

Model Reconstruction Tricks Using the Package

Author : Tomasz Konopka

Date : October 5, 2010

The objective of this notebook is to demonstrate some additional model reconstruction techniques using the package. The notebook is divided into several section, each dealing with a separate technique.

Prep

It is nice to clear variables before starting a new computation to avoid conflicts

```
In[1]:= ClearAll["Global`*"];
```

Load the package. Sometimes the path of the package must be added to the path in order for Mathematica to find it

```
In[2]:= $Path = Prepend[$Path, "/home/tomasz/MyP_Bruxelles/MoreFourier"];  
Needs["Autoosc`"]; 
```

To check if the package is loaded properly, execute the following command. If all is ok, it should display some text

```
In[4]:= AOAbout []
```

```
Package Packagename  
Version: 0.1  
Author: Tomasz Konopka
```

Set up some settings for the reconstruction calculations

```
In[5]:= GoDepth = 2;  
ToleranceArray = {1, 1, 1};  
RIToleranceArray = {0.5, 1, 2};  
parameterranges = {{pp1, -6000, 6000}, {pp2, 0, 300}, {qq1, -6000, 6000}, {qq2, 0, 300}};  
AOSetLevels[GoDepth, parameterranges, ToleranceArray, RIToleranceArray]  
AOGetLevels[]
```

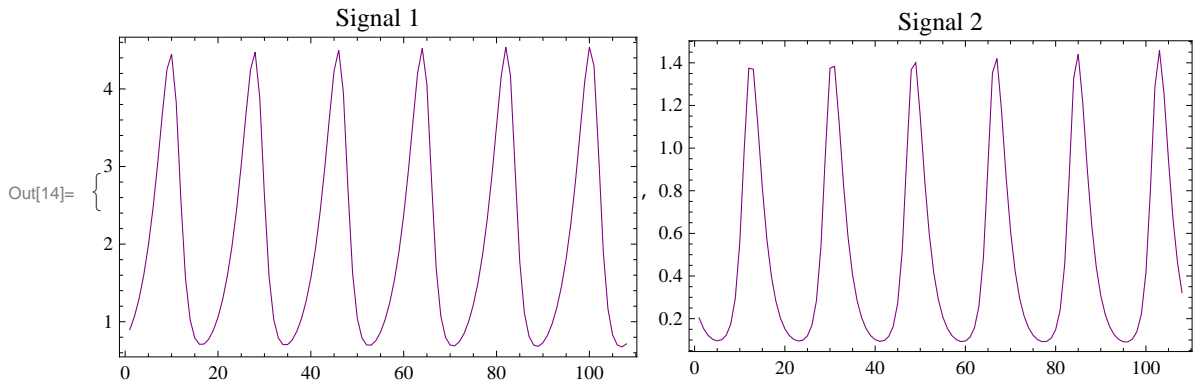
```
depth: 2  
Tolerances: {1, 1, 1}  
ReImTolerances: {0.5, 1, 2}  
Parameter ranges: {{-6000, 6000}, {0, 300}, {-6000, 6000}, {0, 300}}
```

Read data from disk, show it, and compute the spectra. The signals are the Lotka - Volterra oscillators from the previous example

```

In[11]:= LoadedSignals = {0, 0};
LoadedSignals[[1]] =
  ReadList["/home/tomasz/MyP_Bruxelles/MoreFourier/Examples/exempladata1.dat"];
LoadedSignals[[2]] = ReadList[
  "/home/tomasz/MyP_Bruxelles/MoreFourier/Examples/exempladata2.dat"];
{AOPlotModel[LoadedSignals[[1]], 250, "Signal 1"],
 AOPlotModel[LoadedSignals[[2]], 250, "Signal 2"]}
Spectra = {0, 0};
Signals = LoadedSignals;
Spectra[[1]] = AOComputeSpec[Signals[[1]]];
Spectra[[2]] = AOComputeSpec[Signals[[2]]];

```



```

In[19]:= numcycles = 6;

```

Model selection with time delays

This section shows a way to attempt model reconstruction with models with an explicit time delay, i.e. equations in which one of the fields is $\Psi(t-\text{delay})$ rather than $\Psi(t)$.

