Application to LIGO data

February 17, 2016

0.1 Applying to the LIGO data of gravitational wave, GW150914

In [2]:
```
using Winston;
using Color;
import DSP
using PyPlot
```

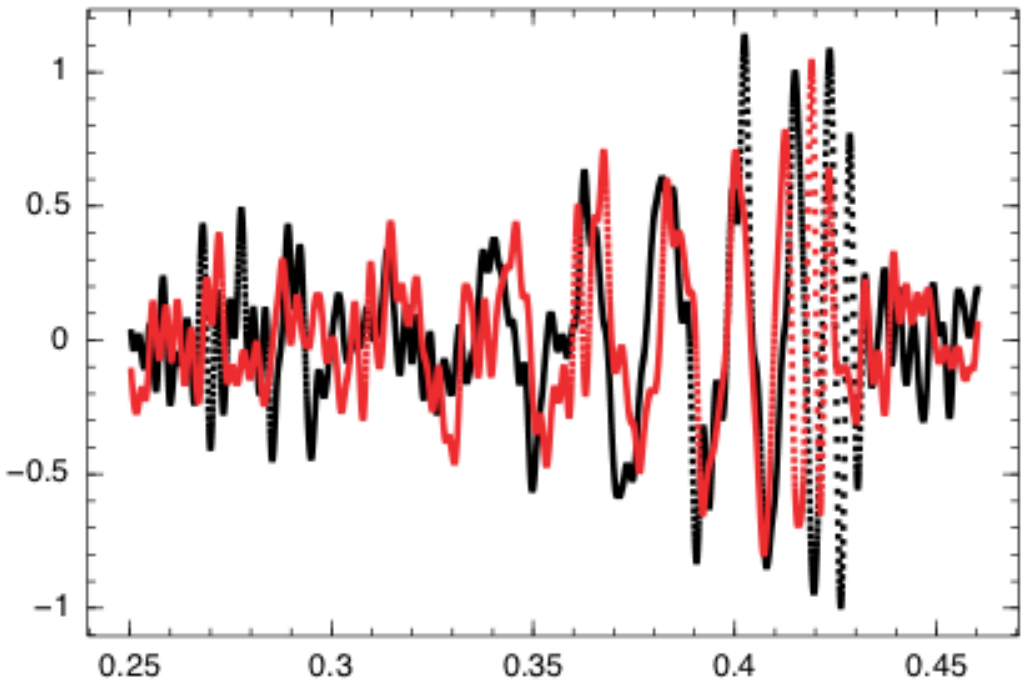
In [3]:
```
include("../juwvid.jl")
```

Out[3]: juwvid

In [5]:
```
### Using LIGO data, provided by https://losc.ligo.org/events/GW150914/
data=readdlm("fig1-observed-H.txt"); # Hanford: Get from https://ligo.caltech.edu/WA
t=data[:,1]
y=data[:,2];
data=readdlm("fig1-observed-L.txt"); # Livingston: Get from https://ligo.caltech.edu/LA
t2=data[:,1]
y2=data[:,2];
```

In [6]:
```
Winston.plot(t,y,".",t2,y2,".")
```

Out[6]:



In [13]:
```
# compute frequency indices corresponding to 0-500 Hz
dt=t[2]-t[1]
juwutils.frequency_to_index([0.0,500.0], dt, length(y))
```

