```
clearvars
format long
```

Research Techniques Project Just Milestone 5

Input all data

Text files:

```
opts = detectImportOptions("BD+55_441.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
```

"time" is in the units of days and "flux" is "rel_flux_T1" from AstroImageJ outputs.

```
opts.VariableTypes = {'double','double','double','double','double','double'};
preview("BD+55_441.txt",opts)
```

```
ans = 8×6 table
```

	B_time	B_flux	R_time	R_flux	V_time	V_flux
1	5.873268430000000e+04	3.497996000000000	5.873268459000000e+04	1.961830000000000	5.873268445000000e+04	2.352058000000000
2	5.873268430000000e+04	3.497996000000000	5.873268504000000e+04	1.942857000000000	5.873268490000000e+04	2.360458000000000
3	5.873268475000000e+04	3.487289000000000	5.873268549000000e+04	1.947021000000000	5.873268535000000e+04	2.329957000000000
4	5.873268520000000e+04	3.450942000000000	5.873268594000000e+04	1.930594000000000	5.873268580000000e+04	2.344889000000000
5	5.873268565000000e+04	3.497567000000000	5.873268639000000e+04	1.942537000000000	5.873268625000000e+04	2.361048000000000
6	5.873268610000000e+04	3.461555000000000	5.873268684000000e+04	1.935141000000000	5.873268669000000e+04	2.333341000000000
7	5.873268655000000e+04	3.431248000000000	5.873268729000000e+04	1.931290000000000	5.873268714000000e+04	2.331904000000000
8	5.873268700000000e+04	3.455091000000000	5.873268774000000e+04	1.928260000000000	5.873268759000000e+04	2.323782000000000

```
BD55_441 = readmatrix("BD+55_441.txt",opts);
whos BD55_441
```

```
  Name             Size             Bytes  Class     Attributes

  BD55_441       101x6               4848  double              
```

```
%BD+48_1098
opts = detectImportOptions("BD+48_1098.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
BD48_1098 = rmmissing(readmatrix("BD+55_441.txt",opts)); %the matrix data of the text file
%preview("BD+48_1098.txt",opts)
whos BD48_1098
```

```
  Name              Size             Bytes  Class     Attributes

  BD48_1098       100x6               4800  double              
```

```
%HD28497
opts = detectImportOptions("HD28497.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD28497 = rmmissing(readmatrix("HD28497.txt",opts)) %the matrix data of the text file
```

```
HD28497 = 600×6
10^4 ×

    5.8442    0.1519    5.8442    0.0475    5.8442    0.0698
    5.8442    0.1577    5.8442    0.0479    5.8442    0.0656
    5.8442    0.1577    5.8442    0.0479    5.8442    0.0656
    5.8442    0.1476    5.8442    0.0507    5.8442    0.0744
    5.8442    0.1541    5.8442    0.0487    5.8442    0.0718
    5.8442    0.1544    5.8442    0.0465    5.8442    0.0708
    5.8442    0.1540    5.8442    0.0529    5.8442    0.0773
    5.8442    0.1585    5.8442    0.0501    5.8442    0.0713
    5.8442    0.1597    5.8442    0.0538    5.8442    0.0721
    5.8442    0.1406    5.8442    0.0509    5.8442    0.0710
    5.8442    0.1598    5.8442    0.0533    5.8442    0.0746
```

```
%preview("HD28497.txt",opts)
whos HD28497
```

```
  Name           Size             Bytes  Class     Attributes

  HD28497      600x6              28800  double
```

```
%HD46131
opts = detectImportOptions("HD46131.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_flux','B_time','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD46131 = rmmissing(readmatrix("HD46131.txt",opts)); %the matrix data of the text file
preview("HD46131.txt",opts)
```

ans = 8×6 table

	B_flux	B_time	R_time	R_flux	V_time	V_flux
1	80.304181000000000	5.844670852000000e+04	29.694341000000001	5.844670891000000e+04	43.242789999999999	5.844670875000000e+04
2	87.015710999999996	5.844670916000000e+04	29.507301999999999	5.844670956000000e+04	43.895994999999999	5.844670939000000e+04
3	86.150750000000002	5.844670981000000e+04	29.594172000000000	5.844671021000000e+04	43.113675000000001	5.844671004000000e+04
4	84.585120000000003	5.844671046000000e+04	29.019480000000001	5.844671085000000e+04	43.536724999999997	5.844671069000000e+04
5	82.723350999999994	5.844671111000000e+04	29.257328999999999	5.844671150000000e+04	44.345345000000002	5.844671134000000e+04
6	79.700693000000001	5.844671175000000e+04	30.099762999999999	5.844671215000000e+04	43.526121000000003	5.844671199000000e+04
7	84.427550999999994	5.844671240000000e+04	29.990238999999999	5.844671280000000e+04	42.635756000000001	5.844671263000000e+04
8	79.153425999999996	5.844671305000000e+04	29.780404000000001	5.844671344000000e+04	43.533228000000001	5.844671329000000e+04

```
whos HD46131
```

```
  Name           Size             Bytes  Class     Attributes

  HD46131      250x6              12000  double
```

```
%HD88661
opts = detectImportOptions("HD88661.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD88661 = rmmissing(readmatrix("HD88661.txt",opts)); %the matrix data of the text file
%preview("HD88661.txt",opts)
whos HD88661
```

```
  Name          Size             Bytes  Class     Attributes

  HD88661      35x6               1680  double
```

```
%HD105521
opts = detectImportOptions("HD105521.txt");
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD105521 = rmmissing(readmatrix("HD105521.txt",opts)); %the matrix data of the text file
%preview("HD105521.txt",opts)
whos HD105521
```

```
  Name            Size             Bytes  Class     Attributes

  HD105521      180x6               8640  double
```

```
%HD105521
opts = detectImportOptions("HD105521.txt");
```

```
opts.DataLines = 3;
opts.VariableNames = {'B_time','B_flux','R_time','R_flux','V_time','V_flux'};
opts.VariableTypes = {'double','double','double','double','double','double'};
HD105521 = rmmissing(readmatrix("HD105521.txt",opts)); %the matrix data of the text file
%preview("HD105521.txt",opts)
whos HD105521
```

```
  Name            Size            Bytes  Class     Attributes

  HD105521      180x6              8640  double
```

CSV files

```
%HD106306
%B_filter
opts = detectImportOptions('HD106306_B.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_B.csv",opts)
HD106306_B = readmatrix("HD106306_B.csv",opts);

%R_filter
opts = detectImportOptions('HD106306_R.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_R.csv",opts)
HD106306_R = readmatrix("HD106306_R.csv",opts);

%V_filter
opts = detectImportOptions('HD106306_V.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD106306_V.csv",opts)
HD106306_V = readmatrix("HD106306_V.csv",opts);

HD106306 = rmmissing([HD106306_B HD106306_R HD106306_V]); %the matrix data of the text file
whos HD106306
```

```
  Name            Size            Bytes  Class     Attributes

  HD106306      100x6              4800  double
```

```

%HD147302
%B_filter
opts = detectImportOptions('HD147302_B.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_B.csv",opts)
HD147302_B = readmatrix("HD147302_B.csv",opts);

%R_filter
opts = detectImportOptions('HD147302_R.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_R.csv",opts)
HD147302_R = readmatrix("HD147302_R.csv",opts);

%V_filter
opts = detectImportOptions('HD147302_V.csv');
opts.DataLines = [2 Inf];
All_fields_available = opts.VariableNames;
opts.SelectedVariableNames = {'J_D__2400000','rel_flux_T1'};
%preview("HD147302_V.csv",opts)
HD147302_V = readmatrix("HD147302_V.csv",opts);

HD147302 = rmmissing([HD147302_B HD147302_R HD147302_V]); %the matrix data of the text file
whos HD147302
```

```
  Name            Size             Bytes  Class     Attributes

  HD147302      100x6               4800  double
```