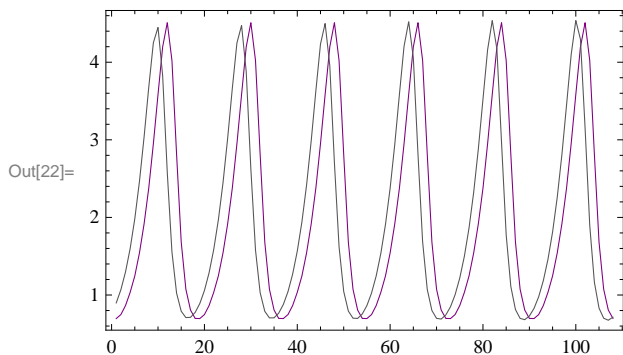
With this technique, it is the driving signal only that can be delayed.

Create a new signal that is a time - delayed version of signal 1. The delay is computed using the spectrum. Visually, it can be checked using the signal ansatz procedure. The plot shows two curves, one slightly delayed than the other

```

In[20]:= DelayedSpec = AODelaySpec[Spectra[[1]], 2];
DelayedSignalAnsatz = AOShowAnsatz[DelayedSpec, 6, 6];
AOPlotModel[{DelayedSignalAnsatz, Signals[[1]]}, 250, ""]

```



Model reconstruction can be applied to the original signals as well as the time delayed.

When using the original signal and the Lotka-Volterra equation structure, the model selection should return True

```
In[23]:= try1 = AComputeParameters[Spectra[[1]],
      Spectra[[2]], numcycles, 6, {{2, 1, 0}, {4, 1, 1}}, 2, "S", False]
      AOInterpretParameters[try1[[3]]]

Out[23]= {pp1 X[t] + qq1 X[t] XD[t] - X'[t], {{2, 1, 0}, {4, 1, 1}},
      {{1, {{pp1 → -0.551676 + 0.00160894 i, qq1 → 0.270391 - 0.000788583 i}}},
      {2, {{pp1 → -0.547606 + 0.00249228 i, qq1 → 0.268396 - 0.00122153 i}}}}}

Out[24]= {True, {{pp1 → -0.551676 + 0.00160894 i, qq1 → 0.270391 - 0.000788583 i}}}
```

When using the delayed spectrum, the result may be True or False, depending on how strong the delay is

```
In[25]:= try1 = AComputeParameters[DelayedSpec,
      Spectra[[2]], numcycles, 6, {{2, 1, 0}, {4, 1, 1}}, 2, "S", False]
      AOInterpretParameters[try1[[3]]]

Out[25]= {pp1 X[t] + qq1 X[t] XD[t] - X'[t], {{2, 1, 0}, {4, 1, 1}},
      {{1, {{pp1 → -0.591161 - 0.420582 i, qq1 → 0.205629 + 0.146295 i}}},
      {2, {{pp1 → -0.399101 - 0.528826 i, qq1 → 0.138823 + 0.183946 i}}}}}

Out[26]= {False, {}}
```

Since the delayed spectrum is now saved in a separate list from the spectrum of the original signal 1, it can be tested whether the delayed spectrum of signal 1 is a driving signal for signal 1

```
In[27]:= try1 = AComputeParameters[DelayedSpec,
      Spectra[[1]], numcycles, 6, {{2, 1, 0}, {4, 1, 1}}, 2, "S", False]
      AOInterpretParameters[try1[[3]]]

Out[27]= {pp1 X[t] + qq1 X[t] XD[t] - X'[t], {{2, 1, 0}, {4, 1, 1}},
      {{1, {{pp1 → 0.371869 - 0.286663 i, qq1 → -0.143349 + 0.110503 i}}},
      {2, {{pp1 → 0.396255 - 0.385471 i, qq1 → -0.152749 + 0.148592 i}}}}}

Out[28]= {False, {}}
```

```
In[29]:=
```

Comment : this technique allows to carry out model selection analysis using models with explicit time delays. However, the time delays are tested one at a time. I.e. to test whether a particular delayed signal is driving another, it is necessary to generate X signals with different time delays and test each in turn. The computational time for model selection is therefore multiplied by a factor of X.

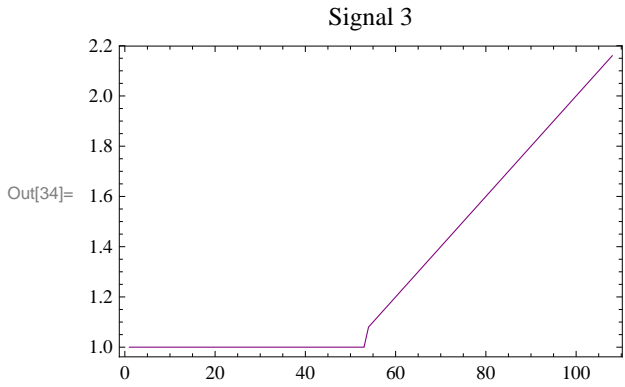
```
In[30]:=
```

Using multiple driving signals

This section illustrates possibilities for testing model structures which depend on multiple driving signals, or driving signals that enter in different forms in the equation than immediately supported by the package.

