Wheresimple

April 20, 2015

1 WHERE1 UWB Measurement campaign M1

```
In [1]: %matplotlib inline
        from pylayers.measures.mesuwb import *
```

```
In [2]: from pylayers.measures.mesuwb import *
        from pylayers.gis.layout import *

        from pylayers.simul.link import *
        from pylayers.signal.waveform import *
```

```
WARNING:traits.has_traits:DEPRECATED: traits.has_traits.wrapped_class, 'the 'implements' class advisor h

<matplotlib.figure.Figure at 0x7f8ca4ba9790>
```

First of all we load the Layout of the environment. If the Layout associated graphs have already been built, one can load them with the `dumpr()` method.

```
In [3]: L=Layout('WHERE1.ini')
        L.dumpr()
```

```
In [4]: try:
            del td1
            del td2
            del td3
            del td4
            del te1
            del te2
            del te3
            del te4
            del tt1
            del tt2
            del tt3### Simulation section

            del tt4
        except:
            pass
```

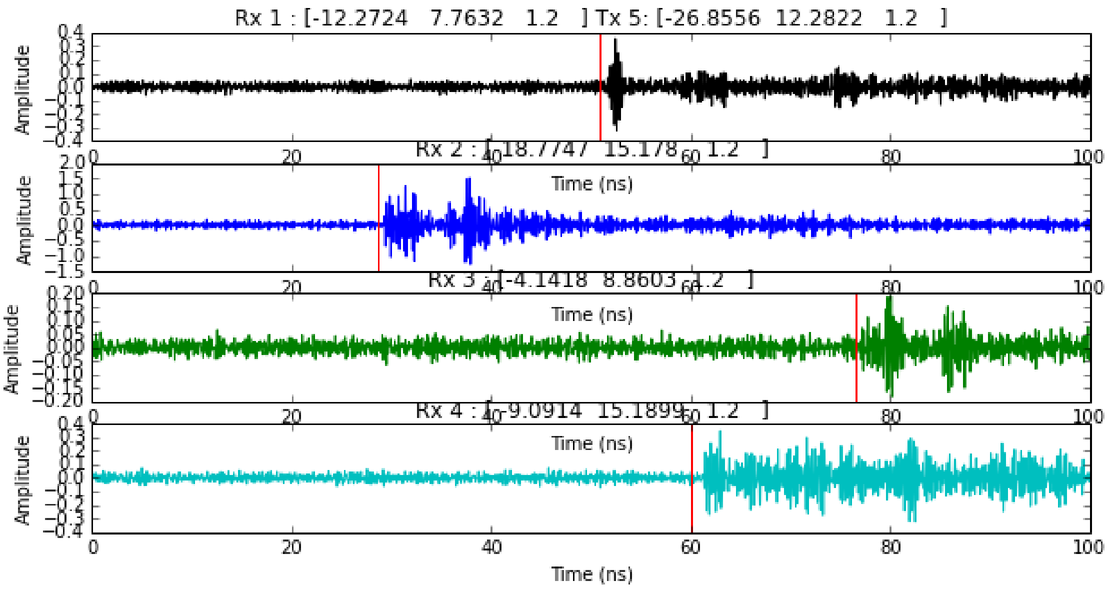
```
In [5]: K=UWBMeasure(5)
```

```
In [6]: K.de

Out[6]: array([ 50.89106921,  28.61363359,  76.5670447 ,  60.00199265])
```

```
In [7]: K.info()
```

```
Date_Time : [u'31-Jul-2008 08:14:48']
Tx_height : [u'120cm']
Tx_position : [u'P005']
Tx :  [-26.8556  12.2822   1.2   ]
------Tx1 ------
delays     (ns): 50.8910692056
range  (meters): 15.2673207617
visibility     : NLOS2
angular (degree)  : 2.84109909504
LQI Meth1 10.3676202597  (dB)
LQI Meth2 -0.0464251069027  (dB)
------Tx2 ------
delays     (ns): 28.6136335901
range  (meters): 8.58409007702
visibility     : NLOS2
angular (degree)  : 3.48568781284
LQI Meth1 15.5920243795  (dB)
LQI Meth2 7.02848427115  (dB)
------Tx3 ------
delays     (ns): 76.5670446987
range  (meters): 22.9701134096
visibility     : NLOS2
angular (degree)  : 2.99206422733
LQI Meth1 15.8266138647  (dB)
LQI Meth2 1.72677266474  (dB)
------Tx4 ------
delays     (ns): 60.0019926459
range  (meters): 18.0005977938
visibility     : NLOS
angular (degree)  : 3.30383704128
LQI Meth1 28.4222937655  (dB)
LQI Meth2 6.01984060663  (dB)
```

```
In [8]: ### Simulation section
        fig=plt.figure(figsize=(10,5))
        K.show(delay=K.de)
```



```
Out[8]: (<matplotlib.figure.Figure at 0x7f8ca017d350>,
         <matplotlib.axes.AxesSubplot at 0x7f8ca0cb3150>)
```

```
In [9]: K.toa_new
```

```
Out[9]: <bound method UWBMeasure.toa_new of Date_Time : 31-Jul-2008 08:14:48
        Tx_height : 120cm
        Tx_position :P005
        Tx : [-26.8556  12.2822   1.2   ]
        >
```

```
In [10]: K.tau_Emax()
```

```
Out[10]: array([[ 52.44 ],
                [ 37.825],
                [ 80.03 ],
                [ 62.935]])
```

```
In [11]: np.vstack((K.rx))
```

