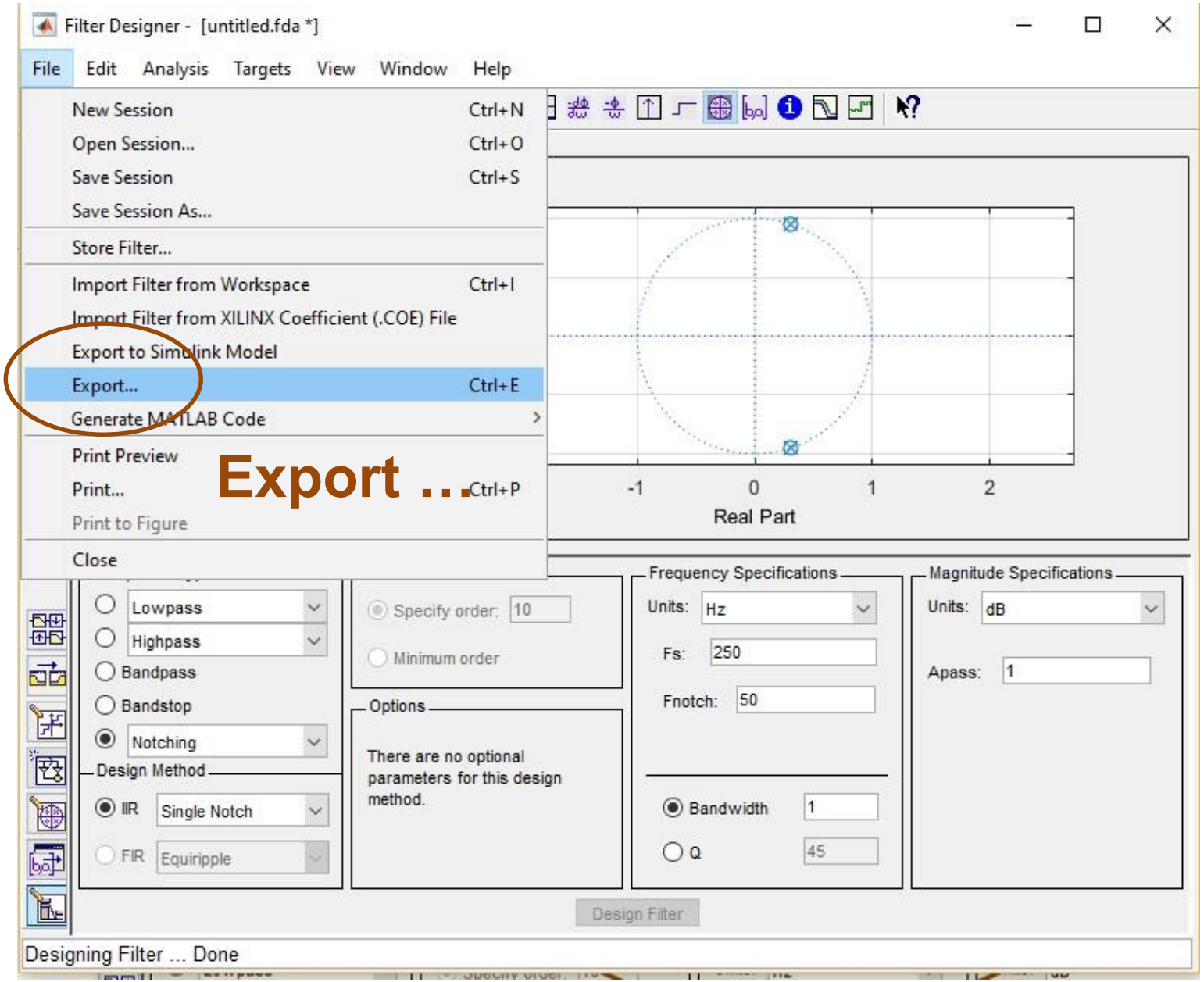
Single-sided magnitude spectrum

Notch filter for dummies

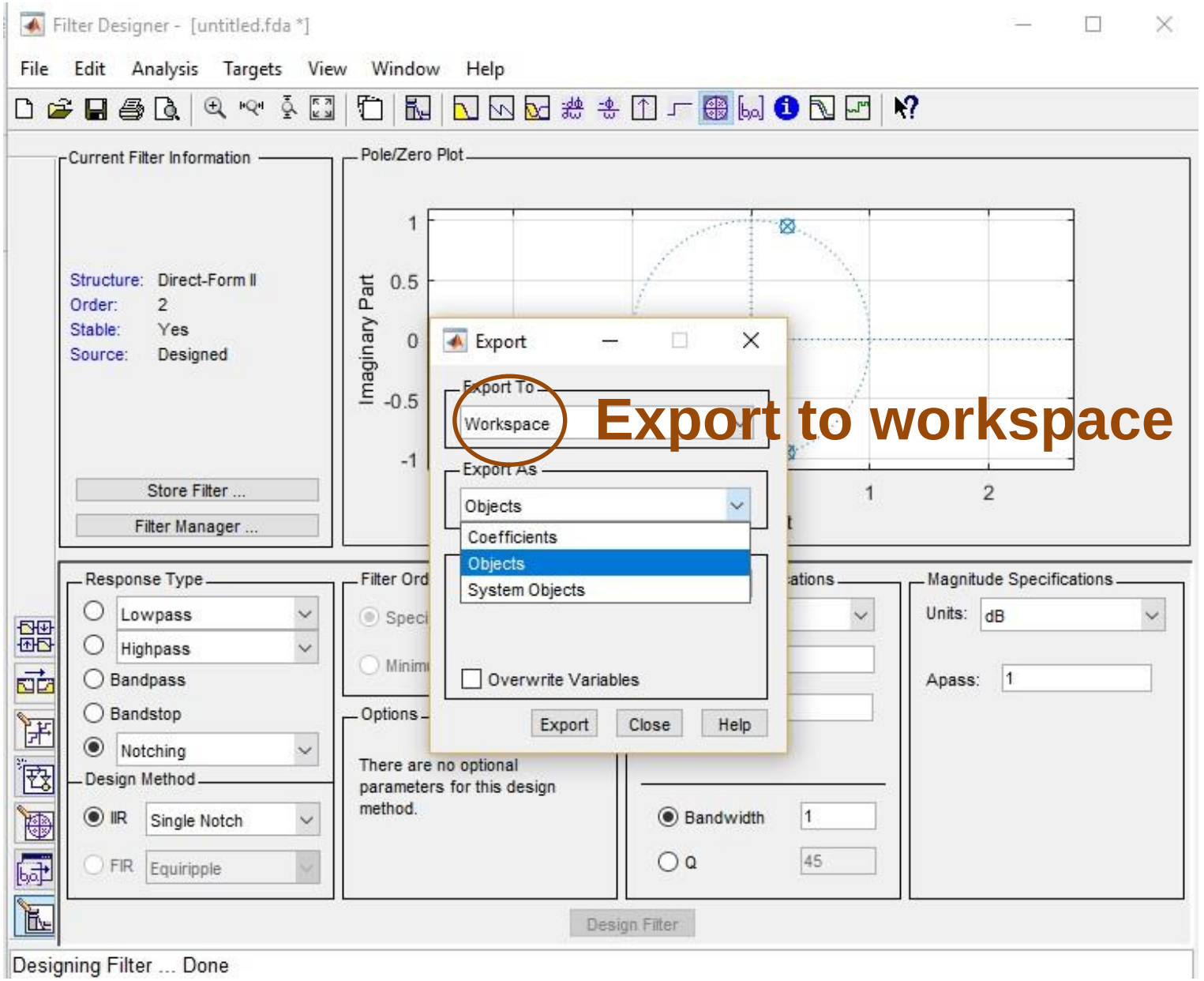


Pole and zeros

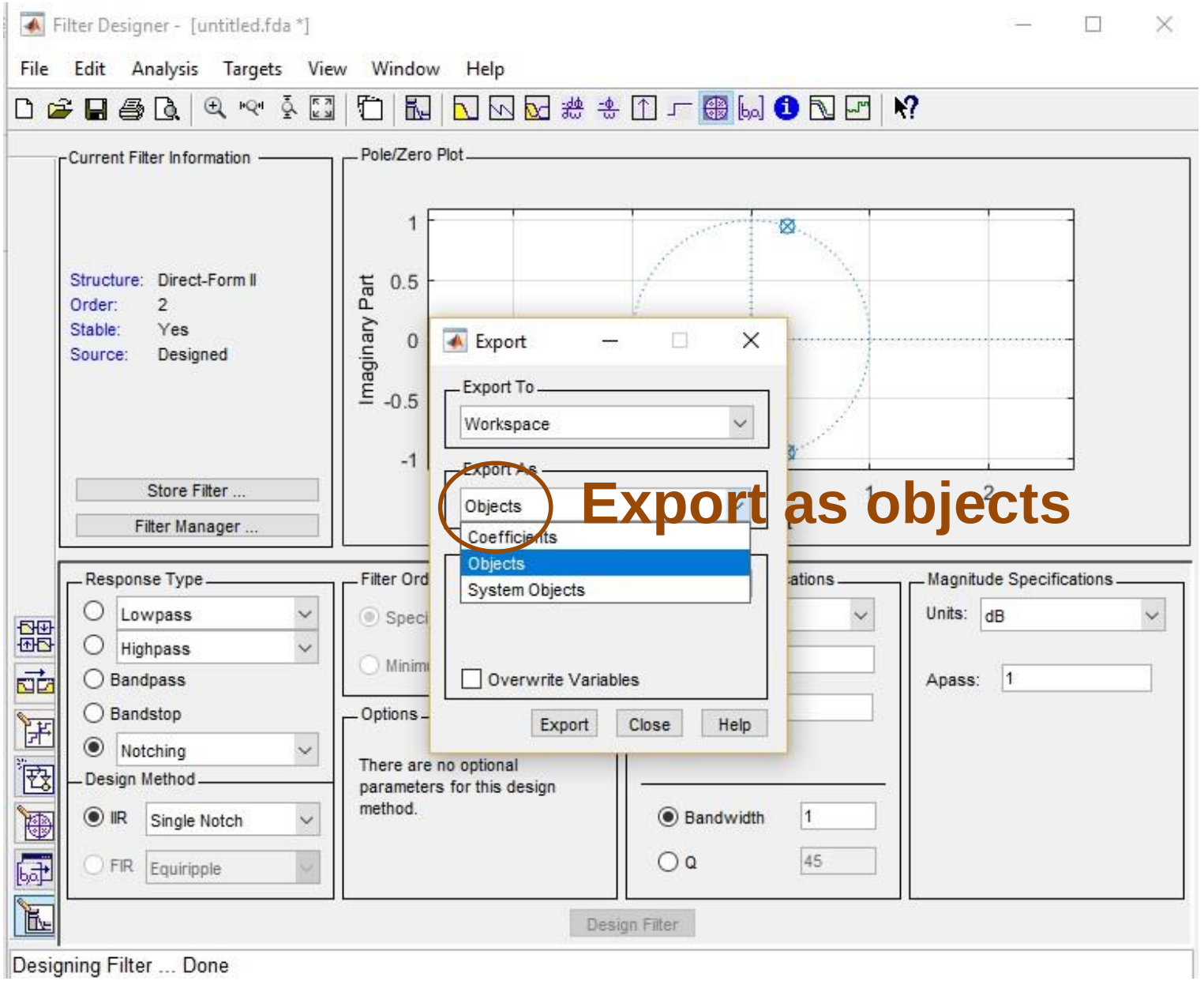
Notch filter for dummies