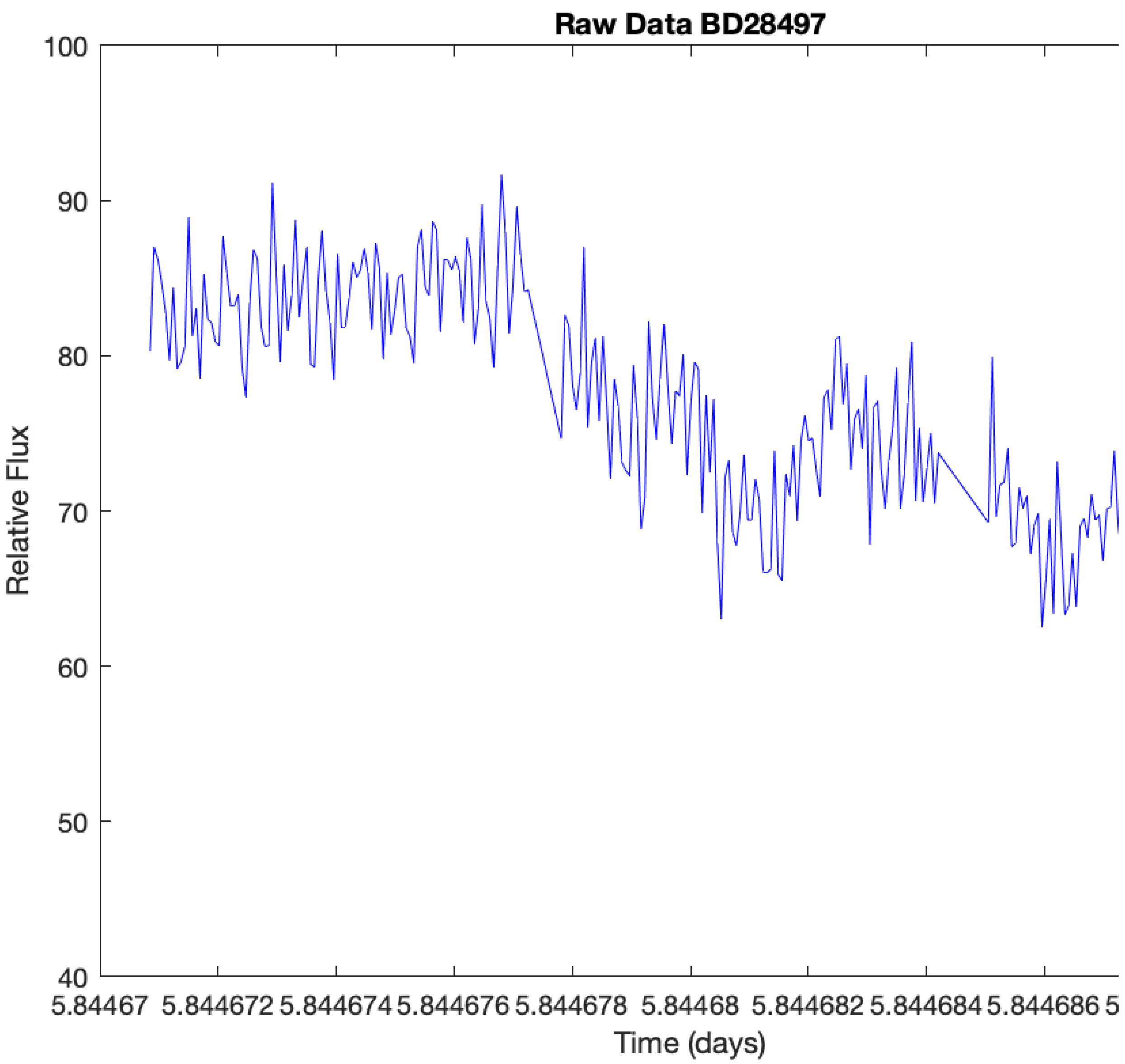
HD152060 and HD209522 can't be described in the same form as the others. Will investigate

```
close all   %closes all previous figures from milestone 1 so that the plots in milestone 2 and 3 work correctly
```

HD46131

Plot the raw data

```
plot(HD46131(:,2),HD46131(:,1),'b');
title('Raw Data BD28497');
xlabel('Time (days)');
ylabel('Relative Flux');
```

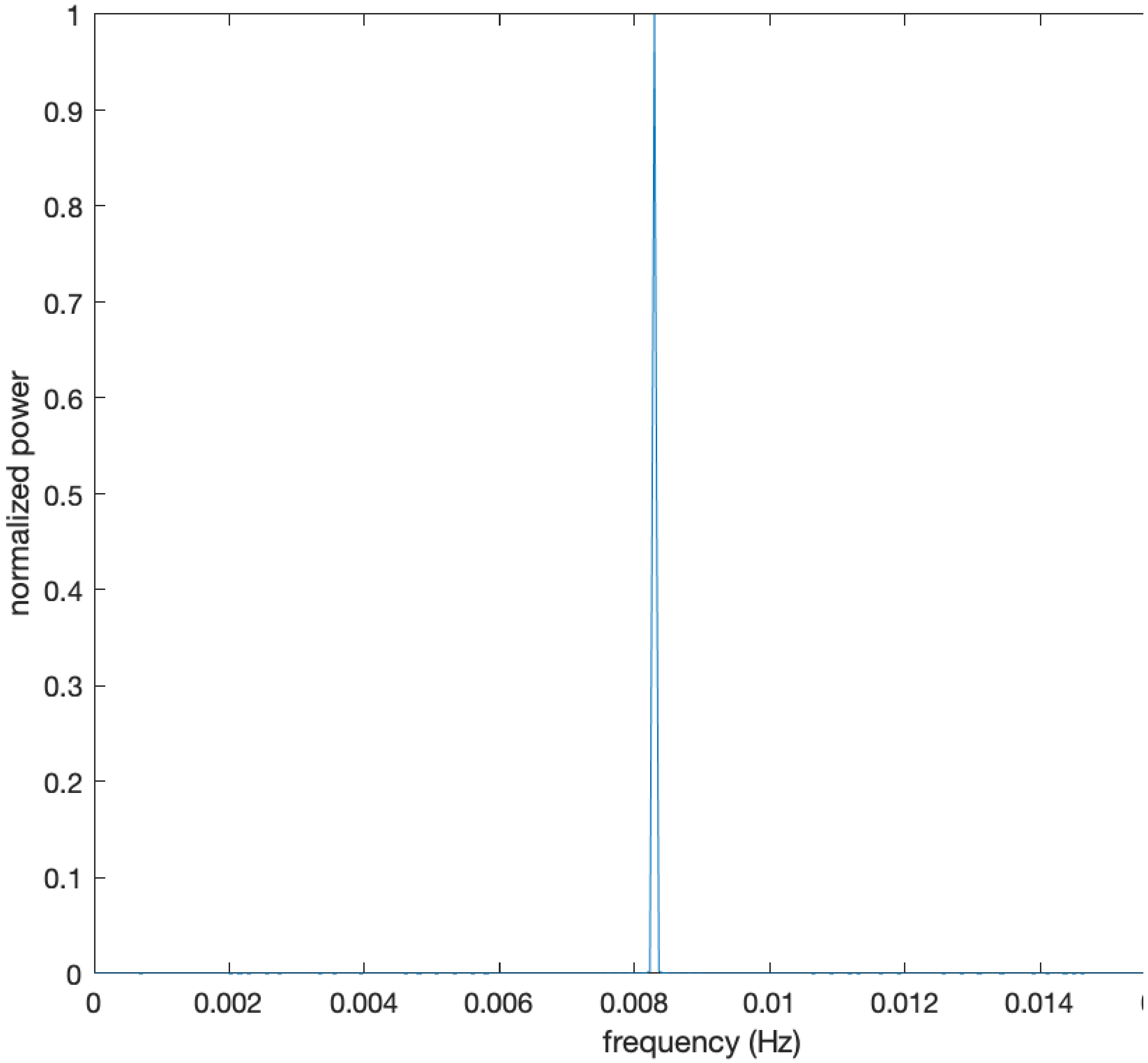


```
avg_freq =1/( mean( diff(HD46131(:,2)) ) *86400 )          %Dr. Riousset showed us how to use mean(diff)
```

```
avg_freq =
   0.016592460385689
```