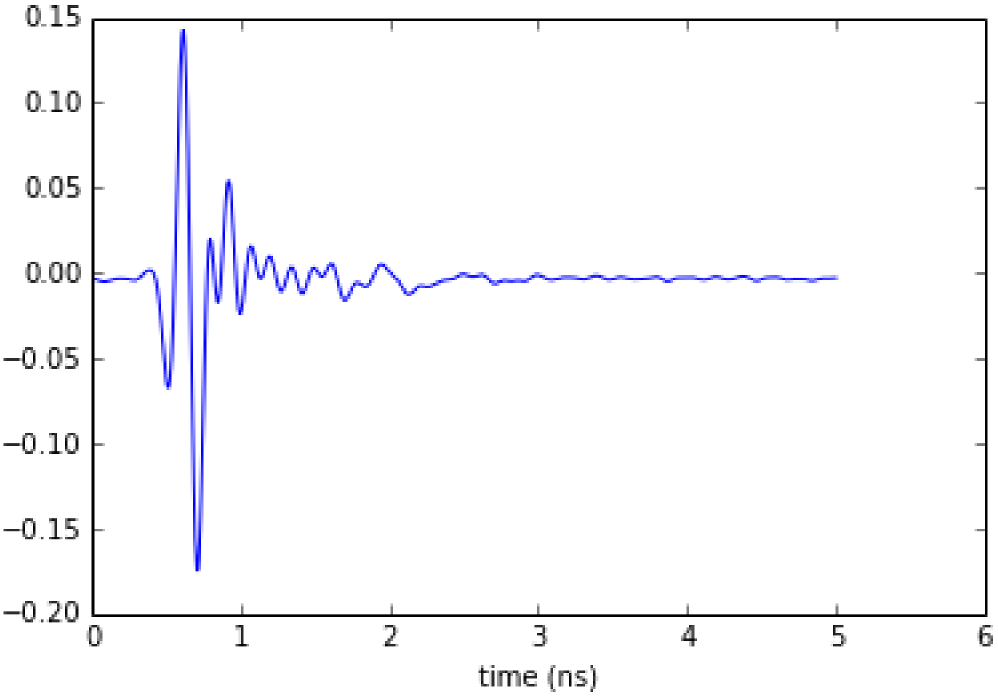
```
Out[11]: array([[  0.    ,   0.    ,   1.2   ],
                [-12.2724,   7.7632,   1.2   ],
                [-18.7747,  15.178 ,   1.2   ],
                [ -4.1418,   8.8603,   1.2   ],
                [ -9.0914,  15.1899,   1.2   ]])
```

The code below reads data from the M1-WHERE2 measurement campaign.

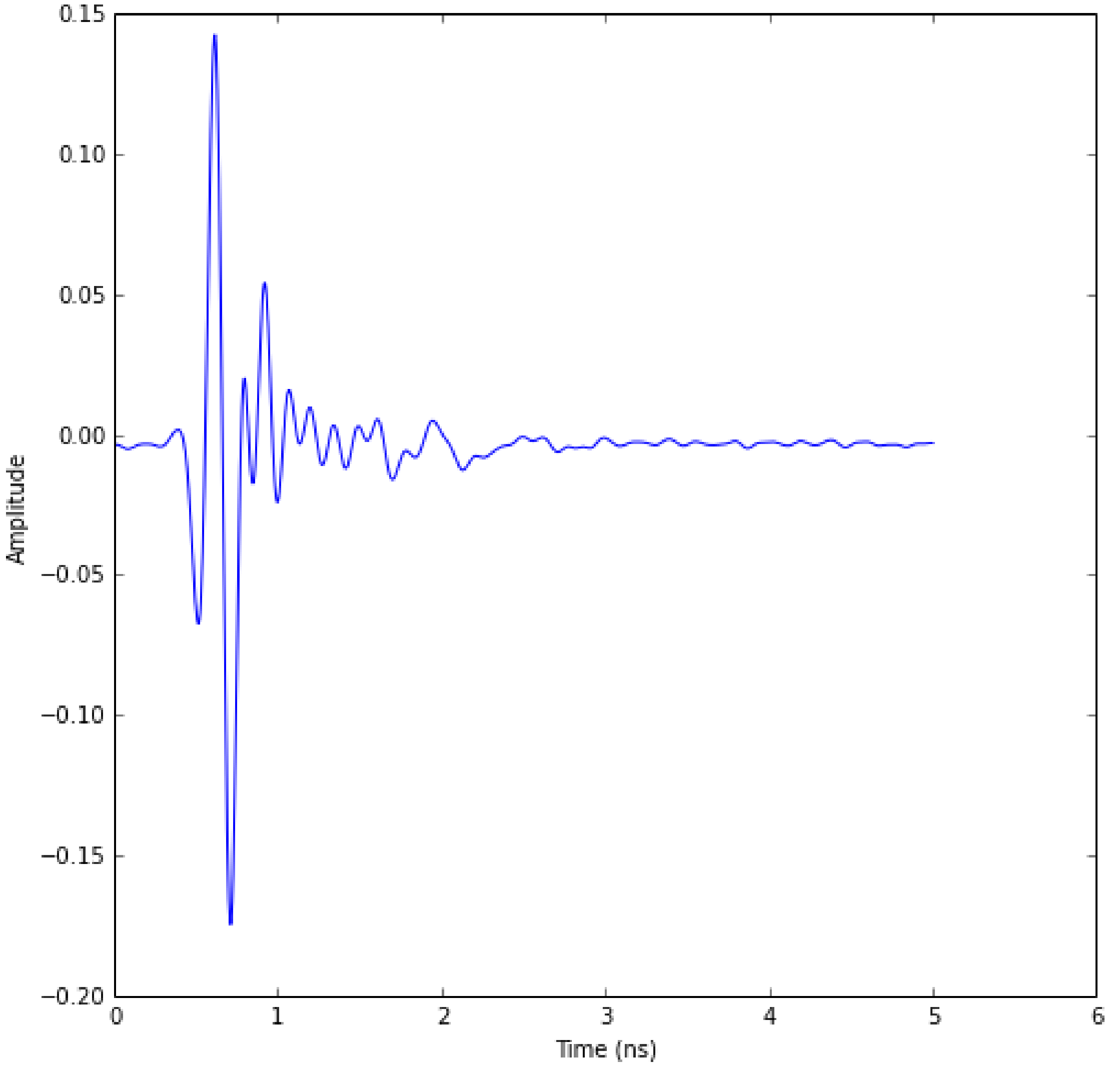
```
In [12]: for k in range(300):
             try:
                 M  = UWBMeasure(k)
                 tx = M.tx
                 D  = M.rx-tx[np.newaxis,:]
                 D2 = D*D
                 dist = np.sqrt(np.sum(D2,axis=1))[1:]
                 Emax = M.Emax()
                 Etot = M.Etot()[0]
                 try:
                     td1 = np.hstack((td1,dist[0]))
                     td2 = np.hstack((td2,dist[1]))
                     td3 = np.hstack((td3,dist[2]))
                     td4 = np.hstack((td4,dist[3]))

                     te1 = np.hstack((te1,Emax[0]))
                     te2 = np.hstack((te2,Emax[1]))
                     te3 = np.hstack((te3,Emax[2]))
                     te4 = np.hstack((te4,Emax[3]))

                     tt1 = np.hstack((tt1,Etot[0]))
                     tt2 = np.hstack((tt2,Etot[1]))
                     tt3 = np.hstack((tt3,Etot[2]))
                     tt4 = np.hstack((tt4,Etot[3]))
                     #tdist = np.hstack((tdist,dist))
                     #te = np.hstack((te,Emax))
                 except:
                     td1=np.array(dist[0])
                     td2=np.array(dist[1])
                     td3=np.array(dist[2])
                     td4=np.array(dist[3])
                     te1 =np.array(Emax[0])
                     te2 =np.array(Emax[1])
                     te3 =np.array(Emax[2])
                     te4 =np.array(Emax[3])
                     tt1 =np.array(Etot[0])
                     tt2 =np.array(Etot[1])
                     tt3 =np.array(Etot[2])
                     tt4 =np.array(Etot[3])
             except:
                 pass
```

