

Examples for denoising using wavelets

This Mathematica notebook uses a package for discrete wavelet transforms by P. van Fleet with examples taken from his book *Discrete Wavelet Transforms*

In[137]=

```
<< DiscreteWavelets`DiscreteWavelets`
```

SetDelayed::write : Tag Entropy in Entropy[v_] is Protected. >>

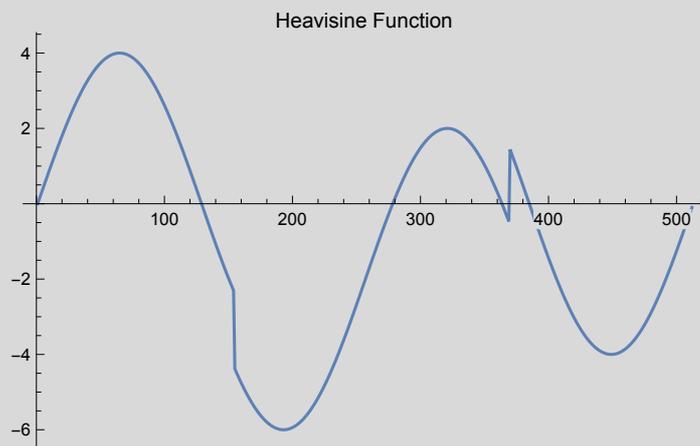
Signal and noise

In[138]=

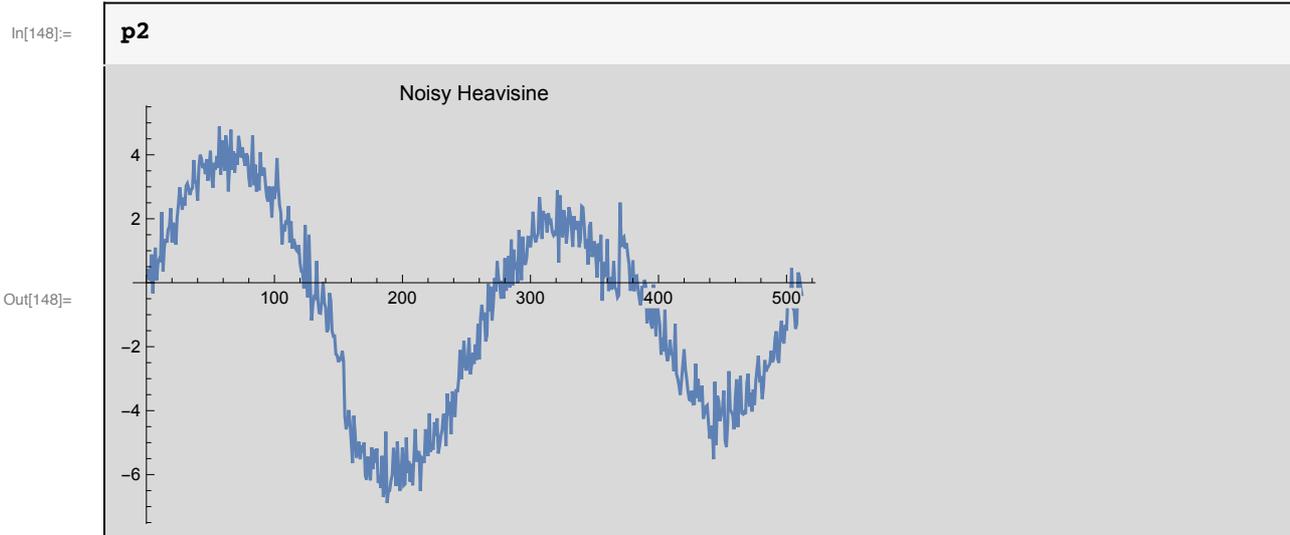
```
(* The Heavisine function evaluated 512 times on [0,1] *)  
f[t_] := 4 * Sin[4 * Pi * t] - Sign[t - .3] - Sign[.72 - t];  
v = Table[f[k / 512], {k, 0, 511}];  
p1 = ListPlot[v, Joined -> True, PlotLabel -> "Heavisine Function"];  
  
(* add some Gaussian white noise to the samples *)  
  
n = Length[v];  
nd = NormalDistribution[0., 1.];  
SeedRandom[];  
noise = Table[Random[nd], {k, 1, n}];  
w = v + 0.5 noise;  
p2 = ListPlot[w, Joined -> True, PlotLabel -> "Noisy Heavisine"];  
  
(* Compute "nit" iteration of the Haar wavelet transform. *)
```

