Multivariate Smooth Transition and Threshold AutoRegressive Model

Application to artificial time series

Initial settings

■ packages read-in

```
<< Statistics`LinearRegression`
<< Statistics`HypothesisTests`
<< "timeseri\\timeseri.m"
<< "timeseri\\userfunc.m"
```

■ system settings

```
SetDirectory["d:\\doc\\phd\\_vypocty\\MLSTAR"];
```

```
SetOptions[ListPlot, PlotJoined → True, PlotRange → All, DisplayFunction → Identity];
SetOptions[Plot, PlotRange → All, DisplayFunction → Identity];
Off[General::spell1];
```

■ data

Bivariate data in *ARmodel1.dat* were generated from 2-regime TAR model with VAR(1) in both regimes $y_t = \Phi_1 X_t (1 - I[z_t - r]) + \Phi_2 X_t I[z_t - r] + \epsilon_t$ and the following setup: $z_t = y_{1,t-d}$ and r=0, d=1, $\epsilon_t \sim N(\mathbf{0}, \Sigma)$
Φ1={{0.7,0.0},{0.3,0.7}}; Σ1={{1.0,0.2},{0.2,1.0}};
Φ2={{-0.7,0.0},{-0.3,-0.7}}; Σ2={{1.0,-0.3},{-0.3,1.0}};

```
ReadList["ARmodel1.dat", Number, RecordLists → True];
{y1, y2} = Transpose[%];
n = Length[y1]
```

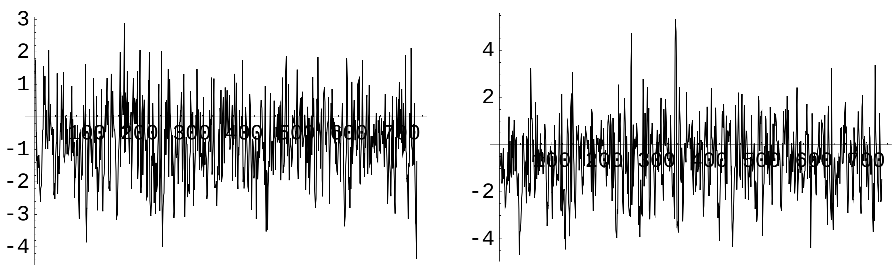
```
730
```

Preliminary diagnostics

■ First look

visual reconnaissance

```
GraphicsArray[ListPlot /@ {y1, y2}] // fShow;
```

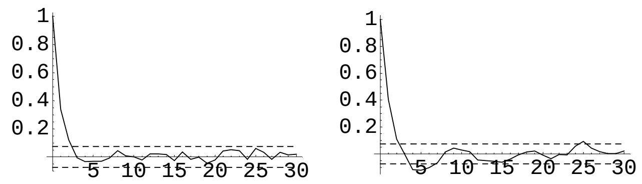


```
MatrixForm[{{"mean: ", Mean /@ #}, {"variance: ", Variance /@ #}}] &[{y1, y2}]
```

```
( mean:       {-0.786145, -0.630419} )
( variance:   {1.25155, 2.21144}     )
```