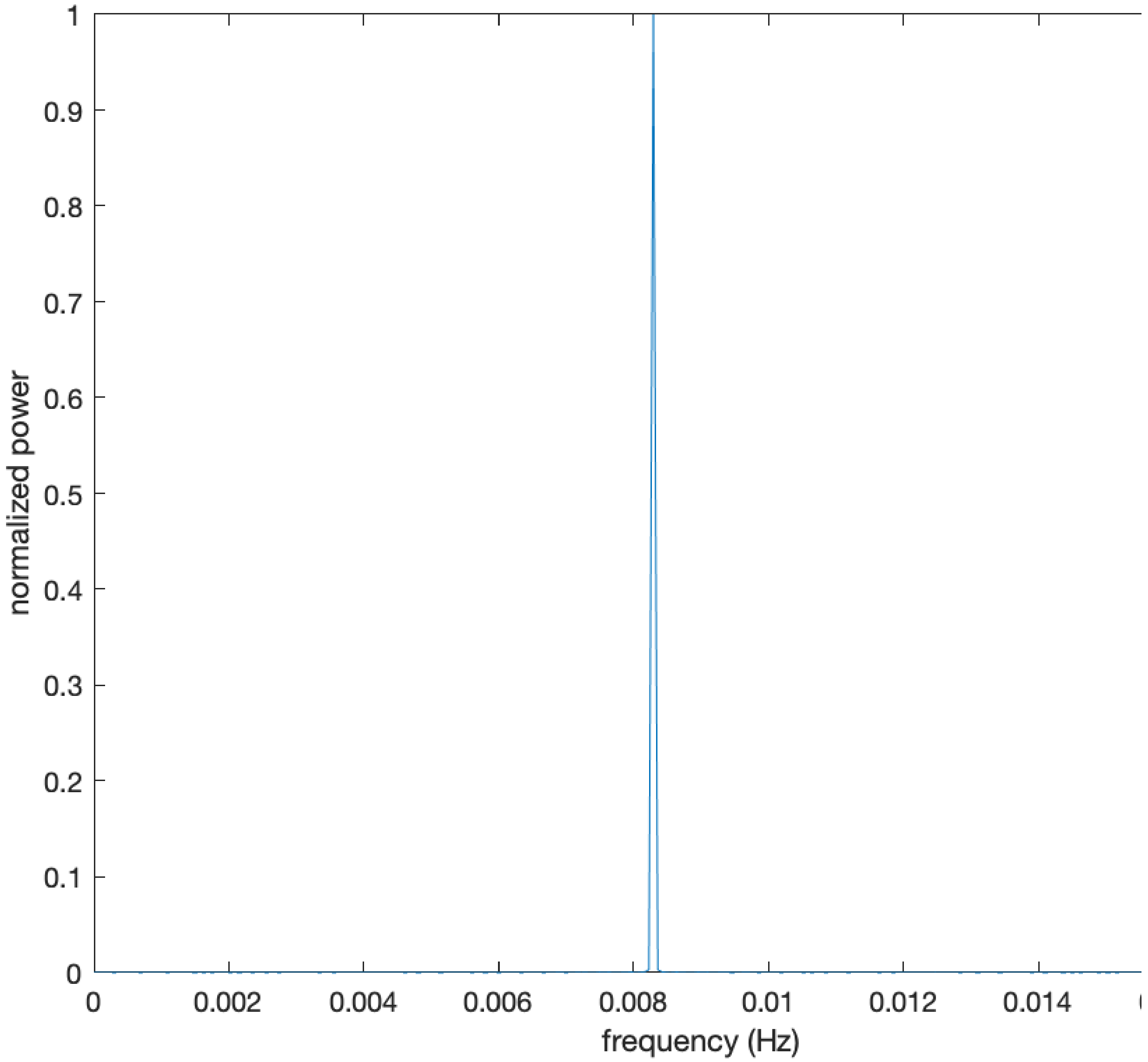
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD46131(:,2),HD46131(:,1));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

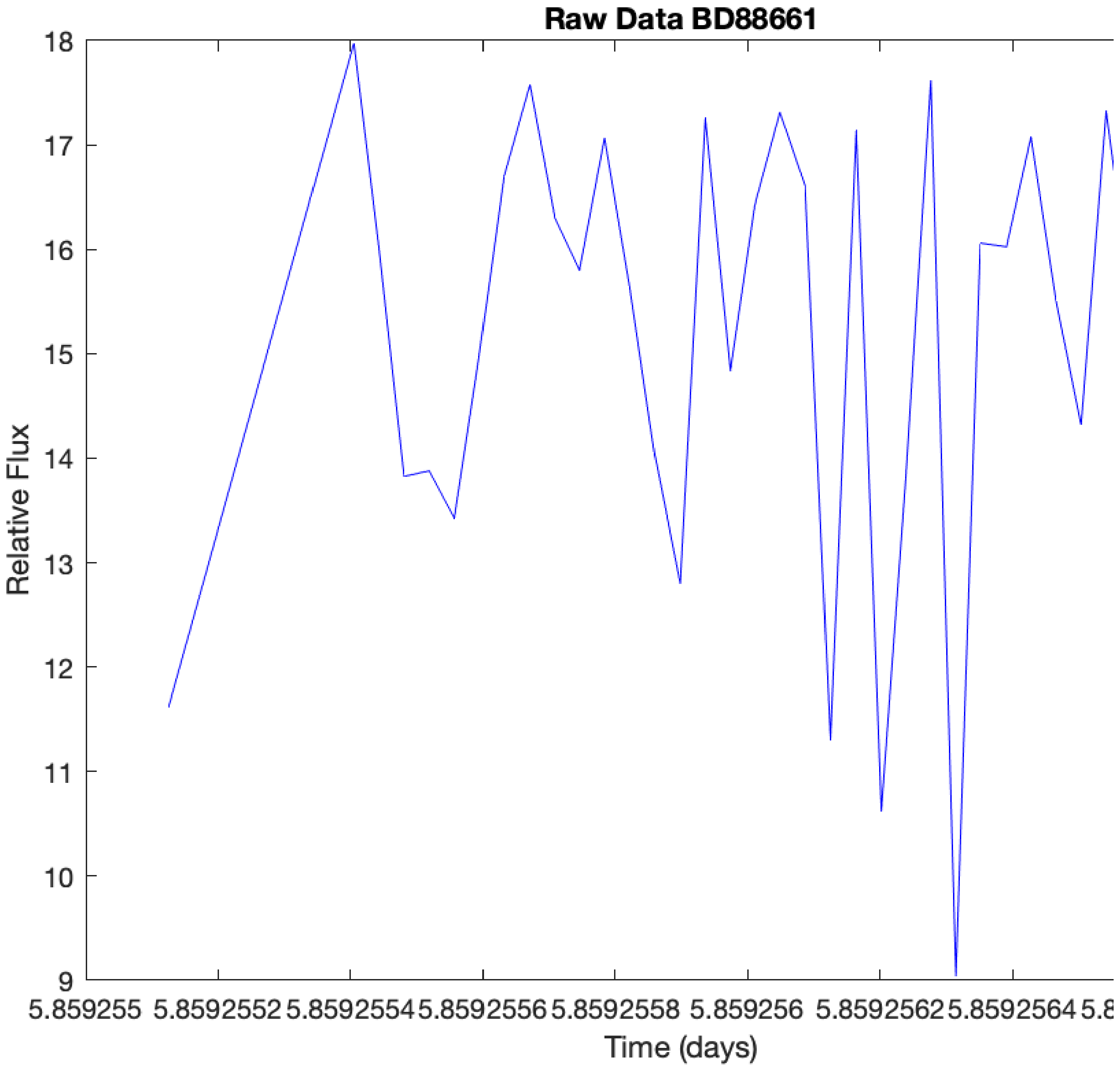
```
[Tnew,Mnew] = Interp_spline(HD46131(:,2),HD46131(:,1));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



HD88661

Plot the raw data

```
plot(HD88661(:,2),HD88661(:,1),'b');
title('Raw Data BD88661');
xlabel('Time (days)');
ylabel('Relative Flux');
```

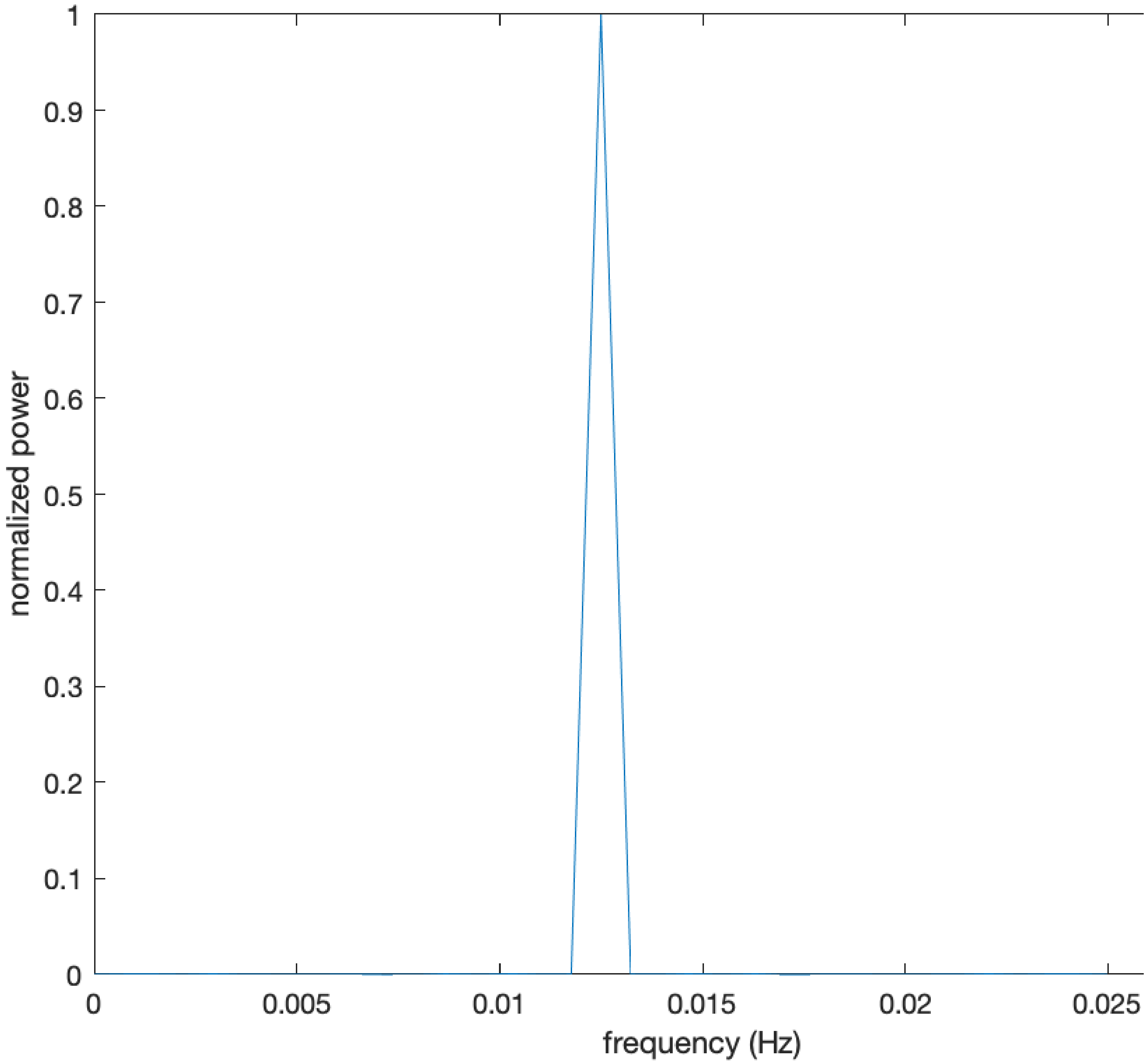