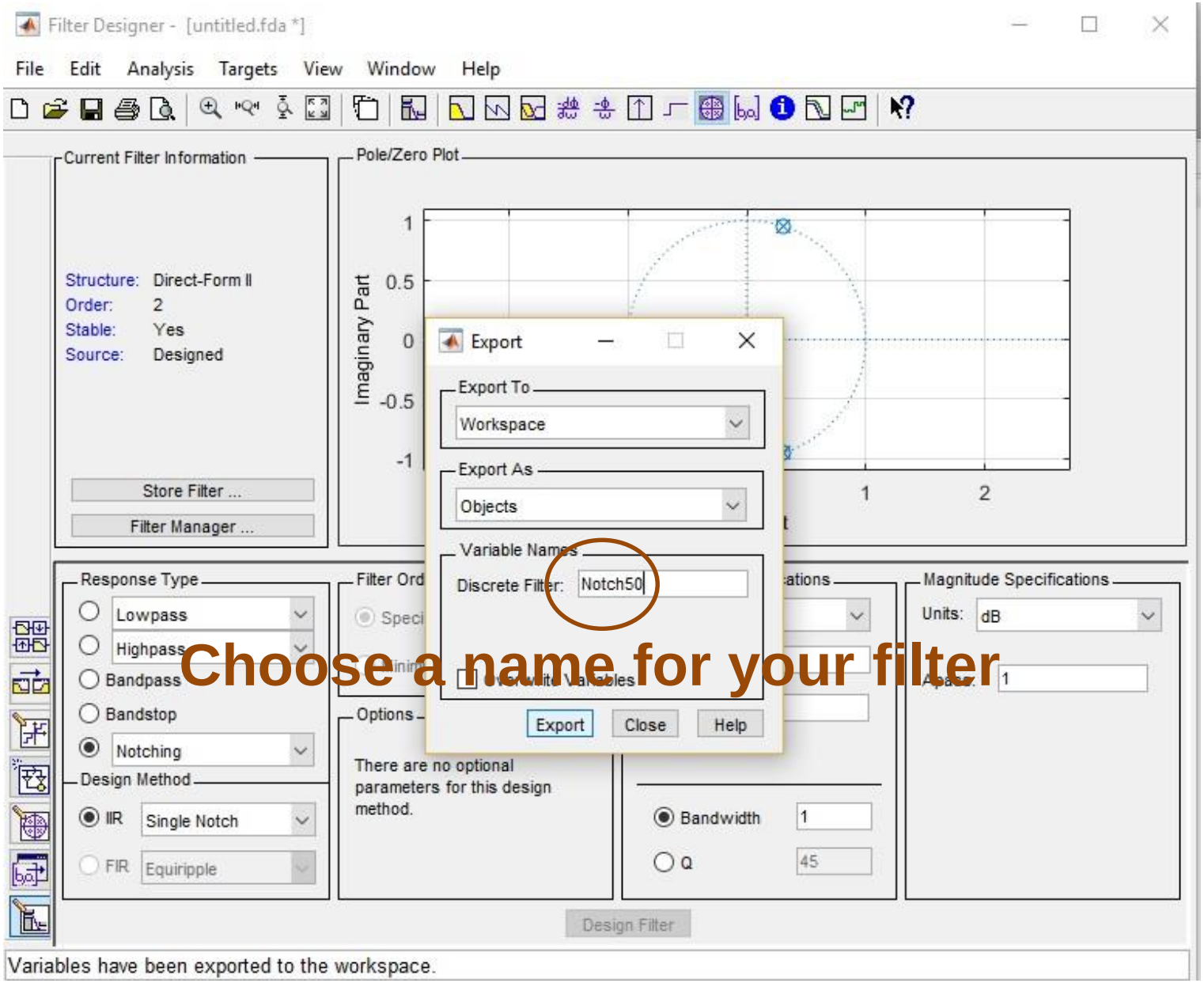
Notch filter for dummies



Notch filter for dummies

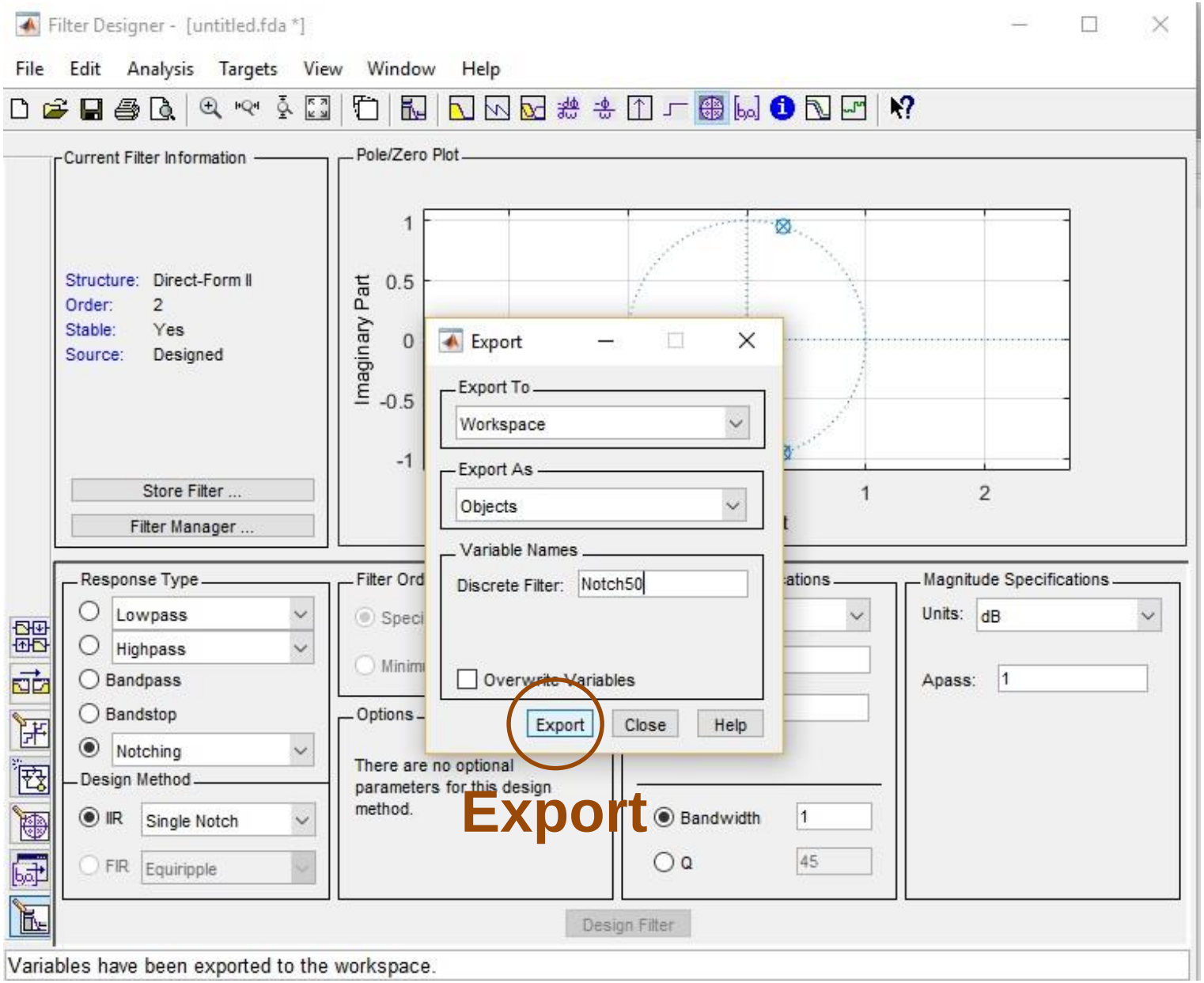


Notch filter for dummies



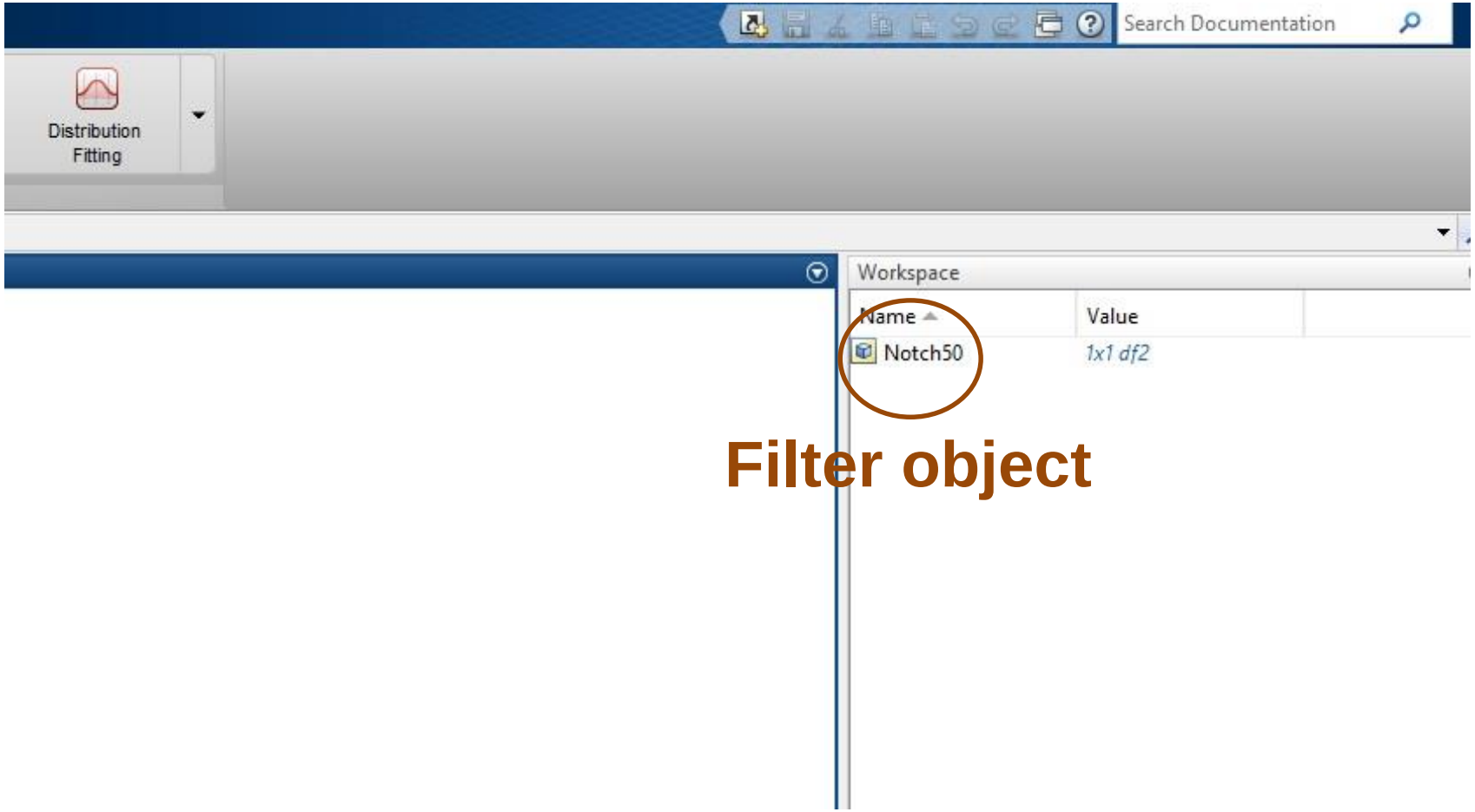
Choose a name for your filter

Notch filter for dummies