The IR-UWB applied waweform is available in the raw data structure and can be extracted as follow. This exracttion is important in order to proceeed to the ray tracing simulation with the same waveform as the one used in the measurement campaign.

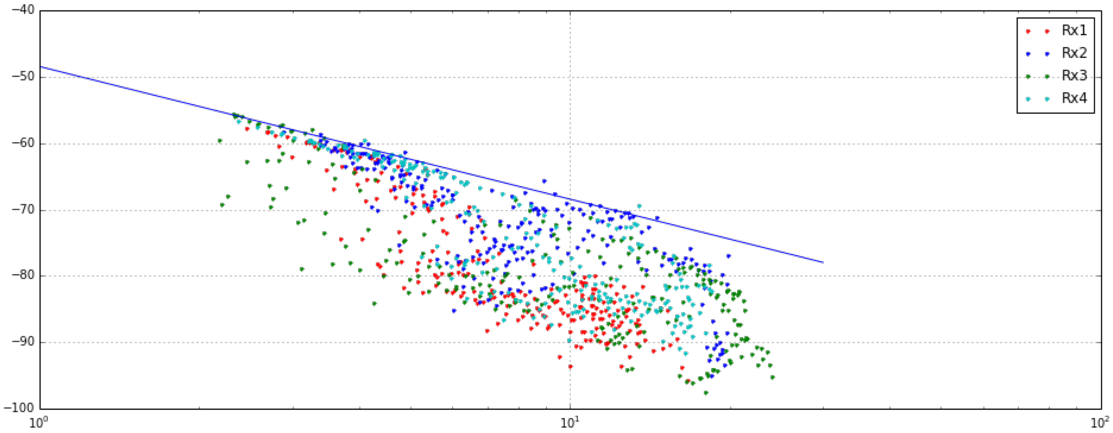
```
In [13]: from pylayers.signal.bsignal import *
         s=M.RAW_DATA.tx[0]
         t=M.RAW_DATA.timetx[0]*1e9
         plt.plot(t,s)
         plt.xlabel('time (ns)')
         se=TUsignal(t,s)
```



```
In [14]: te = t[1]-t[0]
         cs = np.cumsum(s*s)
         E = cs[-1]*te
         EdB = 10*np.log10(E*30)
         print EdB
         print E*30
         use =1/E
         print use

-10.2361907016
0.0947067492189
316.767286888

In [15]: E2=se.Emax()
         print E2*30
         E2dB=10*np.log10(E2*30)
         print E2dB

0.0918920633424
-10.3672199673

In [16]: se.plot(typ='v')

Out[16]: (<matplotlib.figure.Figure at 0x7f8ca0e0c690>,
          array([[<matplotlib.axes.AxesSubplot object at 0x7f8ca09c0cd0>]], dtype=object))
```