```
avg_freq =1/( mean( diff(HD88661(:,2)) ) *86400 )
```

```
avg_freq =
   0.025736986163581
```

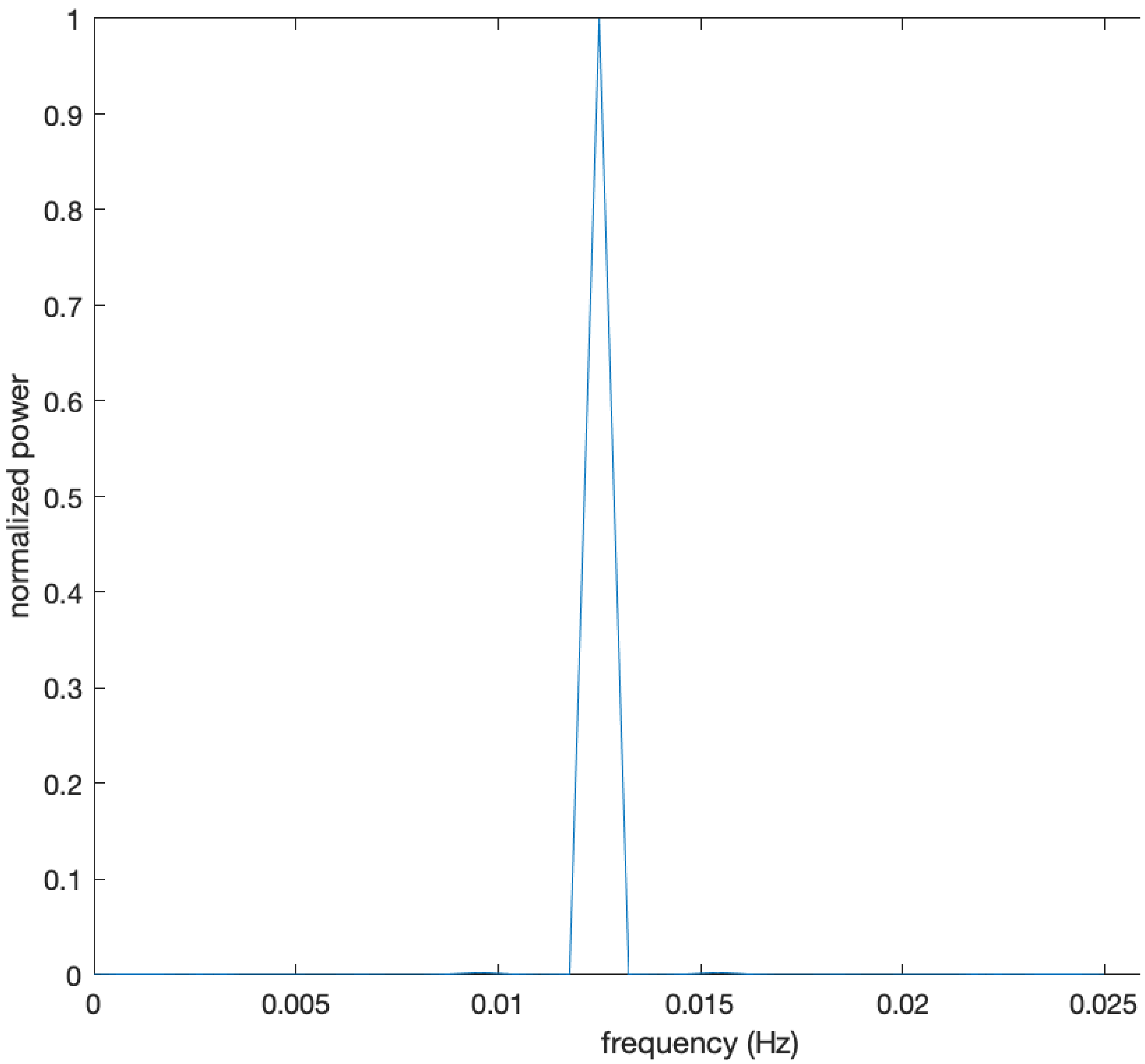
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD88661(:,2),HD88661(:,1));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

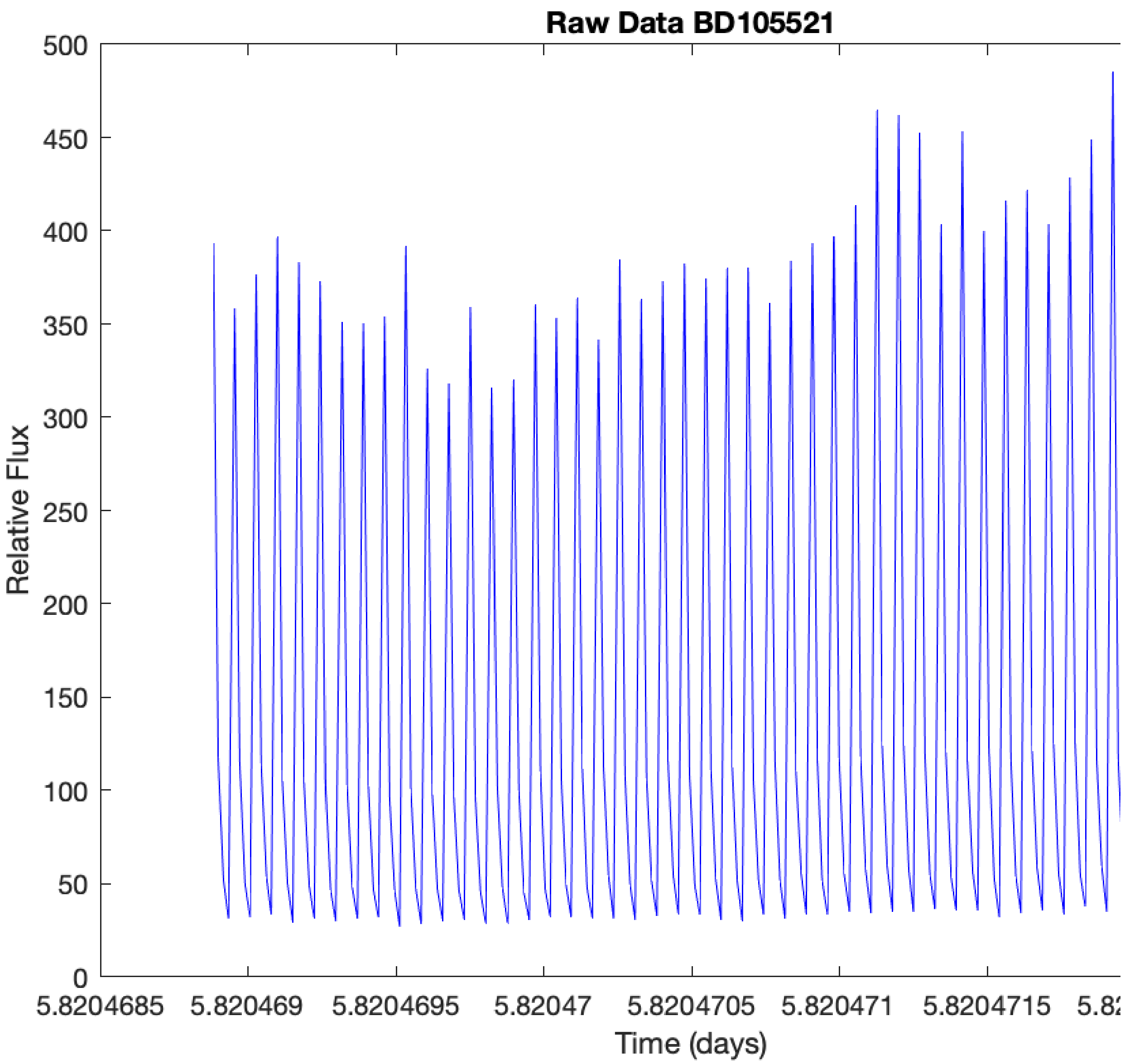
```
[Tnew,Mnew] = Interp_spline(HD88661(:,2),HD88661(:,1));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