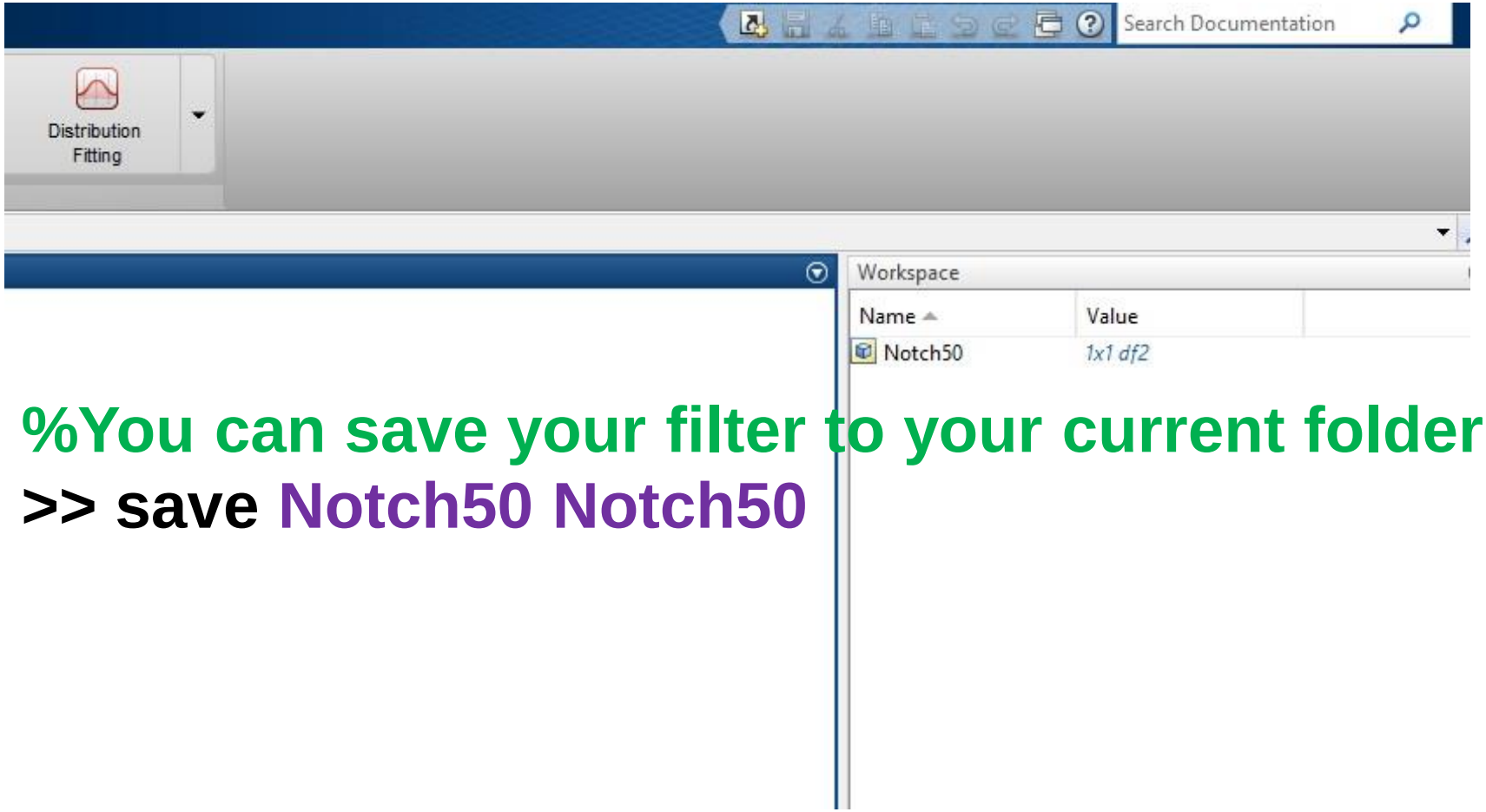
Export

Notch filter for dummies



Filter object

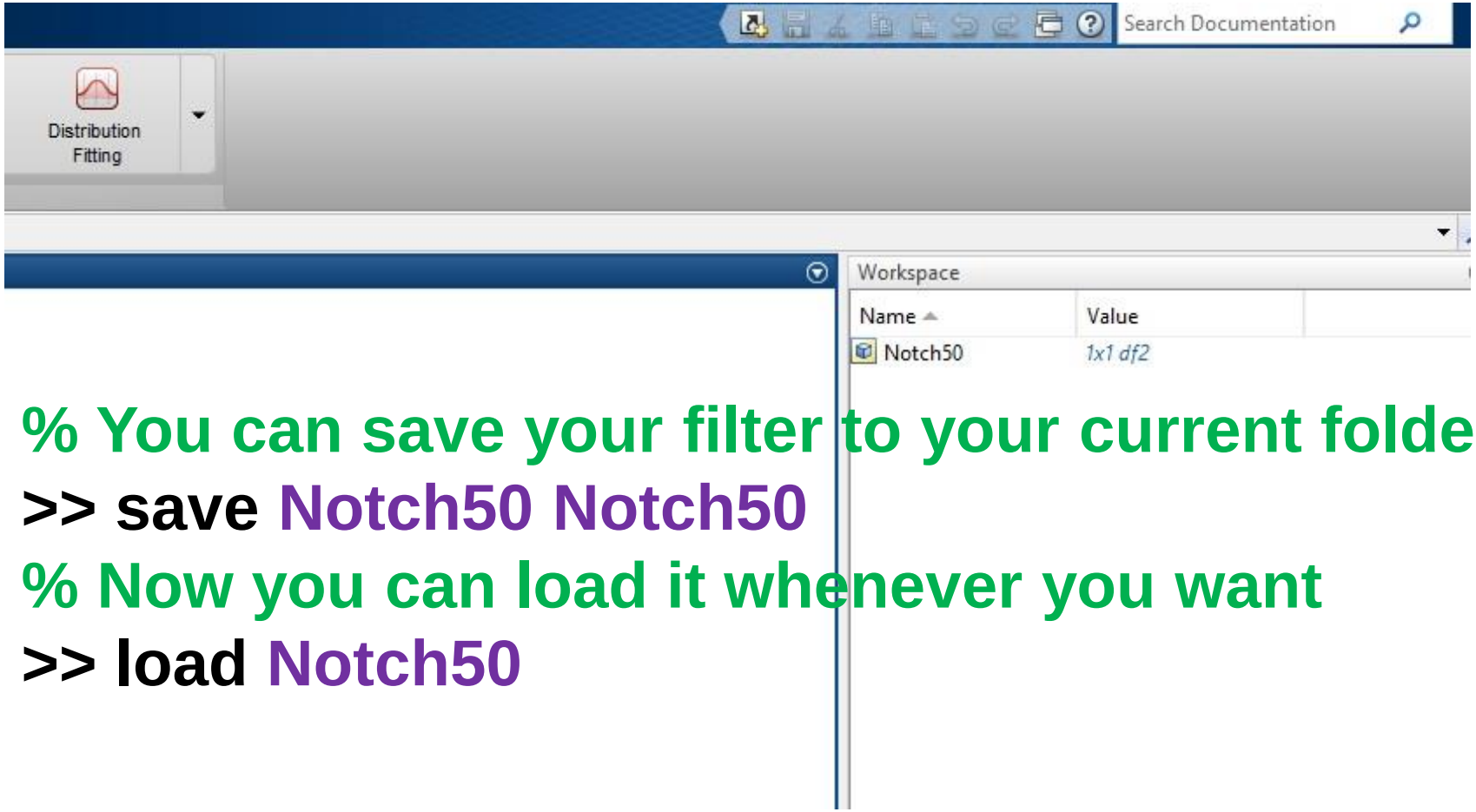
Notch filter for dummies



%You can save your filter to your current folder

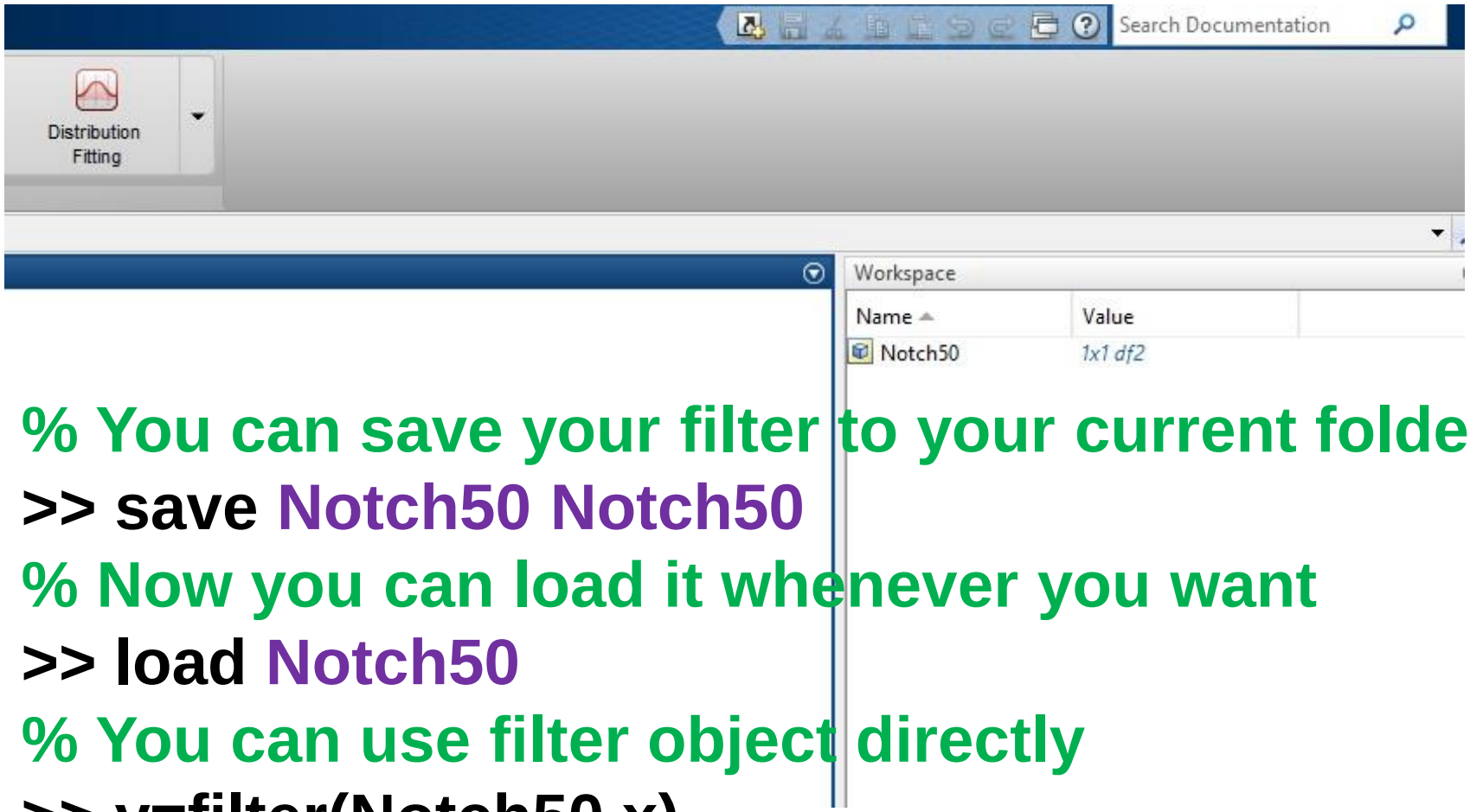