■ Correlogram

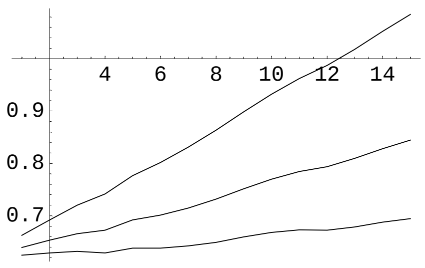
```
fCorPlot[{y1, y2}, 30];
```



■ Choosing the order of (V)AR from LevinsonDurbin procedure

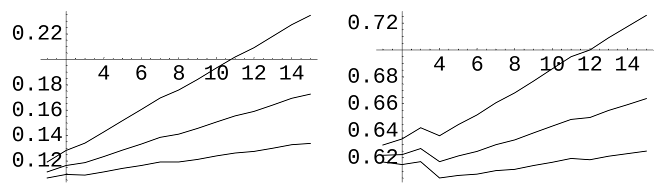
▫ VAR

```
fVABHQIC[{y1, y2}, 15];
```



▫ AR

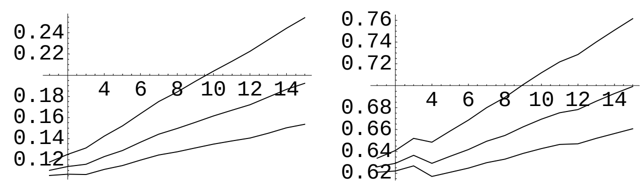
```
fABHQIC[{y1, y2}, 15];
```



■ Choosing the order of AR with no use of TimeSeries package

▫ AR

```
fABHQIC[{y1, y2}, {{y1}, {y2}}, 15];
```



■ Spectral analysis

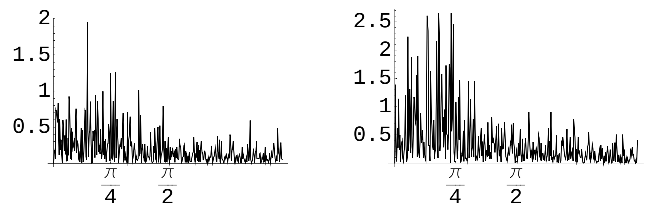
periods with the highest spectral density (not sorted)

```
fGreatestPeriods[{y1, y2}, 3]
```

```
{{13.5185, 8.02198, 7.37374}, {14.898, 11.0606, 8.58824}}
```

spectral density with tics at $2\pi/8$ and $2\pi/4$, where 8 is the period

```
fSpectrumPlot[{y1, y2}, {8}, PlotRange → {All, All}];
```



■ preprocessing and global notation

If data contain linear trend and/or seasonal components, it is recommended to make the data stationary by removing these deterministics. However, this is not the case here.
For the next procedures, we establish following notation:
dMo – data to be modelled, endogenous variables; dEx – exogenous variables; dTh - threshold variable; dDe - deterministic component series.

```
dMo = {y1, y2}; dEx = {}; dTh = y1; dDe = dMo - dMo;
```

Testing for linearity (against regime-switching nonlinearity)

Smaller *p-value* indicates higher probability of regime-switching nonlinearity.

■ Setting aggregation operator

set the function that will be used as aggregation operator

```
fAgOp = fAggregationOperator[#, 8] &;
```

■ Tsay's test

Multivariate test performed over $d \times p$ grid. For the smallest p-value the corresponding *d* and *p* are printed.

```
dlist = Range[1, 4]; plist = Range[1, 4];
Outer[(fChiSquarePValue @@ fTsayLTest[dMo, #2, dEx, 0, dTh, #1, fAgOp]) &, dlist, plist] // MatrixForm
fArgMinTensor[%];
Print["d=", dlist[[%[[1]]]], "   p=", plist[[%[[2]]]], "   p-value= ", %%[[Sequence @@ %]]]
```

0.	0.	0.	0.
0.855733	0.423061	0.551066	0.749172
0.793757	0.355385	0.551066	0.749172
0.770937	0.355503	0.562922	0.749172

```
Warning: 4 solutions found. Add "AllSolutions→True" to see them all.

d=1   p=1   p-value= 0.
```

```
fLTestReport[fTsayLTest[dMo, 1, dEx, 0, dTh, 1], 0.95]
```

t.statistic	dof	quantile	linear ?	p-value
156.647	6	12.5916	False	0.

■ LM_3 test

Multivariate test performed over $d \times p$ grid. For the smallest p-value the corresponding *d* and *p* are printed.

```
dlist = Range[1, 4]; plist = Range[1, 4];
Outer[(fChiSquarePValue @@ fLM3LTest[dMo, dEx, {#2, 0}, dTh, #1, fAgOp]) &, dlist, plist] // MatrixForm
fArgMinTensor[%];
Print["d=", dlist[[%[[1]]]], "   p=", plist[[%[[2]]]], "   p-value= ", %%[[Sequence @@ %]]]
```

0.	0.	0.	0.
0.0000101212	0.0000130528	0.0000810679	0.000858504
0.0334932	0.0583834	0.0471065	0.136665
0.123195	0.294066	0.0898897	0.08878

```
Warning: 4 solutions found. Add "AllSolutions→True" to see them all.

d=1   p=1   p-value= 0.
```

```
fLTestReport[fLM3LTest[dMo, dEx, {1, 0}, dTh, 1, fAgOp], 0.95]
```

t.statistic	dof	quantile	linear ?	p-value
376.189	18	28.8693	False	0.

