HW 3 - ASTR 503

Created with Wolfram Mathematica 11.0 on 9-16-2016

Daniel George - dgeorge5@illinois.edu

Step 1

In[74]:=
```
n = 2048; (*Sample rate*)
```

We normalize all noises to have a standard deviation of 1.

White noise

In[75]:=
```
white = # / StandardDeviation@# &@RandomVariate[NormalDistribution[], 2 n];
```

Counts above 3σ

In[76]:=
```
Length@Select[white - Mean@white, Abs@# ≥ 3 &]
```

Out[76]= 10

Pink noise

In[77]:=
```
pink = # / StandardDeviation@# &@Take[Fourier@Join[{0.}, #, {0.}, Reverse@Conjugate@#] &[
    (Range[1. #]^-.5 RandomVariate[NormalDistribution[], #] * RandomPoint[Circle[], #] &[2 n]).
     {1., I}] // Chop, 2 n];
```

Counts above 3σ

In[78]:=
```
Length@Select[pink - Mean@pink, Abs@# ≥ 3 &]
```

Out[78]= 9

Brownian noise

In[79]:=
```
brown = # / StandardDeviation@# &@Accumulate@RandomVariate[NormalDistribution[], 2 n];
```

Counts above 3σ

In[80]:=
```
Length@Select[brown - Mean@brown, Abs@# ≥ 3 &]
```

Out[80]= 0

Thus we can see that the counts above σ is maximum for white and minimum for brown noise. This is expected because the drift increases as we move from white to pink to brown noise.

5.5Hz sinusoidal signal

In[81]:=
```
sin5p5 = 4. Sin[2. Pi 5.5 Most@Range[0., 2, 1. / n] + 2 Pi RandomReal[]];
```

Dirty 60Hz signal

In[82]:=
```
dirty60 = 2. Sin@Round[2. Pi 60 Most@Range[0., 2, 1. / n] + 2 Pi RandomReal[], 10.^-2];
```

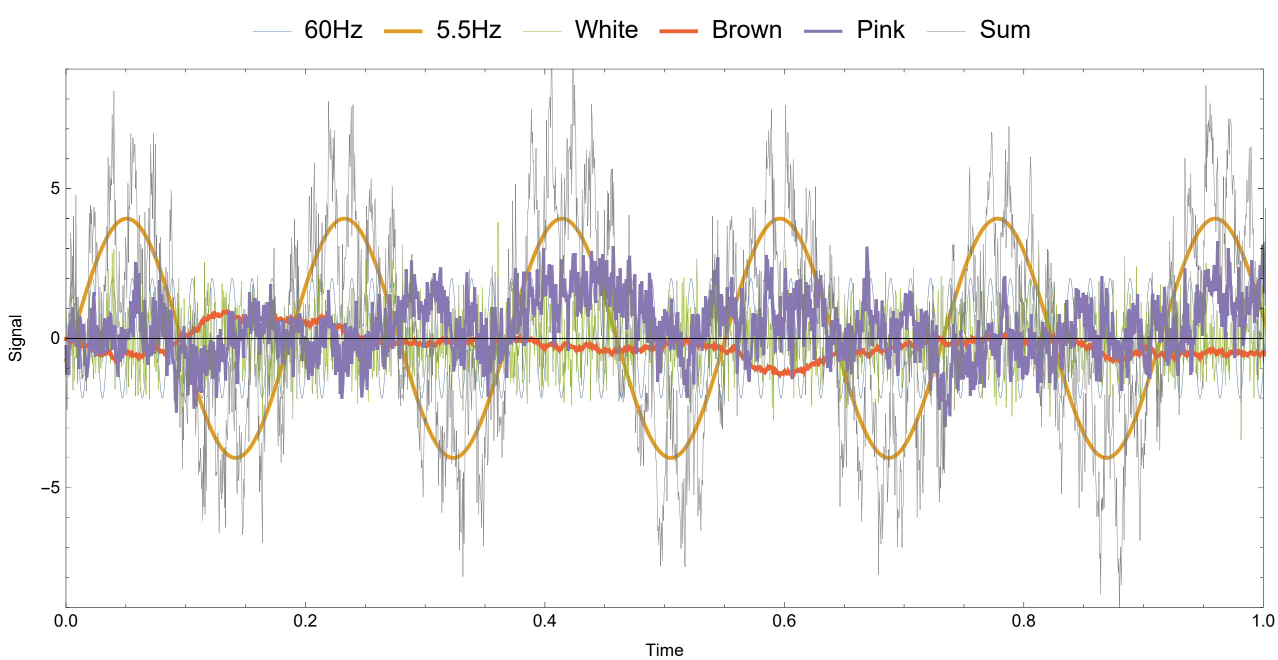
Total signal

In[83]:=
```
list = {dirty60, sin5p5, white, brown, pink}; AppendTo[list, sum = Total@list];
```