>> save Notch50 Notch50

Notch filter for dummies




```
% You can save your filter to your current folder
>> save Notch50 Notch50
% Now you can load it whenever you want
>> load Notch50
```

Notch filter for dummies




```
% You can save your filter to your current folder
>> save Notch50 Notch50
% Now you can load it whenever you want
>> load Notch50
% You can use filter object directly
>> y=filter(Notch50,x)
```

Step 4: filter out undesired frequency components

```
% load the filter object
load Notch50;
% removing the noise
pure_ecg=filter(Notch50,ecg);
```

Step 5: plot the results

```
figure(3)
plot(t,pure_ecg,'LineWidth',2); title('without noise')
xlabel('time (s)')
ylabel('amplitude')
% Zoom in
figure(4)
plot(t(500:length(t)/6),ecg(500:length(t)/6),…
'LineWidth',2.5); title('Noisy and without noise')
xlabel('time')
ylabel('amplitude')
hold on
plot(t(500:length(t)/6),pure_ecg(500:length(t)/6),…
'LineWidth',2.5)
```

Open filter visualization tool

MATLAB:

fvtool(b,a)
or
fvetool(object)
or
…

Useful links

- https://www.youtube.com/watch?v=utrb6DN-Pqc
- https://www.youtube.com/watch?v=aZ9fnLzPIWo
- https://nl.mathworks.com/help/signal/ug/getting-started-with-filter-designer.html