In[147]=

p1



Out[147]=



Denoising using 3-level Haar transform

In[149]:=

```
nit = 3;
wth = HWT1D[w, NumIterations -> nit];
p3 = WaveletVectorPlot[wth,
  NumIterations -> nit, PlotLabel -> "Wavelet Transform"];
```

In[152]:=

```
(* Find the SureShrink tolerance for the highpass portion *)

wthlist = WaveletVectorToList[wth, NumIterations -> nit];

hph = Flatten[Drop[wthlist, 1]];
lambdasure = DonohoSure[hph];
Print["The tolerance to use with SureShrink on the
  highpass portion of the transform is ", lambdasure, "."];

(* Find the universal tolerance for the highpass portion *)

lambdauniv = UniversalThreshold[w, N[Haar[]], NumIterations -> nit];
Print["The universal threshold for three
  iterations of the wavelet transform is ", lambdauniv, "."];

(* Apply SureShrink and universal tolerance to hp *)

newhpsure =
  Map[ShrinkageFunction[#, lambdasure] &, Flatten[Drop[wthlist, 1]]];
newhpuniv = Map[ShrinkageFunction[#, lambdauniv] &,
  Flatten[Drop[wthlist, 1]]];
lph = First[wthlist];
newwtsure = Flatten[Join[lph, newhpsure]];
newwtuniv = Flatten[Join[lph, newhpuniv]];
denoisedsure = IHWT1D[newwtsure, NumIterations -> nit];
denoiseduniv = IHWT1D[newwtuniv, NumIterations -> nit];
```

The tolerance to use with SureShrink on the highpass portion of the transform is 2.0965.

The universal threshold for three iterations of the wavelet transform is 1.81845.

In[164]:=

```

p4sure = WaveletVectorPlot[newwtsure,
  NumIterations → nit, PlotLabel → "Sure Modified Transform"];
p4univ = WaveletVectorPlot[newwtuniv, NumIterations → nit,
  PlotLabel → "Universal Modified Transform"];
p5sure = ListPlot[denoisedsure, Joined → True, PlotLabel → "Sure-denoised"];
p5univ =
  ListPlot[denoiseduniv, Joined → True, PlotLabel → "universal-denoised"];

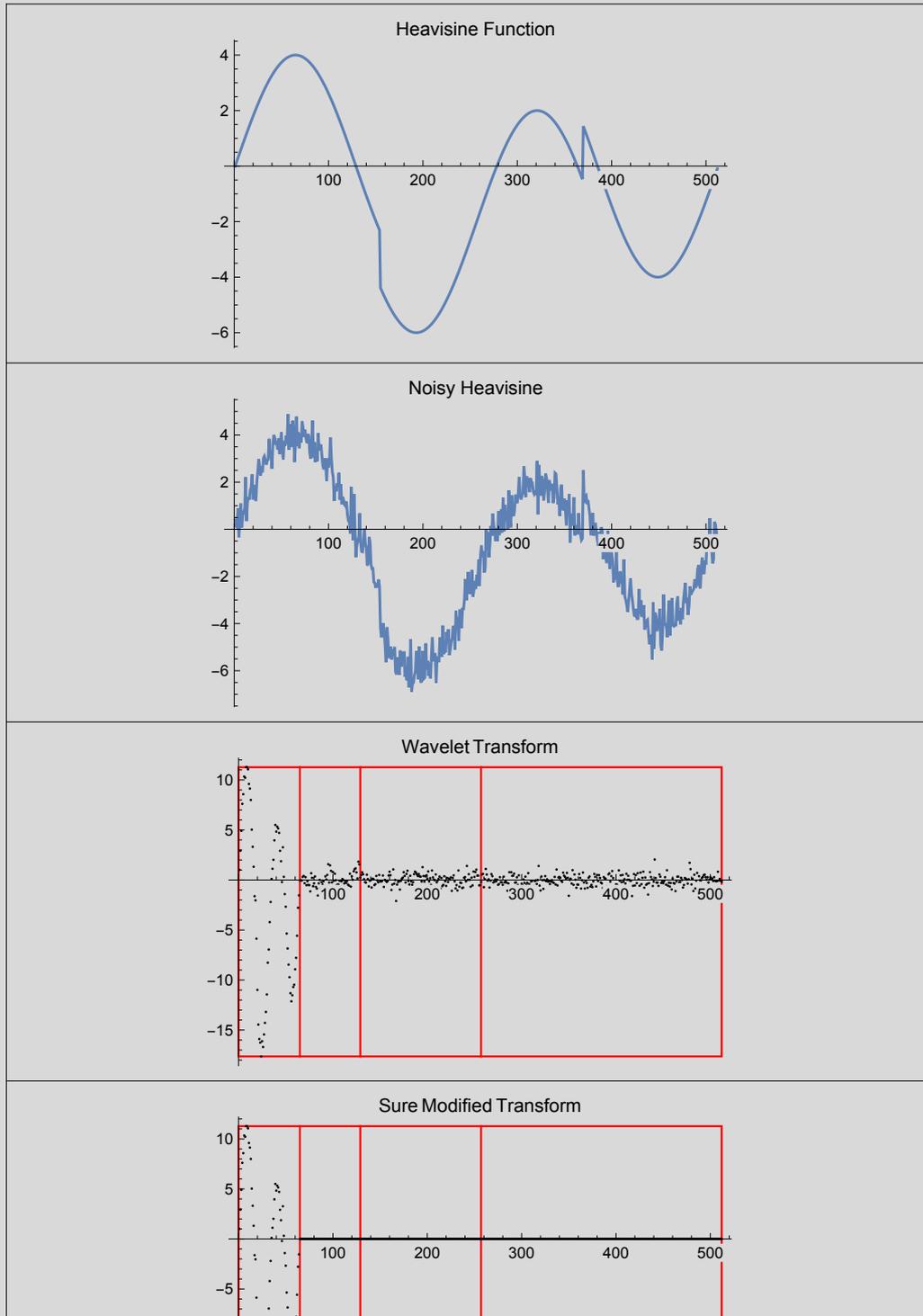
```