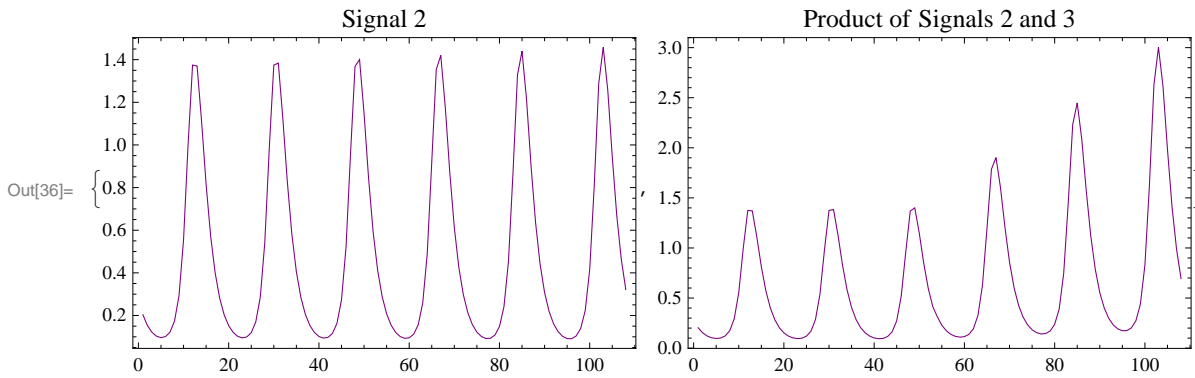
The example used to illustrate this is here the following. Suppose there is another signal 3 is low for some time and higher afterward

```
In[31]:= Signal3 = Table[0, {i, 1, Length[Signals[[2]]}];
Do[Signal3[[i]] = 1;, {i, 1, Floor[Length[Signal3] / 2]}];
Do[Signal3[[i]] = i / 50;, {i, Floor[Length[Signal3] / 2], Length[Signal3]}];
AOPlotModel[Signal3, 250, "Signal 3"]
```



Suppose that the driving signal is a product of Signal 2 and Signal 3.

```
In[35]:= DrivingSignal = Signal3 * Signals[[2]];
{AOPlotModel[Signals[[2]], 250, "Signal 2"],
 AOPlotModel[DrivingSignal, 250, "Product of Signals 2 and 3"]}
```