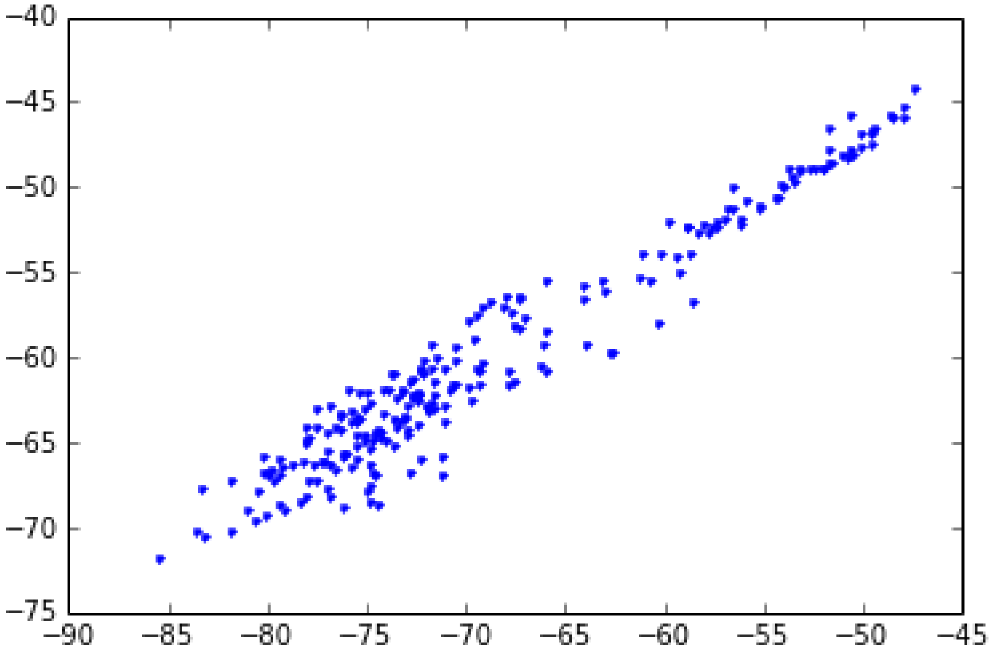


```
In [17]: fig = plt.figure(figsize=(16,6))
         ax = fig.add_subplot(111)
         ax.semilogx(td1,te1+EdB,'.r',label='Rx1')
         ax.semilogx(td2,te2+EdB,'.b',label='Rx2')
         ax.semilogx(td3,te3+EdB,'.g',label='Rx3')
         ax.semilogx(td4,te4+EdB,'.c',label='Rx4')
         d = np.linspace(1,30,100)

         LFS = -(32.4+20*np.log10(4)+20*np.log10(d))-4
         ax.semilogx(d,LFS)
         plt.legend()
         plt.grid()
```



```
In [18]: plt.plot(te1,tt1,'.')
```

```
Out[18]: [<matplotlib.lines.Line2D at 0x7f8ca0aeb190>]
```

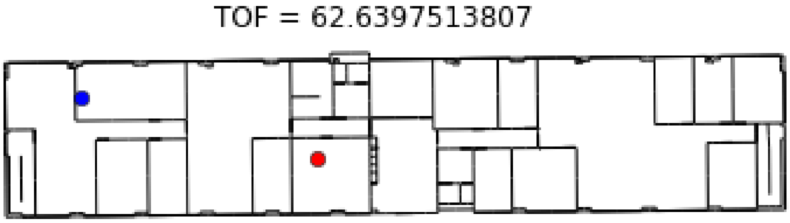


```
In [19]: M.Etot()
```

```
Out[19]: (array([-67.62048799, -64.56362576, -54.22863588, -66.40678426]),)
```

```
In [20]: #measure id
         tx_id = 100 #in M.valid_index
         rx_id = 3 #1,2,3,4
         M=UWBMeasure(tx_id)
         TX = M.tx
         RX =M.rx[rx_id]
```

```
In [21]: TX

Out[21]: array([-22.3797,  13.3897,   1.2   ])

In [22]: M.rx

Out[22]: array([[  0.    ,   0.    ,   1.2   ],
                [-12.2724,   7.7632,   1.2   ],
                [-18.7747,  15.178 ,   1.2   ],
                [ -4.1418,   8.8603,   1.2   ],
                [ -9.0914,  15.1899,   1.2   ]])

In [23]: L.showG('s',figsize=(8,4))
         plt.plot(TX[0],TX[1],'ob')
         plt.plot(RX[0],RX[1],'or')
         plt.title('TOF = '+ str(np.sqrt(np.sum((TX-RX)**2))/0.3))

Out[23]: <matplotlib.text.Text at 0x7f8ca06e0b50>
```



```
In [24]: Lk = DLink(L=L,a=TX,b=RX,cutoff=4,verbose=False)
         Lk.Aa=Antenna('defant.vsh3')
         Lk.Ab=Antenna('defant.vsh3')

In [25]: Lk.eval(force=['ray','Ct','H'],alg=5)
         #f,a = Lk.show(rays=True,labels=False)

Out[25]: (array([  9.79138753e-05,   2.96104973e-04,   1.59308395e-04,
                   2.27580461e-05,   1.34574073e-04,   4.82827749e-04,
                   1.18095058e-03,   1.27977423e-04,   1.34909317e-04,
                   5.48270185e-05,   9.81119862e-05,   3.61269565e-04,
                   2.73004108e-04,   1.24319829e-04,   2.28615744e-04,
                   2.76183582e-04,   2.32071461e-04,   2.86885128e-04,
                   5.08065400e-04,   4.16943960e-05,   9.25585146e-05,
                   3.67562947e-05,   3.26755640e-05,   7.18843013e-05,
                   5.98546831e-05,   5.35179821e-05,   8.90165410e-05,
                   3.64178931e-04,   3.64157003e-04,   1.49754933e-04,
                   9.73558942e-05,   4.65239631e-05,   5.86669038e-05,
                   2.05936803e-04,   9.48731849e-05,   2.36321979e-04,
                   6.88923268e-04,   1.24565228e-04,   4.69513933e-05,
                   2.31194688e-05,   2.94412022e-05,   2.94227854e-05,
                   5.60825032e-05,   7.29054840e-05,   1.70818338e-04,
                   1.70818449e-04,   5.34747295e-05,   4.74216436e-05,
                   1.14993846e-04,   5.08387934e-05]),
          array([  67.6920456 ,   67.6920456 ,  116.7880677 ,  116.7880677 ,
                   68.63195952,   89.0838292 ,   62.63975138,   69.09664151,
                   69.67313591,   70.48894853,   75.87644272,   82.1700362 ,
                   89.4423201 ,   89.8884232 ,   91.14669303,  102.17774607,
                   63.14854276,   63.77882449,   64.67003199,   70.94146788,
                   71.48668315,   71.48668315,   71.5030899 ,   76.29701541,
                   76.81949336,   82.55855407,   83.04164527,   91.30130681,
                   91.30130681,   91.49710187,   91.93323475,  102.49044732,
                  102.87998732,  115.06282092,   65.1629729 ,   65.75514013,
                   65.75514013,   65.7739541 ,   73.27135774,   73.27135774,
                   78.46804802,   78.46804802,   84.56899461,   84.56899461,
                   93.31516303,   93.31516303,  104.11672196,  104.11672196,
                  115.34059458,  115.68687375]))
```

```
In [26]: #%timeit Lk.eval(force=True,alg=7,cutoff=3)
         #f,a = Lk.show(rays=True,labels=False)
```

```
In [27]: Lk.R
```

```
Out[27]: Rays3D
         ----------
         8 / 4 : [0 1 2 3]
         4 / 2 : [4 5]
         5 / 10 : [ 6  7  8  9 10 11 12 13 14 15]
         6 / 18 : [16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33]
         7 / 16 : [34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49]
         -----
         ni : 310
         nl : 670
```

```
In [28]: #%timeit Lk.eval(force=True,alg=7,threshold=0.01)
         #f,a = Lk.show(rays=True,labels=False)
```

```
In [29]: Lk.Si.keys()
```

```
Out[29]: [3, 4, 5, 6, 7, 8, 9, 10]
```

```
In [30]: U=Lk.R[4]['sig2d'][0]
```

```
In [31]: print U.shape
```

```
(2, 4, 2)
```

```
In [32]: s1 = U[:,:,0]
         print s1
```

```
[[328 335  67  73]
 [  2   3   3   3]]
```

```
In [33]: from pylayers.antprop.signature import Signature
```