In[168]:=

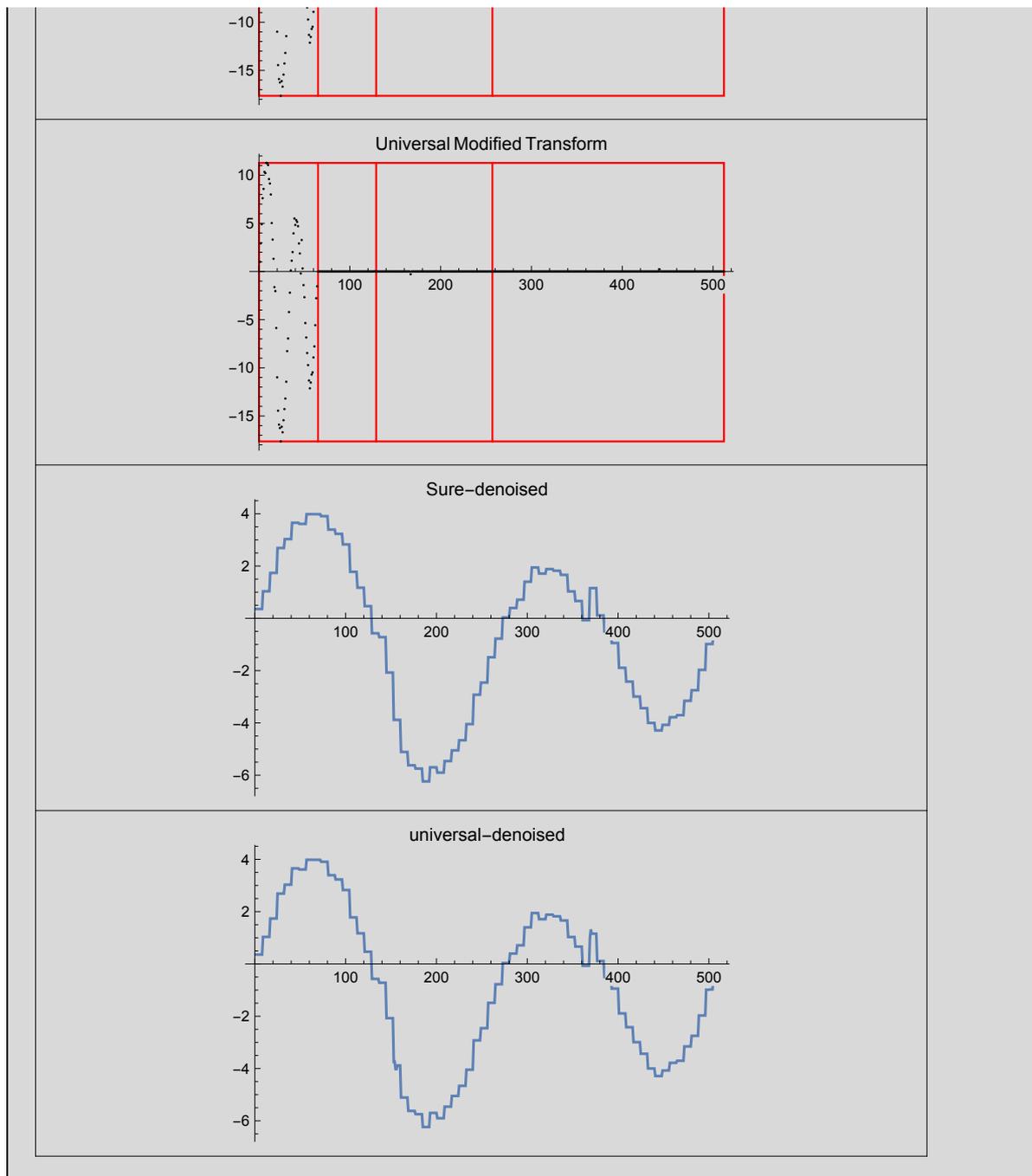
```

GraphicsGrid[{{p1}, {p2}, {p3}, {p4sure}, {p4univ}, {p5sure}, {p5univ}},
  Frame → All , ImageSize → 550]