HD105521

Plot the raw data

```
plot(HD105521(:,1),HD105521(:,2),'b');
title('Raw Data BD105521');
xlabel('Time (days)');
ylabel('Relative Flux');
```

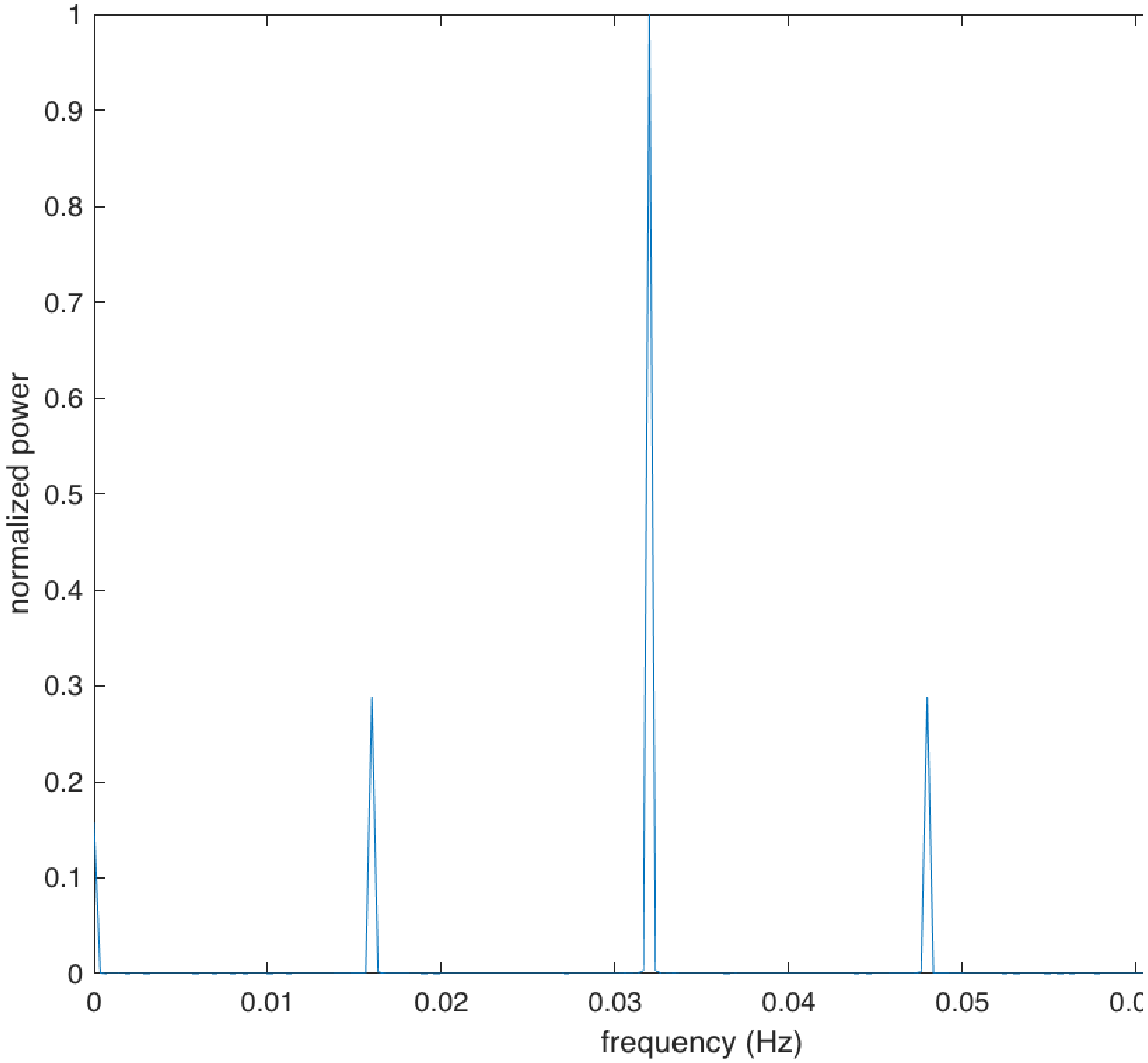


```
avg_freq =1/( mean( diff(HD105521(:,1)) ) *86400 )
```

```
avg_freq =
   0.063943187006892
```

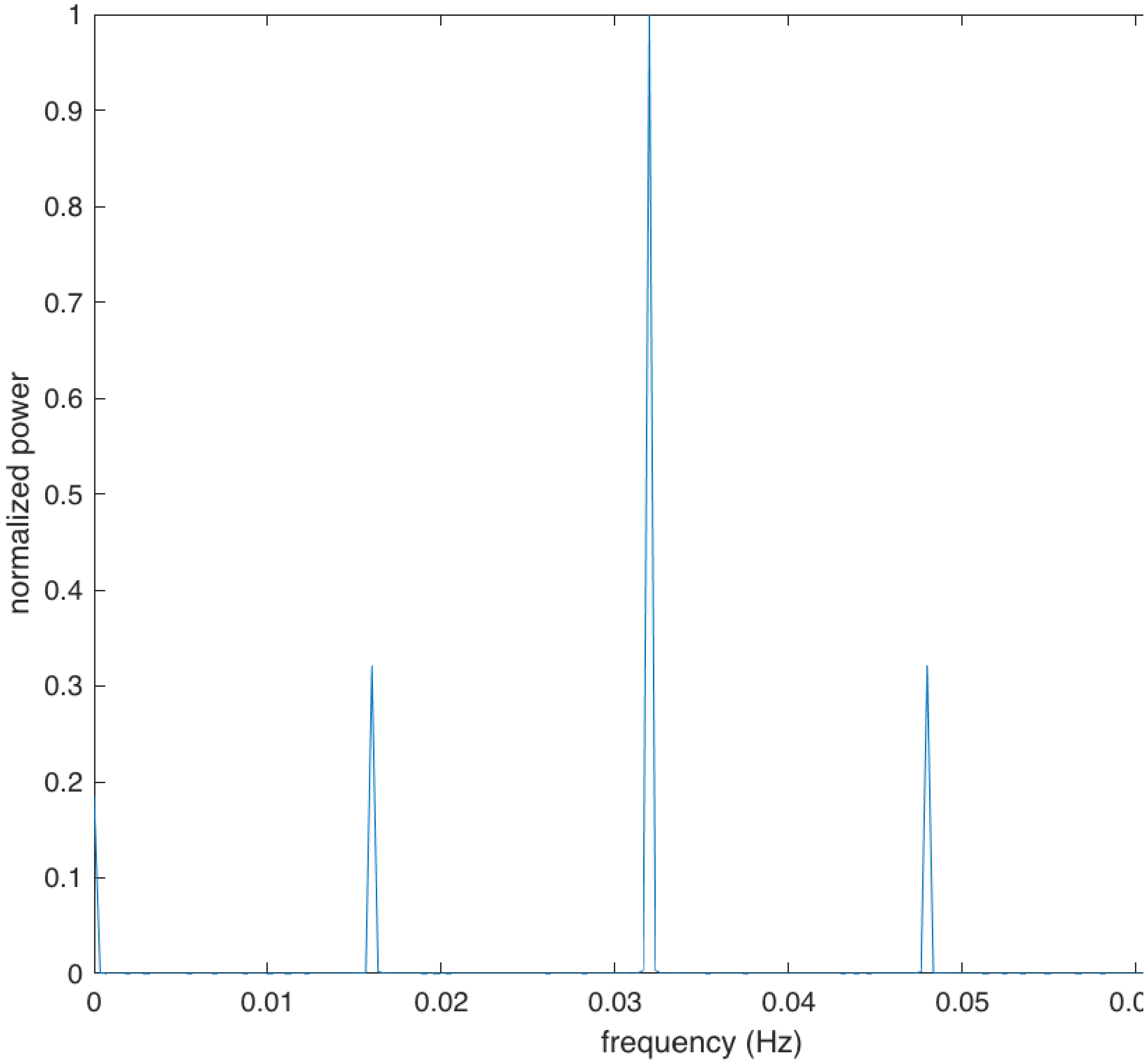
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD105521(:,1),HD105521(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