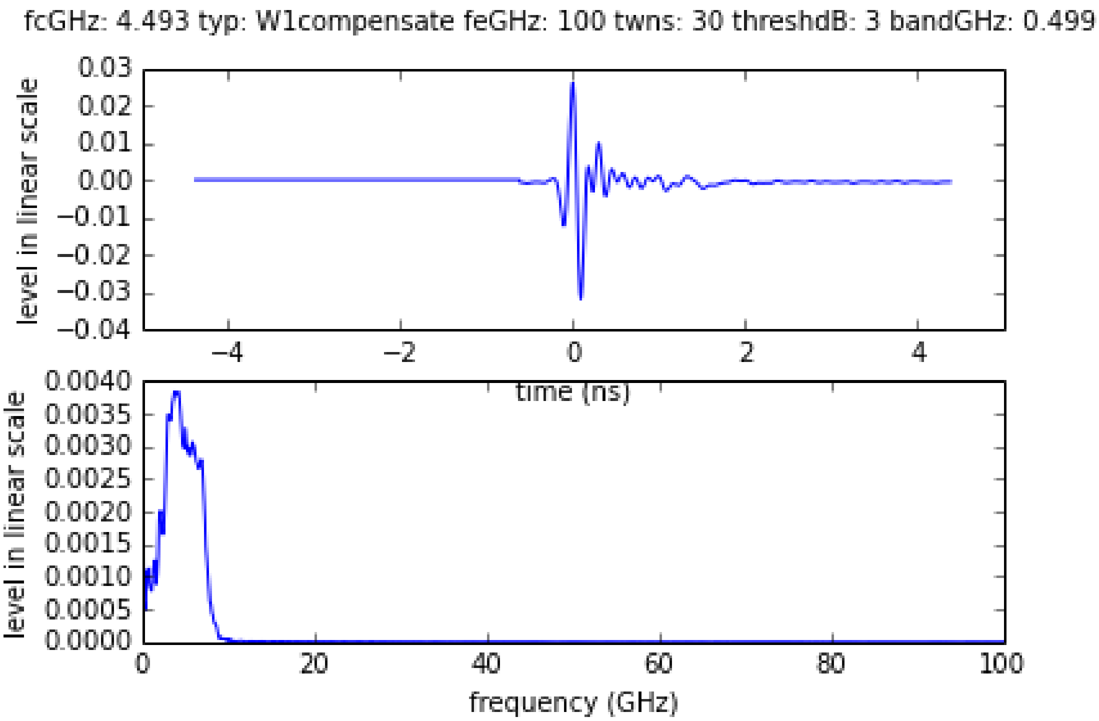
```
In [34]: S=Signature(s1)
```

```
In [35]: S
```

```
Out[35]: [328 335  67  73]
         [2 3 3 3]
```

```
In [36]: wav = Waveform(typ='W1compensate')
```

```
In [37]: wav.show()
```



```
In [38]: #ir = Lk.H.applywavB(wav.sfg)
```

```
In [39]: Lk.H.isFriis
```

```
Out[39]: True
```

```
In [40]: if Lk.H.isFriis:
             ir = Lk.H.applywavB(wav.sf)
         else:
             ir = Lk.H.applywavB(wav.sfg)
```

```
In [41]: Lk.R.los
```

```
Out[41]: False
```

```
In [42]: Lk.H.ak
```

```
Out[42]: array([  9.79138753e-05,   2.96104973e-04,   1.59308395e-04,
                  2.27580461e-05,   1.34574073e-04,   4.82827749e-04,
                  1.18095058e-03,   1.27977423e-04,   1.34909317e-04,
                  5.48270185e-05,   9.81119862e-05,   3.61269565e-04,
                  2.73004108e-04,   1.24319829e-04,   2.28615744e-04,
                  2.76183582e-04,   2.32071461e-04,   2.86885128e-04,
                  5.08065400e-04,   4.16943960e-05,   9.25585146e-05,
                  3.67562947e-05,   3.26755640e-05,   7.18843013e-05,
                  5.98546831e-05,   5.35179821e-05,   8.90165410e-05,
                  3.64178931e-04,   3.64157003e-04,   1.49754933e-04,
                  9.73558942e-05,   4.65239631e-05,   5.86669038e-05,
                  2.05936803e-04,   9.48731849e-05,   2.36321979e-04,
                  6.88923268e-04,   1.24565228e-04,   4.69513933e-05,
                  2.31194688e-05,   2.94412022e-05,   2.94227854e-05,
                  5.60825032e-05,   7.29054840e-05,   1.70818338e-04,
                  1.70818449e-04,   5.34747295e-05,   4.74216436e-05,
                  1.14993846e-04,   5.08387934e-05])
```