```



Out[168]=



Denoising using 4-level D12 transform

In[169]:=

```
nit = 4;
wtd = WT1D[w, Daub[12], NumIterations → nit];
q3 = WaveletVectorPlot[wtd,
    NumIterations → nit, PlotLabel → "Wavelet Transform"];
```

In[171]:=

```

(* Find the SureShrink tolerance for the highpass portion *)

wtdlist = WaveletVectorToList[wtd, NumIterations → nit];

hpd = Flatten[Drop[wtdlist, 1]];

lambdasure = DonohoSure[hpd];
Print["The tolerance to use with SureShrink on the
      highpass portion of the transform is ", lambdasure, "."];

(* Find the universal tolerance for the highpass portion *)

lambdauniv = UniversalThreshold[w, N[Daub[12]], NumIterations → nit];
Print["The universal threshold for three
      iterations of the wavelet transform is ", lambdauniv, "."];

(* Apply SureShrink and universal tolerance to hp *)

newhpsure =
  Map[ShrinkageFunction[#, lambdasure] &, Flatten[Drop[wtdlist, 1]]];
newhpuniv = Map[ShrinkageFunction[#, lambdauniv] &,
  Flatten[Drop[wtdlist, 1]]];
lpd = First[wtdlist];
newwtsure = Flatten[Join[lpd, newhpsure]];
newwtuniv = Flatten[Join[lpd, newhpuniv]];
denoisedsure = IWT1D[newwtsure, Daub[12], NumIterations → nit];
denoiseduniv = IWT1D[newwtuniv, Daub[12], NumIterations → nit];

```

The tolerance to use with SureShrink on the highpass portion of the transform is 1.71468.

The universal threshold for three iterations of the wavelet transform is 1.96432.

In[183]:=

```

q4sure = WaveletVectorPlot[newwtsure,
  NumIterations → nit, PlotLabel → "Sure Modified Transform"];
q4univ = WaveletVectorPlot[newwtuniv, NumIterations → nit,
  PlotLabel → "Universal Modified Transform"];
q5sure = ListPlot[denoisedsure, Joined → True, PlotLabel → "Sure-denoised"];
q5univ =
  ListPlot[denoiseduniv, Joined → True, PlotLabel → "universal-denoised"];

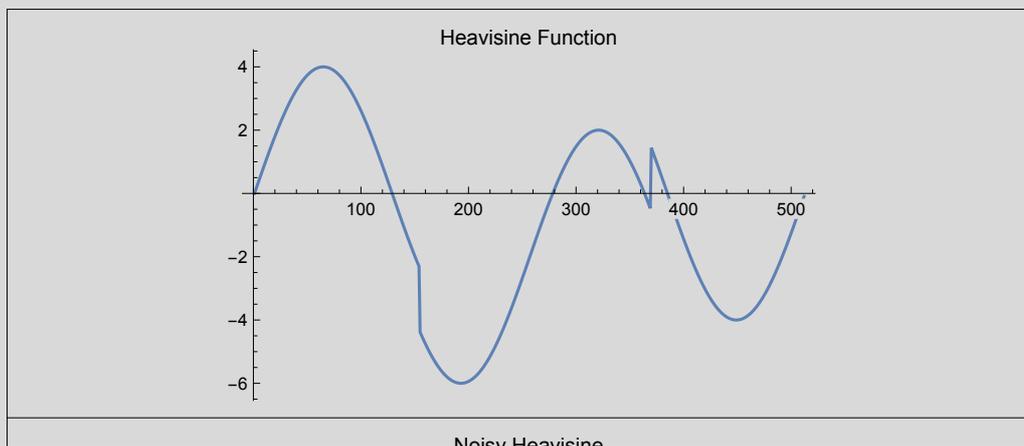
```