Out[13]:
```
2-element Array{Int64,1}:
   1
 211
```

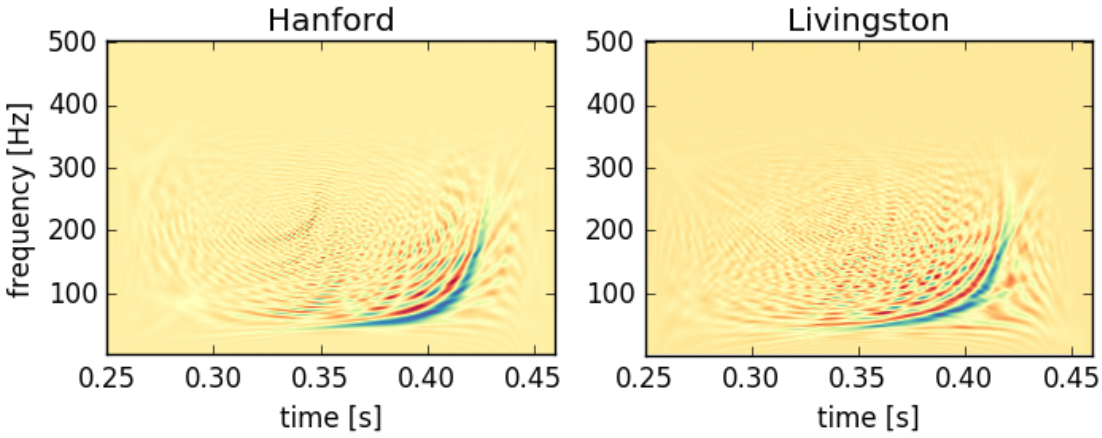
In [17]:
```
# define the frequency range
fin=collect(linspace(1.0,211.0,1024));
# compute the analytic signals
z=DSP.Util.hilbert(y);
z2=DSP.Util.hilbert(y2);
```

0.1.1 Wigner Ville Distribution

In [18]:
```
tfrwv=cohenclass.tfrwv(z,NaN,NaN,fin,NaN,0);
tfrwv2=cohenclass.tfrwv(z2,NaN,NaN,fin,NaN,0);
```

```
Single Wigner Ville
Use nufft.
Single Wigner Ville
Use nufft.
```

In [26]:
```
fig=PyPlot.figure()
ax = fig[:add_subplot](1,2,1)
a=juwplot.tfrshow(tfrwv,dt,t[1],t[end],fin[1],fin[end],0.7,"Spectral")
PyPlot.xlabel("time [s]")
PyPlot.ylabel("frequency [Hz]")
PyPlot.title("Hanford")
ax = fig[:add_subplot](1,2,2)
a=juwplot.tfrshow(tfrwv2,dt,t[1],t[end],fin[1],fin[end],0.7,"Spectral")
PyPlot.title("Livingston")
PyPlot.xlabel("time [s]")
```



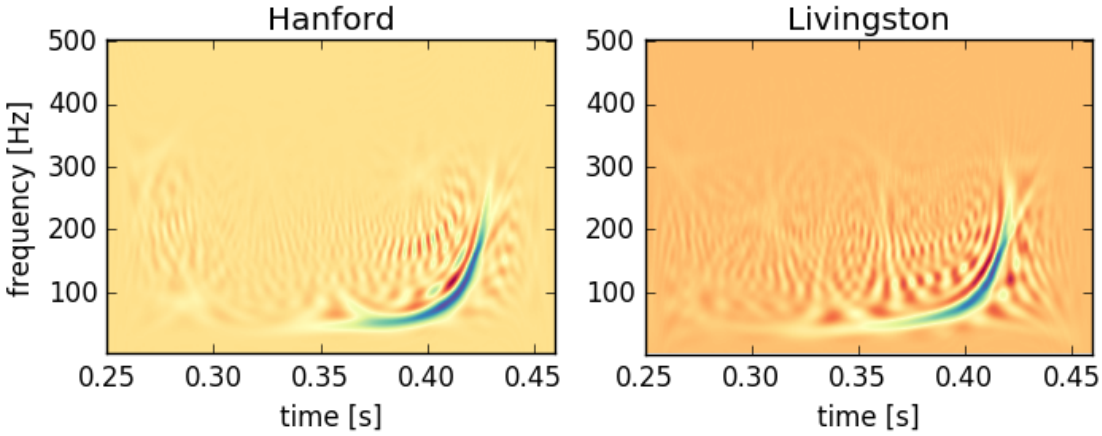
Out[26]:
```
PyObject <matplotlib.text.Text object at 0x36363a790>
```