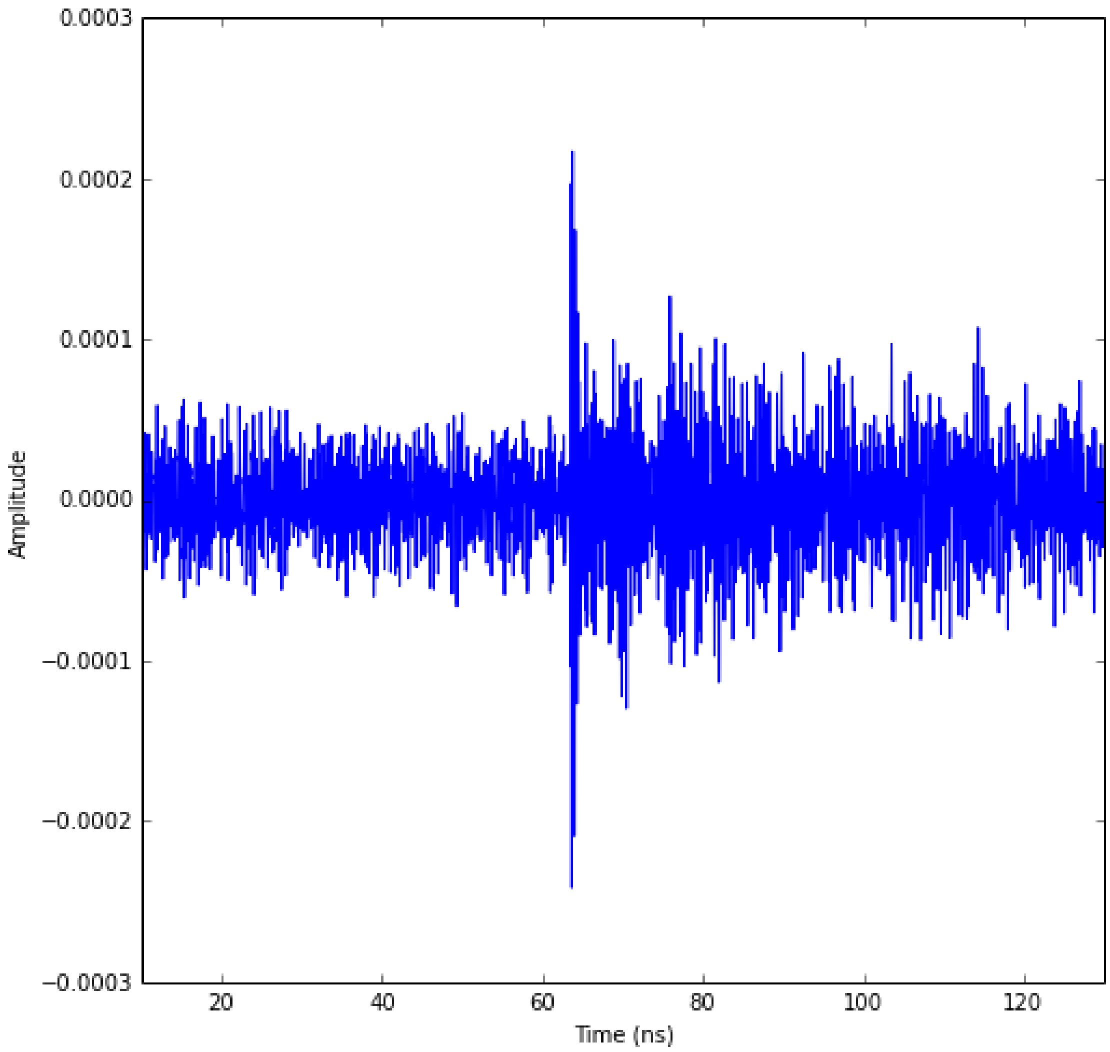
```
In [43]: Lk.H.taud
```

```
Out[43]: array([  67.6920456 ,   67.6920456 ,  116.7880677 ,  116.7880677 ,
                  68.63195952,   89.0838292 ,   62.63975138,   69.09664151,
                  69.67313591,   70.48894853,   75.87644272,   82.1700362 ,
                  89.4423201 ,   89.8884232 ,   91.14669303,  102.17774607,
                  63.14854276,   63.77882449,   64.67003199,   70.94146788,
                  71.48668315,   71.48668315,   71.5030899 ,   76.29701541,
                  76.81949336,   82.55855407,   83.04164527,   91.30130681,
                  91.30130681,   91.49710187,   91.93323475,  102.49044732,
                 102.87998732,  115.06282092,   65.1629729 ,   65.75514013,
                  65.75514013,   65.7739541 ,   73.27135774,   73.27135774,
                  78.46804802,   78.46804802,   84.56899461,   84.56899461,
                  93.31516303,   93.31516303,  104.11672196,  104.11672196,
                 115.34059458,  115.68687375])
```

```
In [44]: G=Lk.H.ift()
```

```
In [45]: M.tdd.ch3.plot(typ='v')
         plt.xlim([10,130])
```

```
Out[45]: (10, 130)
```