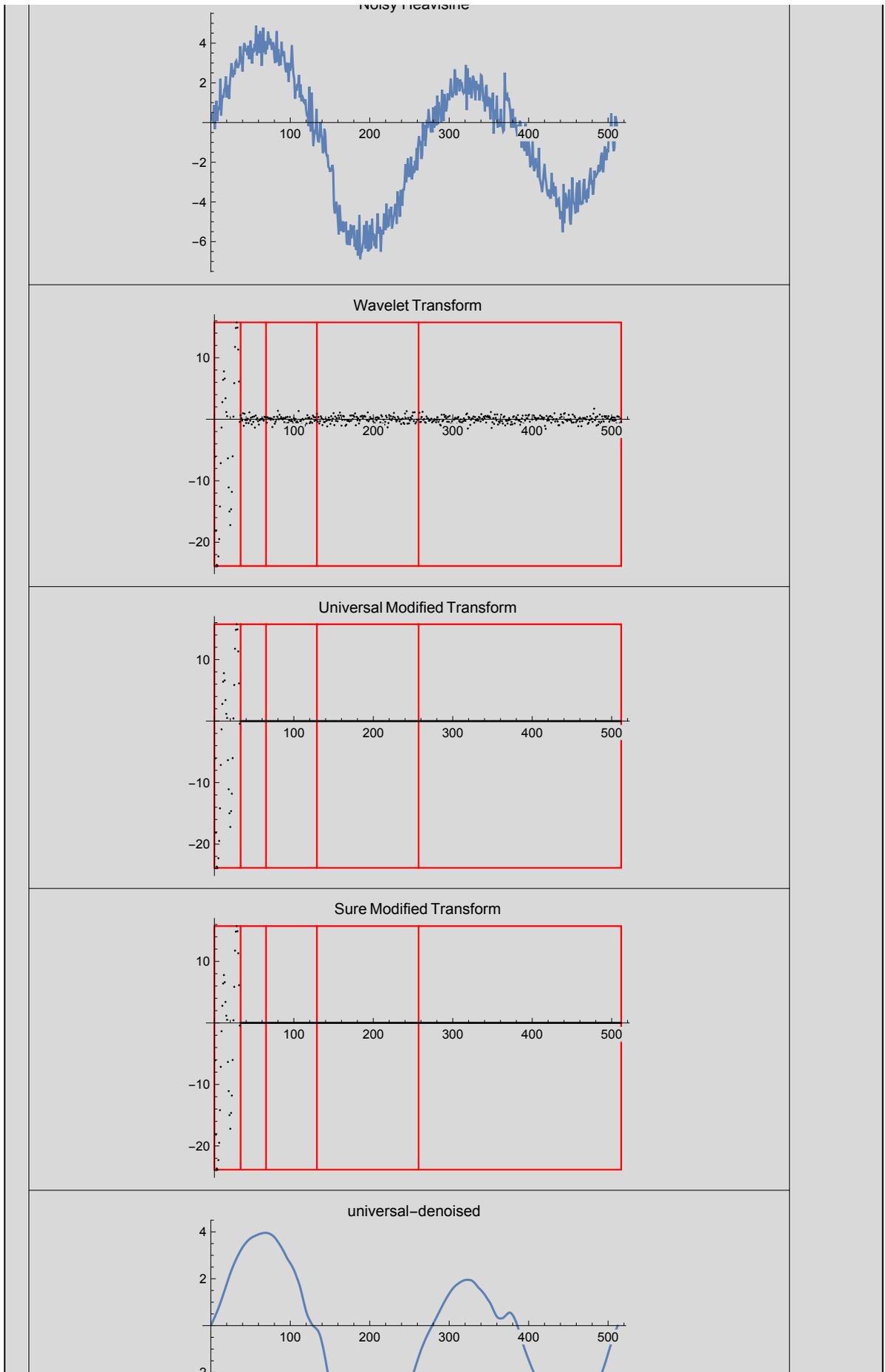
In[186]:=

```

GraphicsGrid[{{p1}, {p2}, {q3}, {q4univ}, {q4sure}, {q5univ}, {q5sure}},
  Frame → All, ImageSize → 550]

```





Out[186]=