Time series plots

In[102]:=
```
ListLinePlot[list[[All, ;; 2 n]], DataRange → {0, 2 - 1. / n},
 Frame → True, PlotStyle → {Thin, Thick, Thin, Thick, Medium, {Thin, Gray}},
 ImageSize → 660, AspectRatio → .45, PlotRange → {{0, 1}, {-9, 9}},
 PlotLegends → Placed[{"60Hz", "5.5Hz", "White", "Brown", "Pink", "Sum"}, Above],
 FrameLabel → {"Time", "Signal"}]
```

Out[102]=



Step 2

Power spectrum of different noises

Finding power law fits to each periodogram

In[85]:=
```
fits = LinearModelFit[Log@Transpose@{Range[1., n / 4], Take[PeriodogramArray[#[[ ;; n]]], n / 4]},
     {x}, x] & /@ {white, brown, pink};
```

Slopes of best fit lines and their errors

In[86]:=
```
Insert[#, "±", 3] & /@
  (Join[{{"α(White) =", "α(Brown) =", "α(Pink)  ="}}, (Through@#)[[All, 2]] & /@
     fits /@ {"BestFitParameters", "ParameterErrors"}] // Transpose) // Grid
```

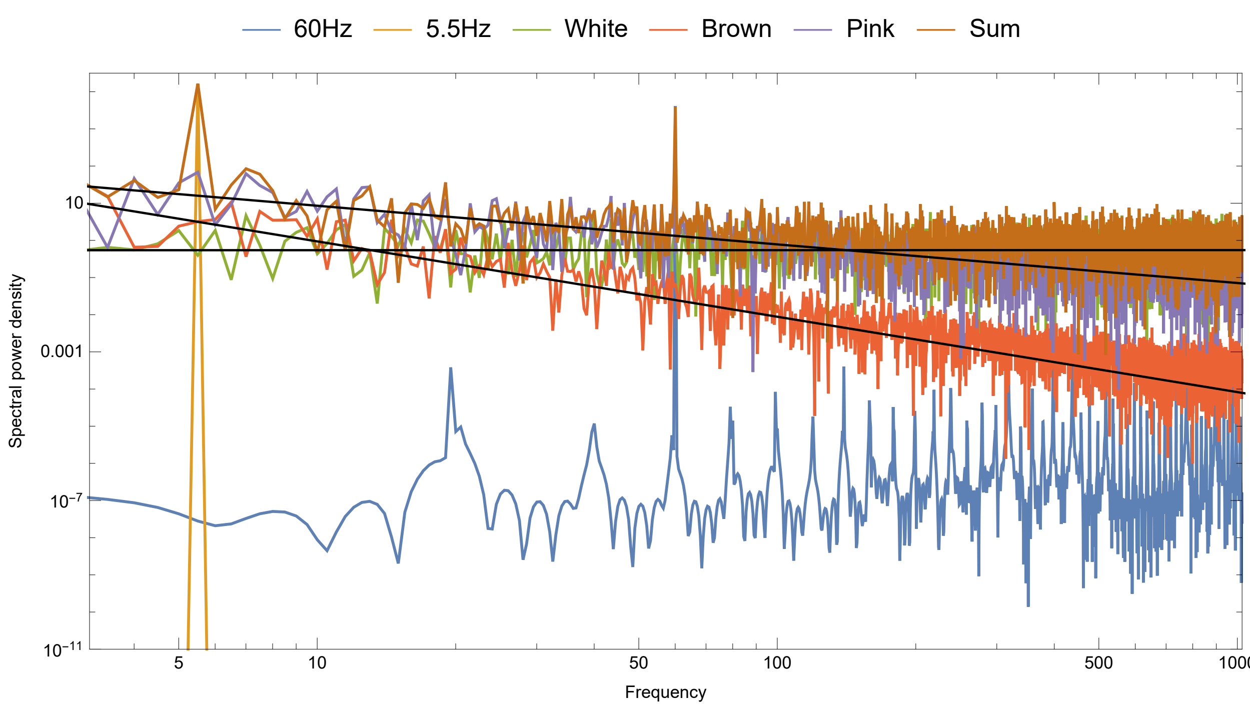
Out[86]=

α(White)	=	0.00123843222208577	±	0.058490418291661
α(Brown)	=	-2.0274452240553	±	0.05386682042377
α(Pink)	=	-1.0393853032656	±	0.060980294131712