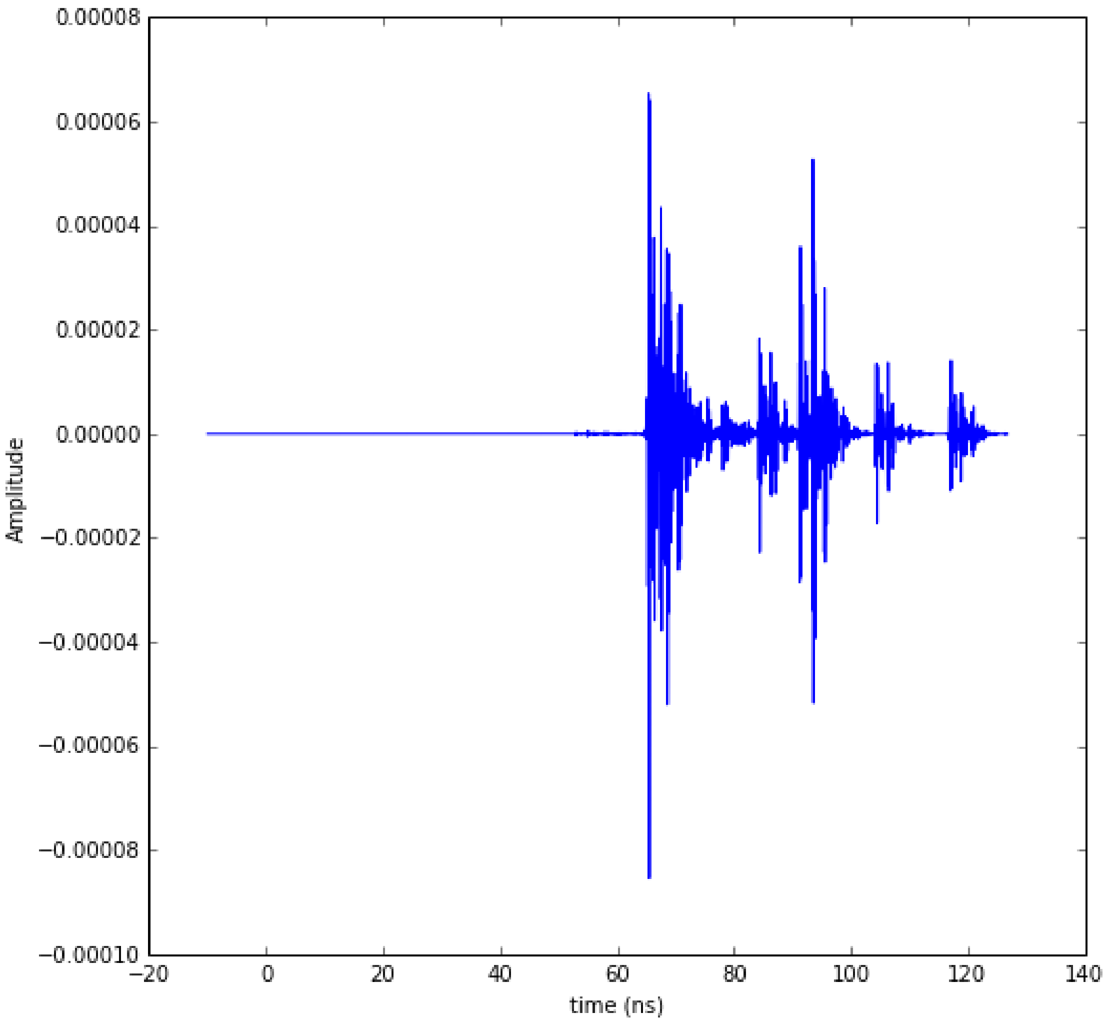
```
In [46]: M.tx

Out[46]: array([-22.3797,  13.3897,   1.2   ])

In [47]: M.rx

Out[47]: array([[  0.    ,   0.    ,   1.2   ],
                [-12.2724,   7.7632,   1.2   ],
                [-18.7747,  15.178 ,   1.2   ],
                [ -4.1418,   8.8603,   1.2   ],
                [ -9.0914,  15.1899,   1.2   ]])

In [48]: np.sqrt(np.sum((M.tx-M.rx[3,:])*(M.tx-M.rx[3,:]),axis=0))/0.3

Out[48]: 62.639751380717335

In [49]: Lk.H.ak
```

```
Out[49]: array([  9.79138753e-05,   2.96104973e-04,   1.59308395e-04,
                  2.27580461e-05,   1.34574073e-04,   4.82827749e-04,
                  1.18095058e-03,   1.27977423e-04,   1.34909317e-04,
                  5.48270185e-05,   9.81119862e-05,   3.61269565e-04,
                  2.73004108e-04,   1.24319829e-04,   2.28615744e-04,
                  2.76183582e-04,   2.32071461e-04,   2.86885128e-04,
                  5.08065400e-04,   4.16943960e-05,   9.25585146e-05,
                  3.67562947e-05,   3.26755640e-05,   7.18843013e-05,
                  5.98546831e-05,   5.35179821e-05,   8.90165410e-05,
                  3.64178931e-04,   3.64157003e-04,   1.49754933e-04,
                  9.73558942e-05,   4.65239631e-05,   5.86669038e-05,
                  2.05936803e-04,   9.48731849e-05,   2.36321979e-04,
                  6.88923268e-04,   1.24565228e-04,   4.69513933e-05,
                  2.31194688e-05,   2.94412022e-05,   2.94227854e-05,
                  5.60825032e-05,   7.29054840e-05,   1.70818338e-04,
                  1.70818449e-04,   5.34747295e-05,   4.74216436e-05,
                  1.14993846e-04,   5.08387934e-05])
```