0.1.2 Pseudo Wigner Ville Distribution

```
In [20]: tfrpwv=cohenclass.tfrpwv(z,NaN,NaN,fin,NaN,NaN,0);
         tfrpwv2=cohenclass.tfrpwv(z2,NaN,NaN,fin,NaN,NaN,0);

Single pseudo Wigner Ville
Use nufft.
Single pseudo Wigner Ville
Use nufft.

In [25]: fig=PyPlot.figure()
         ax = fig[:add_subplot](1,2,1)
         a=juwplot.tfrshow(tfrpwv,dt,t[1],t[end],fin[1],fin[end],0.7,"Spectral")
         PyPlot.xlabel("time [s]")
         PyPlot.ylabel("frequency [Hz]")
         PyPlot.title("Hanford")
         ax = fig[:add_subplot](1,2,2)
         a=juwplot.tfrshow(tfrpwv2,dt,t[1],t[end],fin[1],fin[end],0.7,"Spectral")
         PyPlot.title("Livingston")
         PyPlot.xlabel("time [s]")
```




```
Out[25]: PyObject <matplotlib.text.Text object at 0x34eb28050>
```

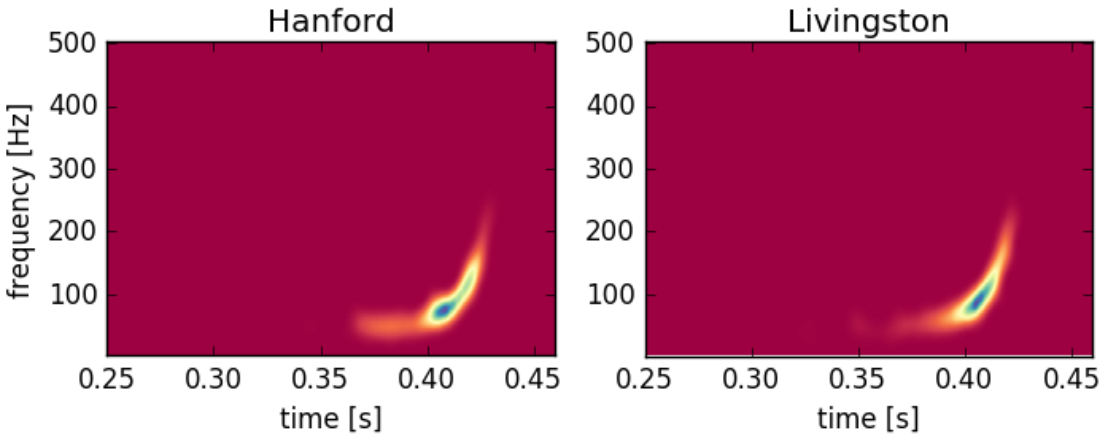
0.1.3 polynomial Wigner Ville distribution

```
In [22]: tfrpo=polywv.tfrpowv(y,NaN,NaN,fin,8,8);
         tfrpo2=polywv.tfrpowv(y2,NaN,NaN,fin,8,8);

Use nufft.
Use nufft.

In [24]: fig=PyPlot.figure()
         ax = fig[:add_subplot](1,2,1)
         a=juwplot.wtfrshow(tfrpo,dt,t[1],t[end],fin[1],fin[end],0.7,"Spectral")
         PyPlot.xlabel("time [s]")
         PyPlot.ylabel("frequency [Hz]")
         PyPlot.title("Hanford")
         ax = fig[:add_subplot](1,2,2)
```

```
a=juwplot.wtfrshow(tfrpo2,dt,t[1],t[end],fin[1],fin[end],0.7,"Spectral")
PyPlot.title("Livingston")
PyPlot.xlabel("time [s]")
```



Out[24]: PyObject <matplotlib.text.Text object at 0x34a059f50>