Spectral power density plots

In[87]:=
```
Show[Periodogram[list, Frame → True,
   ImageSize → 650, AspectRatio → .5, PlotStyle → Thickness[.0025],
   PlotRange → {{3.2, n / 2}, {10^-11, 10^4.5}}, ScalingFunctions → {"Log", "Log"},
   SampleRate → n , FrameLabel → {"Frequency", "Spectral power density"},
   PlotLegends → Placed[{"60Hz", "5.5Hz", "White", "Brown", "Pink", "Sum"}, Above]],
  LogLogPlot[E^fits[[#]]@Log[x] & /@ Range[3] // Evaluate,
   {x, 0, n}, PlotStyle → {{Black, Thickness[.002]}}]]
```

Out[87]=

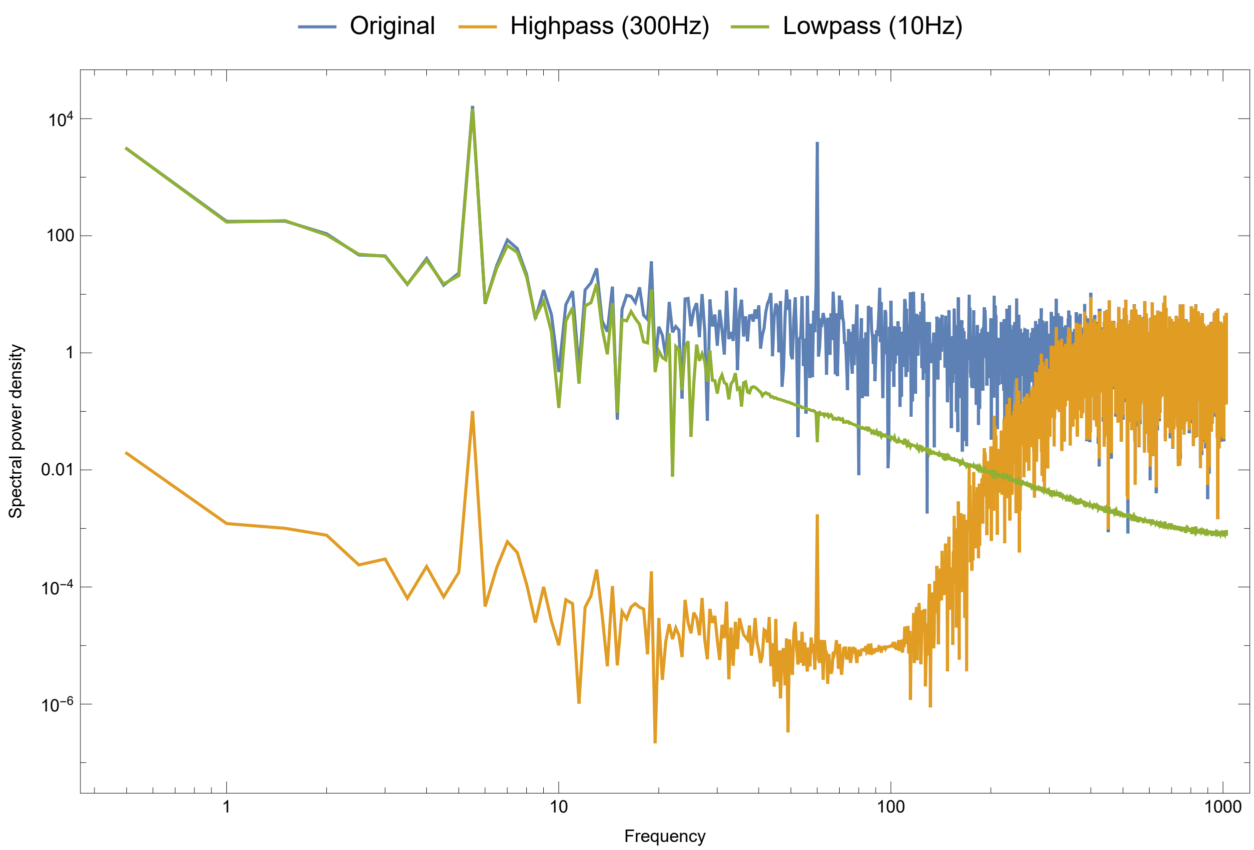


Step 3

High & low pass filters (300Hz, 10Hz)