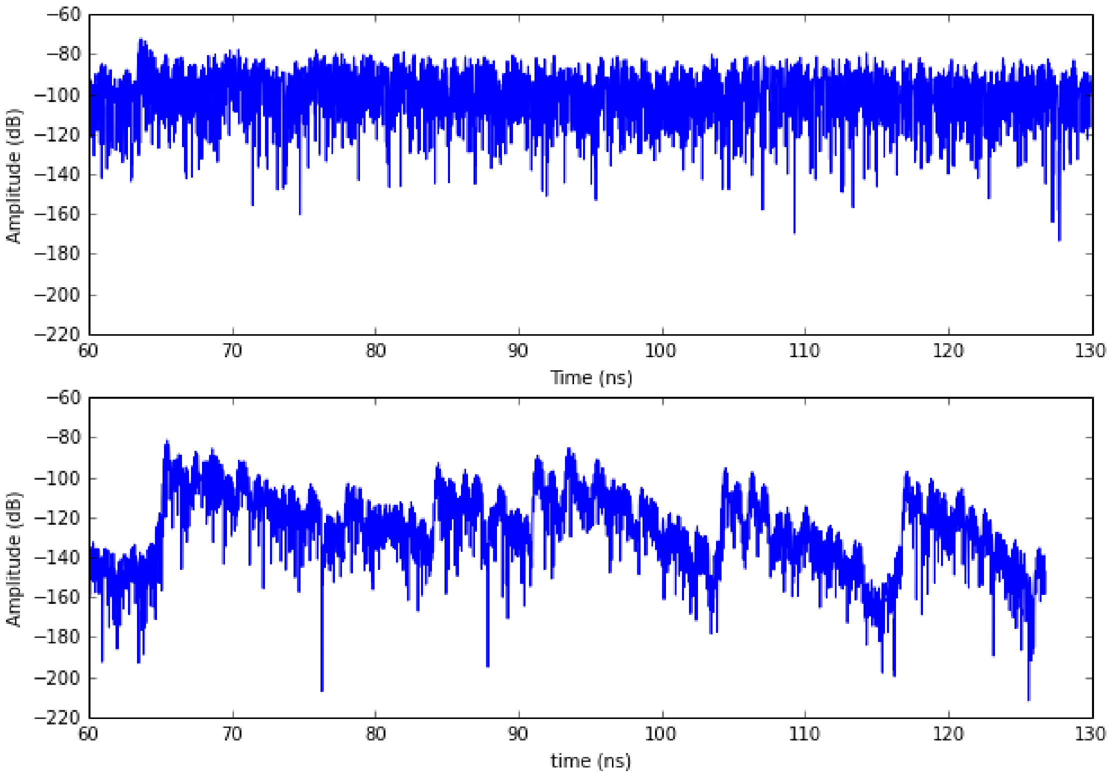
```
In [50]: Lk.wav=wav
```

```
In [51]: ir.plot(typ='v')
```

```
Out[51]: (<matplotlib.figure.Figure at 0x7f8ca0887fd0>,
          array([[<matplotlib.axes.AxesSubplot object at 0x7f8ca123aa50>]], dtype=object))
```



```
In [52]: ir

Out[52]: Usignal :  (9047,)  (9047,)
         ax0 : 9047

In [53]: fig = plt.figure(figsize=(10,7))
         ax1=fig.add_subplot(211)
         cmd='M.tdd.ch' + str(rx_id) + '.plot(typ=[\'l20\'],fig=fig,ax=ax1)'
         eval(cmd)
         ax2 = fig.add_subplot(212,sharex=ax1,sharey=ax1)
         #Lk.chanreal.plot(typ=['v'],fig=fig,ax=ax2)
         ir.plot(typ=['l20'],fig=fig,ax=ax2)
         plt.xlim(60,130)

Out[53]: (60, 130)
```


−60
−80
−100
−120
−140
−160
−180
−200
−220
Amplitude (dB)
60
70
80
90
100
110
120
130
Time (ns)
−60
−80
−100
−120
−140
−160
−180
−200
−220
Amplitude (dB)
60
70
80
90
100
110
120
130
time (ns)

```
In [54]:
```