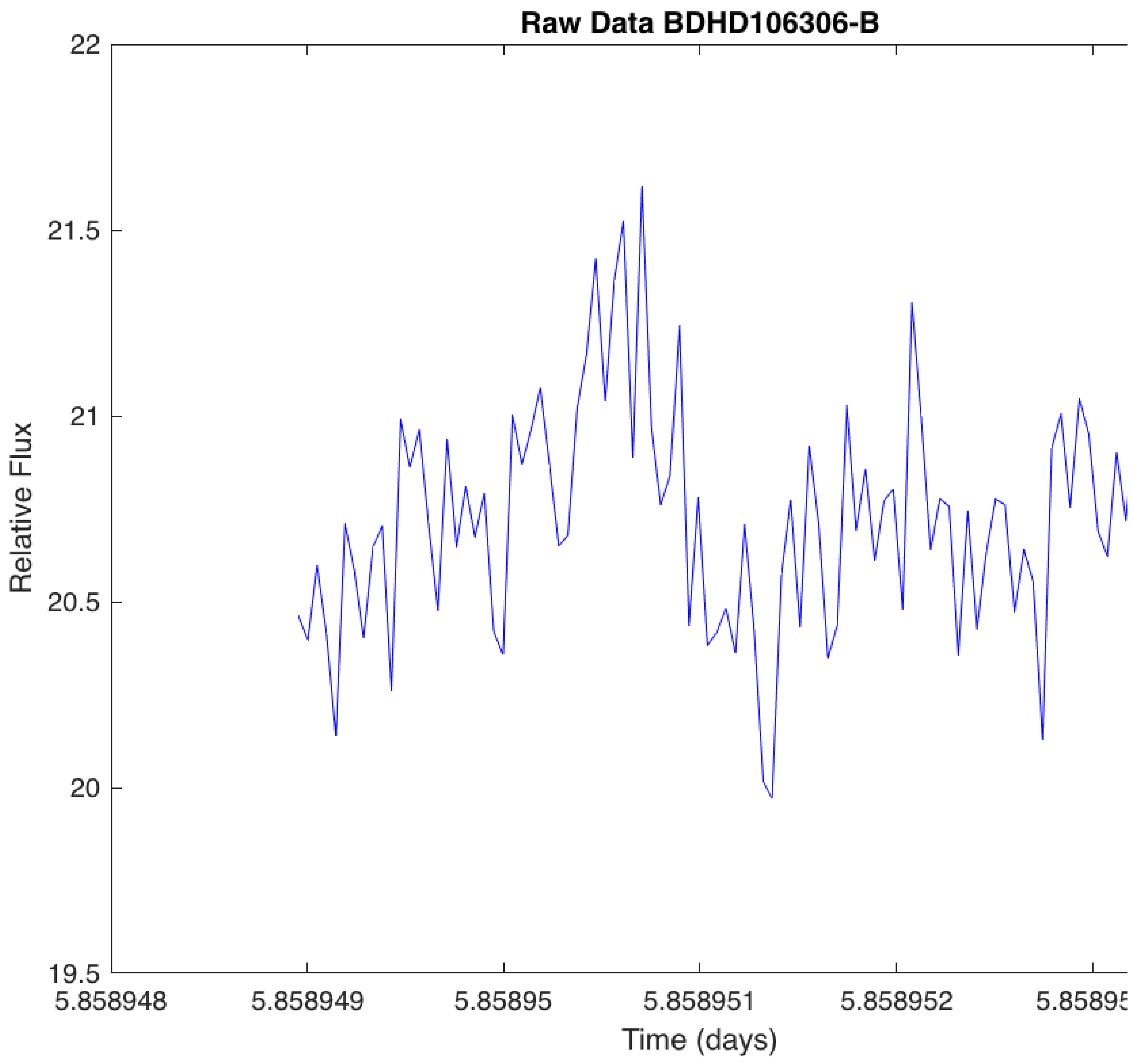
```
[Tnew,Mnew] = Interp_spline(HD105521(:,1),HD105521(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



HD106306 B

Plot the raw data

```
plot(HD106306_B(:,1),HD106306_B(:,2),'b');
title('Raw Data BDHD106306-B');
xlabel('Time (days)');
ylabel('Relative Flux');
```

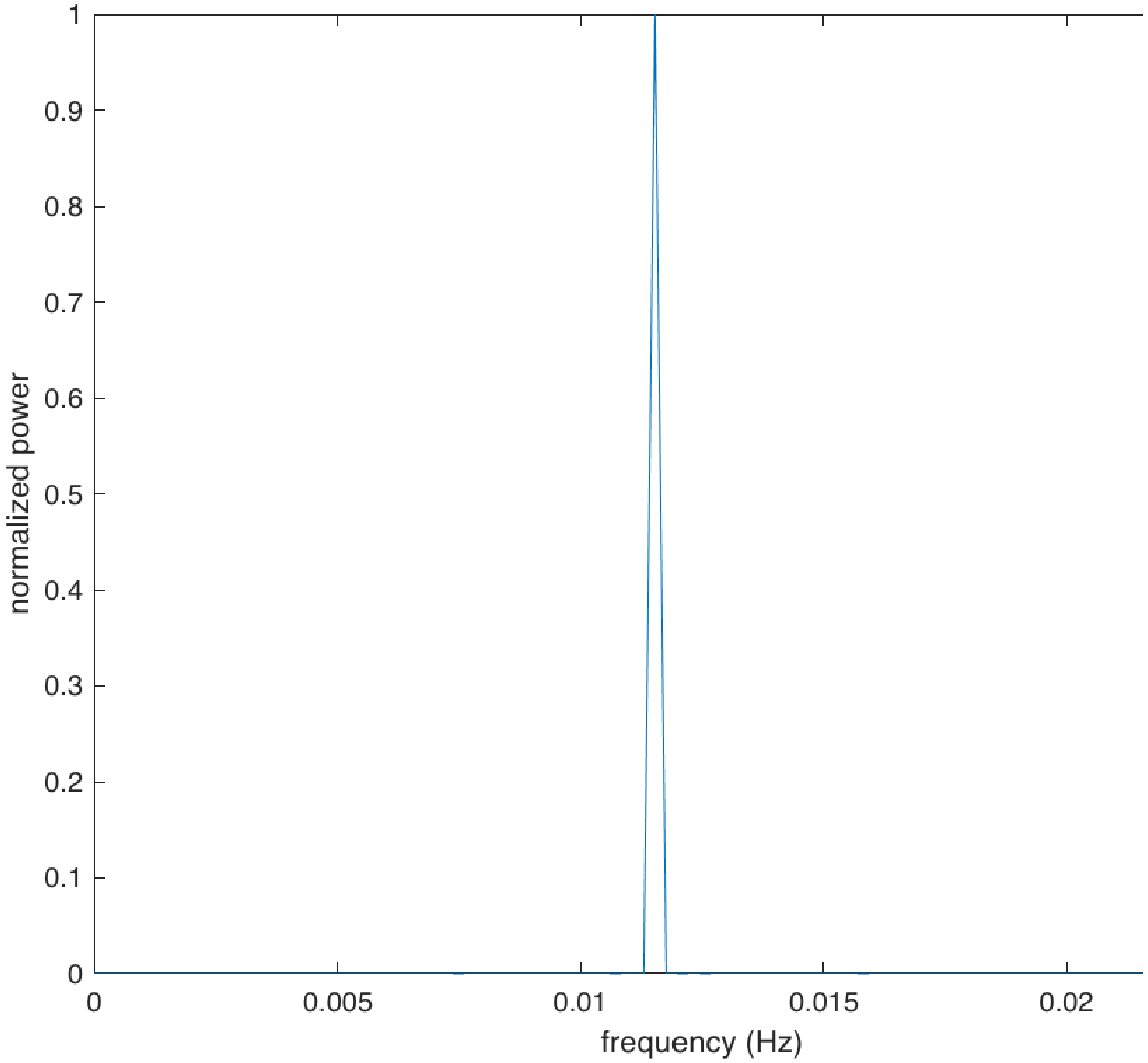