In[88]:=
```
Periodogram[{sum, HighpassFilter[sum, 300 × 2 Pi, SampleRate → n],
  LowpassFilter[sum, 10 × 2 Pi , SampleRate → n]}, SampleRate → n,
 PlotRange → All, ScalingFunctions → {"Log", "Log"}, Frame → True, ImageSize → 650,
 PlotLegends → Placed[{"Original", "Highpass (300Hz)", "Lowpass (10Hz)"}, Above],
 FrameLabel → {"Frequency", "Spectral power density"}]
```

Out[88]=

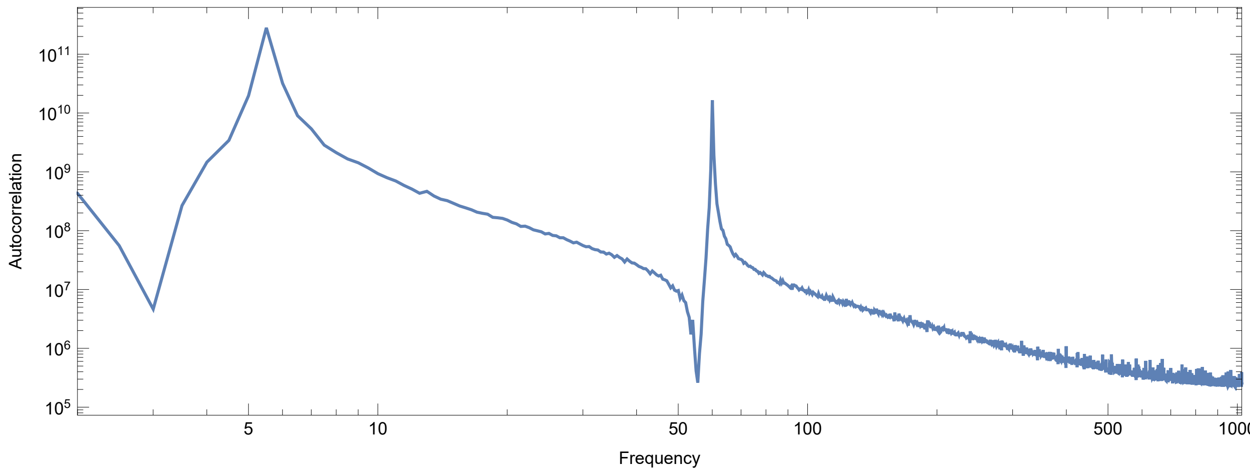


Extracting the signal

Plotting the cross-correlation of the signal with itself

In[89]:=
```
Periodogram[aCorr = ListCorrelate[sum, sum, {1, 1}, 0], ScalingFunctions → {"Log", "Log"},
  SampleRate → n, Frame → True, AspectRatio → .35, ImageSize → 650,
  PlotRange → {{2, n / 2}, All}, FrameLabel → {"Frequency", "Autocorrelation"}]
```

Out[89]=



Finding the largest peaks

In[99]:=
```
(Reverse[SortBy[FindPeaks[PeriodogramArray[aCorr][[3 ;; n]]], Last]][[1 ;; 2, 1]] + 1.) / 2 "Hz"
```

Out[99]= {5.5 Hz, 60. Hz}

Applying bandpass filter from 5Hz to 6Hz

In[106]:=
```
ListLinePlot[
  {sin5p5, LowpassFilter[HighpassFilter[sum, 2 Pi 5, SampleRate → n], 2 Pi 6, SampleRate → n]},
  Frame → True, PlotRange → {-4.1, 4.1}, PlotStyle → {{Medium}, {Red, Thick}},
  DataRange → {0, 2}, AspectRatio → .35, ImageSize → 650,
  PlotLegends → Placed[{"True signal", "Extracted signal"}, Above], FrameLabel → {"Time", "Signal"}]
```

Out[106]=