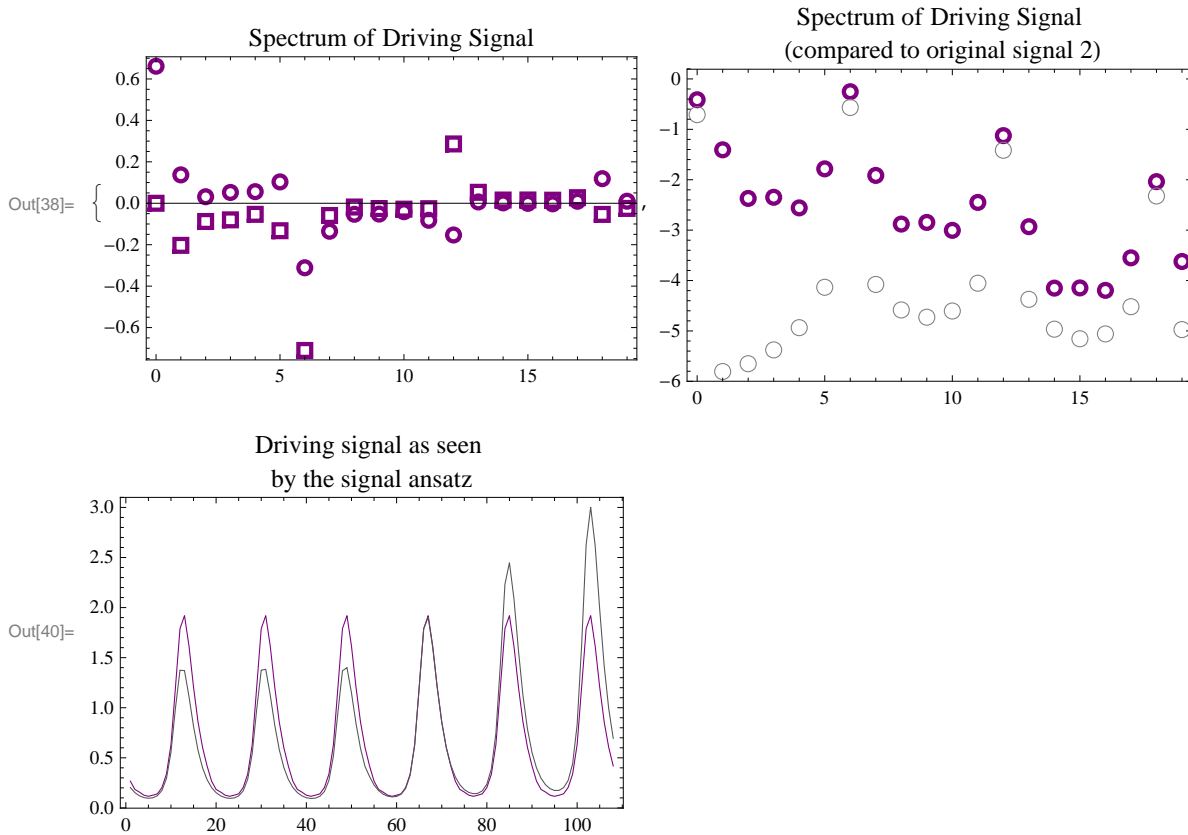


The spectrum of the new driving signal will be somewhat different from the original

```

In[37]:= DrivingSpectrum = AOComputeSpec[DrivingSignal];
{AOPlotModelSpec[DrivingSpectrum, 20, 250, "Spectrum of Driving Signal"],
 AOPlotModelSpecAbs[{DrivingSpectrum, Spectra[[2]]}, 20, 250, True,
 "Spectrum of Driving Signal\n (compared to original signal 2)"]}
DrivingSignalAnsatz = AOShowAnsatz[DrivingSpectrum, 6, 6];
AOPlotModel[{DrivingSignalAnsatz, DrivingSignal},
 250, "Driving signal as seen \nby the signal ansatz"]

```



The new driving signal, formed by a combination of the signal 2 and the step function, can be used for model selection. First, show results from testing the Lotka-Volterra equation with the original data. This should result in True

```

In[41]:= try1 = AOComputeParameters[Spectra[[2]],
  Spectra[[1]], numcycles, 6, {{2, 1, 0}, {4, 1, 1}}, 2, "S", False]
AOInterpretParameters[try1[[3]]]

Out[41]= {pp1 X[t] + qq1 X[t] XD[t] - X'[t], {{2, 1, 0}, {4, 1, 1}},
  {{1, {{pp1 -> 0.270405 + 0.000549996 i, qq1 -> -0.549787 - 0.00111825 i}}},
  {2, {{pp1 -> 0.270205 - 0.00183394 i, qq1 -> -0.549379 + 0.00372877 i}}}}}

Out[42]= {True, {{pp1 -> 0.270405 + 0.000549996 i, qq1 -> -0.549787 - 0.00111825 i}}}

```

When using the new driving signal, the result may still be True, because of the form of the modified driving function.

```

In[43]:= try1 = AOComputeParameters[DrivingSpectrum,
  Spectra[[1]], numcycles, 6, {{2, 1, 0}, {4, 1, 1}}, 2, "S", False]
AOInterpretParameters[try1[[3]]]

Out[43]= {pp1 X[t] + qq1 X[t] XD[t] - X'[t], {{2, 1, 0}, {4, 1, 1}},
  {{1, {{pp1 -> 0.26504 + 0.00628427 i, qq1 -> -0.40764 - 0.00966541 i}}},
  {2, {{pp1 -> 0.269711 + 0.00802746 i, qq1 -> -0.414823 - 0.0123465 i}}}}}

Out[44]= {True, {{pp1 -> 0.26504 + 0.00628427 i, qq1 -> -0.40764 - 0.00966541 i}}}

```

In[45]:=

Comment: Interpretation of such results should be done with care.

Comment: The technique described here can be applied to produce driving signals out of combinations of several raw measured signals.

In[46]:=

Conclusion

The package offers a number of core functions suited for the study of a particular model selection question. However, the functions can also be applied for other types of model selection investigations. The tricks described here are an indication of the possibilities.