```
avg_freq =1/( mean( diff(HD106306_B(:,1)) ) *86400 )
```

```
avg_freq =
   0.023064278044436
```

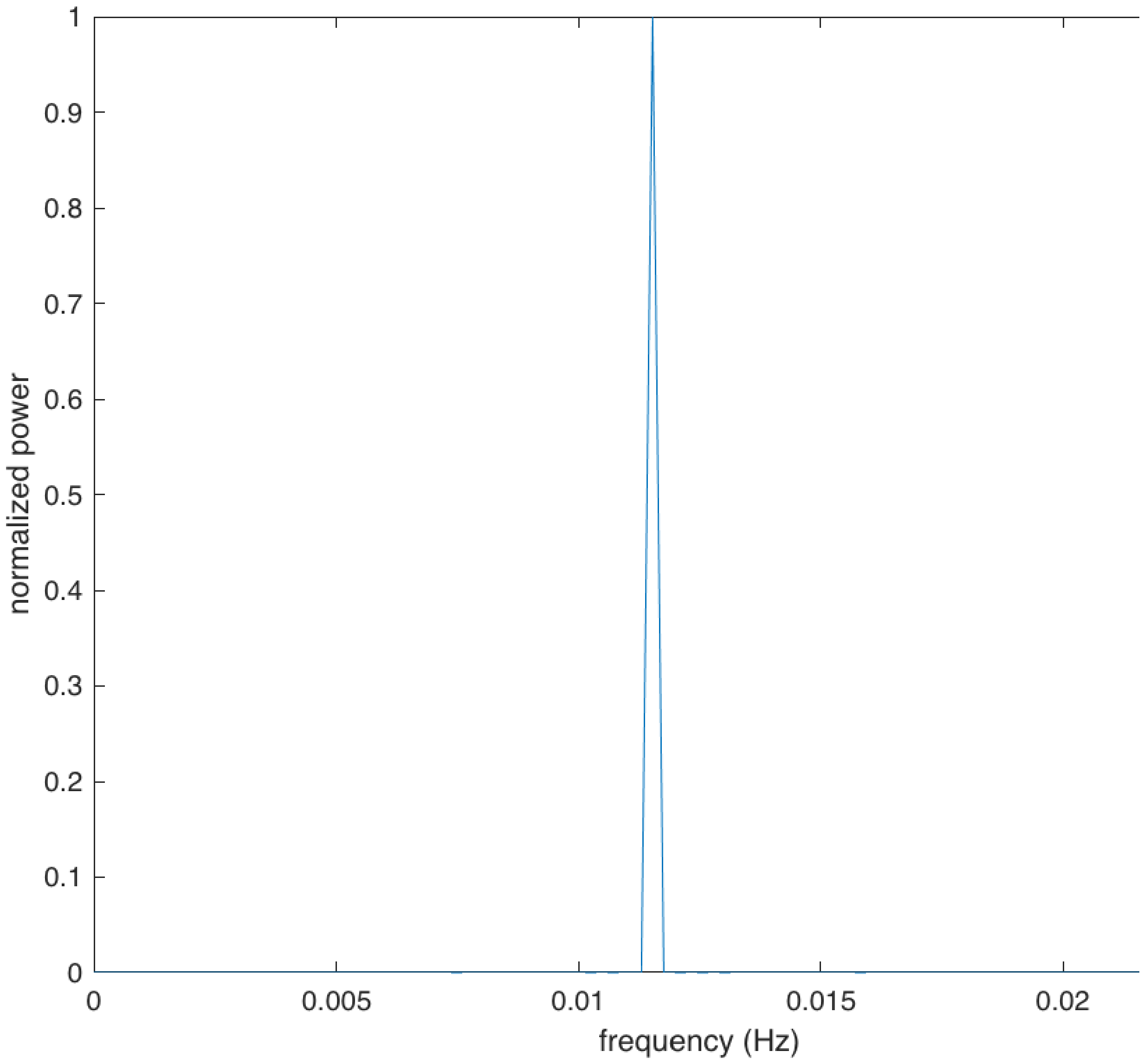
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD106306_B(:,1),HD106306_B(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

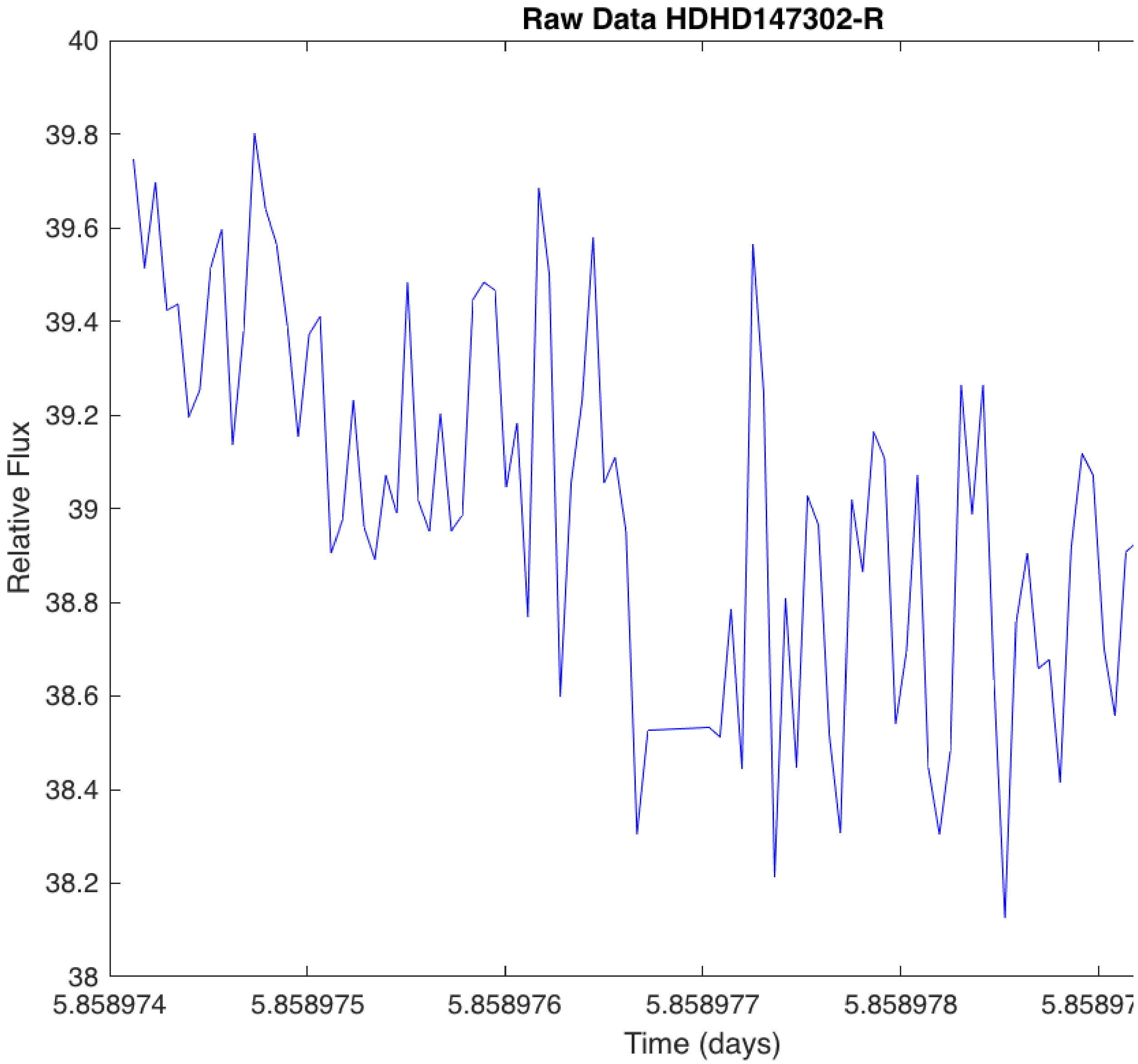
```
[Tnew,Mnew] = Interp_spline(HD106306_B(:,1),HD106306_B(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



HD147302_R

Plot the raw data

```
plot(HD147302_R(:,1),HD147302_R(:,2),'b');
title('Raw Data HDHD147302-R');
xlabel('Time (days)');
ylabel('Relative Flux');
```

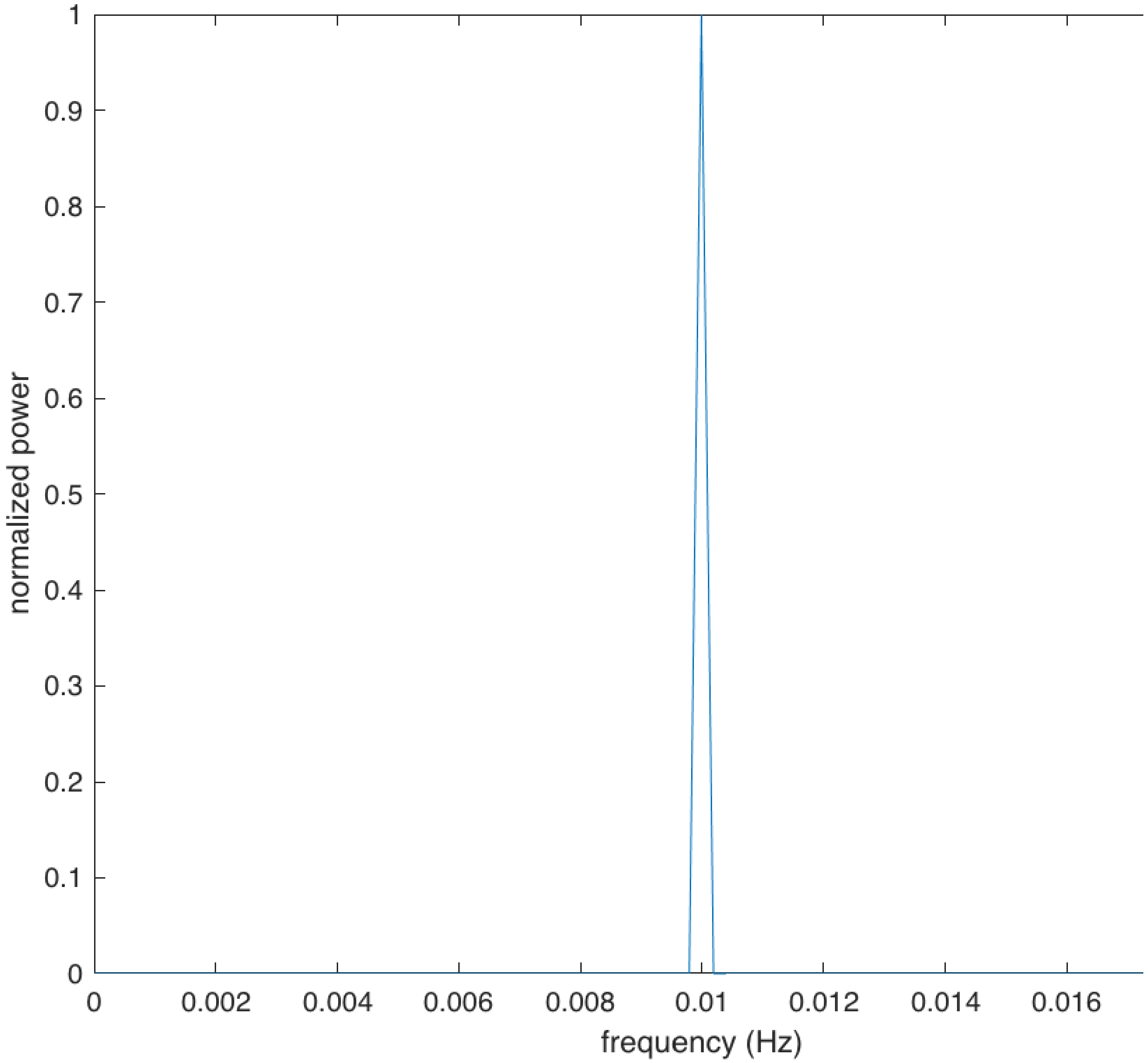


```
avg_freq =1/( mean( diff(HD147302_R(:,1)) ) *86400 )
```

```
avg_freq = 
   0.019970951344655
```

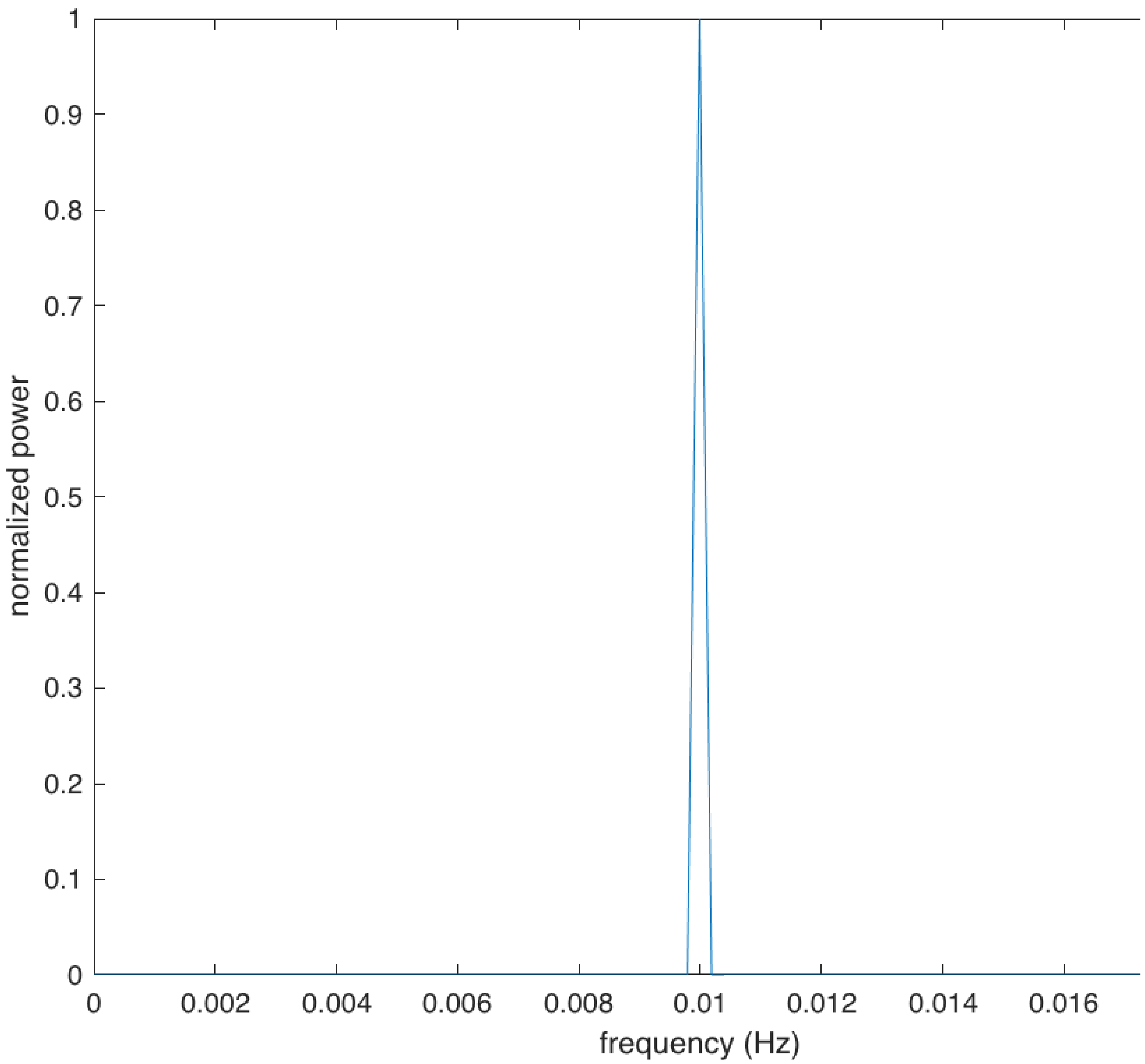
Use Linear Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_Lin(HD147302_R(:,1),HD147302_R(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Use Spline Interpolation, then use a Fourier transform to attain the power spectrum

```
[Tnew,Mnew] = Interp_spline(HD147302_R(:,1),HD147302_R(:,2));
Tnews = Tnew*86400; %convert time from days to seconds
dt = Tnews(2) - Tnews(1);
[fk,powerNor] = EnergySpec(Tnews,Mnew,dt);
```



Interpolation Functions

There are two different linear interpolation functions - the first is without built-in functions and the other is built-in. The results from both funcitons are similar.

Other interpolation methods are displayed.

Custom functions

```
function [Tnew,Mnew] = Interp_Lin(T,M)
%This is Joe's custom linear interpolation:
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

% --Calculate Mnew values--
```

```
m = numel(Tnew);
Mnew = zeros(1,m);

Mnew(1) = M(1);
Mnew(m) = M(n);

for l = 2:m-1
    for k = 1:n
        if T(k) <= Tnew(l)
            if Tnew(l) <= T(k+1)
                Mnew(l) = (M(k+1) - M(k))./(T(k+1) - T(k)).*(Tnew(l)-T(k)) + M(k);
                %eq for a line. i.e. y = mx + b
            end
        end
    end
end

end
```

Various built-in Matlab Functions

```
function [Tnew,Mnew] = Interp_nearest(T,M)
%This uses the built-in function 'interp1' with method 'nearest'
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

% --Calculate Mnew values--
Mnew = interp1(T,M,Tnew,'nearest');
end
```

```
function [Tnew,Mnew] = Interp_linear(T,M)
%This uses the built-in function 'interp1' with method 'linear'
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

%--Calculate Mnew values--
Mnew = interp1(T,M,Tnew,'linear');
end
```

```
function [Tnew,Mnew] = Interp_spline(T,M)
%This uses the built-in function 'interp1' with method 'spline'
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
```

```
Tnew = T(1):avg_dT:T(n);

% --Calculate Mnew values--
Mnew = interp1(T,M,Tnew,'spline');
end
```

```
function [Tnew,Mnew] = Interp_polyfit(T,M)
%This uses the built-in function 'interp1' with method 'polyfit'
%----

% --Sum all of the time differences between measurements--
n = numel(T);
sum = 0;
for l = 1:n-1
    sum = sum + abs(T(l+1) - T(l));
end

% --Find averaged time scale--
avg_dT = sum / (n-1);
Tnew = T(1):avg_dT:T(n);

% --Calculate Mnew values--
n = numel(T);
p = polyfit(T,M,5);
Mnew = polyval(p,Tnew);
end
```

Joe's linear combo of non-linear functions function

```
function C = NLfit(F1,F2,F3,x,y)
%documentation:  This program was developed from Gilat example 6-9

F1 = F1(x);
F2 = F2(x);
F3 = F3(x);

A(1,1) = sum(F1 .* F1);
A(1,2) = sum(F1 .* F2);
A(1,3) = sum(F1 .* F3);
A(2,2) = sum(F2 .* F2);
A(2,3) = sum(F2 .* F3);
A(3,3) = sum(F3 .* F3);

A(2,1) = A(1,2);    % A is symmetric
A(3,1) = A(1,3);
A(3,2) = A(2,3);

B(1,1) = sum(y .* F1);
B(2,1) = sum(y .* F2);
B(3,1) = sum(y .* F3);

%form is A*C=B, Therefore,

C = A\B;

end
```

Joe's power spectrum function

```
function [fk,powerNor] = EnergySpec(t,f,dt)
%this function is developed with refeernce to Gilat Program 7-4 and Example
%7-6

N = length(f);
F = fft(f)/N;
F0 = fftshift(F);

power = F0.*conj(F0)/N;

powerNor = power/max(power);
```

```
%Plot power Spectrum
fs = 1/dt;
fk = (0:N-1)*(fs/N);

plot(fk, powerNor)
xlabel('frequency (Hz)')
ylabel('normalized power')


end
```