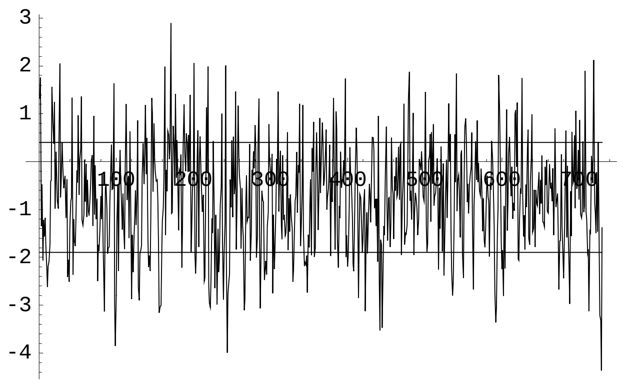
Model selection by in-sample fit

■ initials

Delimit threshold values to be used in grid search, preserving at least 15% of data for each regime. Check the coverage in the plot of threshold variable.

```
fThresholdRange[dTh, 0.15, 20, 1];
Range @@ (rrange = %) // Length;
MatrixForm[{{"threshold range (from,to,step): ", %%}, {"number of treshold values:", %}}]
Show[ListPlot[dTh], Plot[Evaluate[rrange⟦{1, 2}⟧], {i, 1, n}]] // fShow;
```

```
( threshold range (from,to,step):   {-1.9, 0.4, 0.1} )
( number of treshold values:        24               )
```



list of smooth parameter values

```
γlist = {0.5, 2, 10, 50};
```

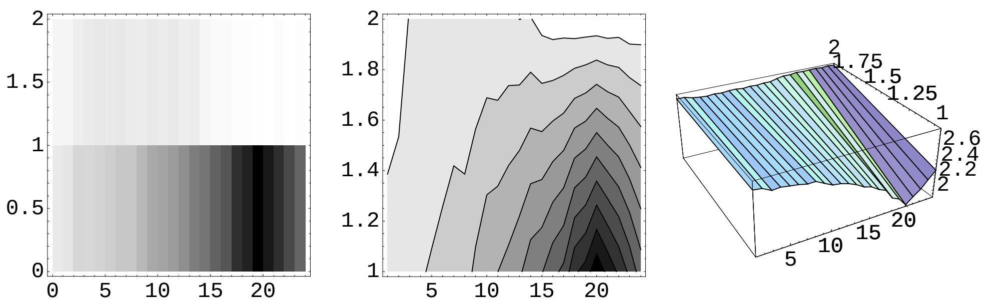
■ minimizing sum of squares for known orders of AR submodels

Sum of suared residuals computed for grid d×r
(modelled: {y1,y2}, exogenous: none, order *p*: 1, threshold variable: 1st variable of the modeled, No of regimes: 2, delay *d* in treshold var.: {1,2}, threshold values: {-1.9,-1.8,... 0.4},
other options: 110 = (1) output is sum of squared residuals, (1) /not used now/ transition function is set to logistic, (0) no aggregation operator is applied to threshold variable)

```
tmp = fConditionalRegimeSwitching[dMo, dEx, {1, 0}, dTh, 2, {{1, 2}, rrange}, {110}];
```

looking for smallest residuals, first visually - (2×24) matrix plot, then using argmin function - output is the parameters that minimize the sum of squared residuals

```
fMatrixPlots[tmp⟦1⟧];
fArgMinTensor[tmp⟦1⟧, tmp⟦2⟧] // Chop
```



```
{1, 0}
```

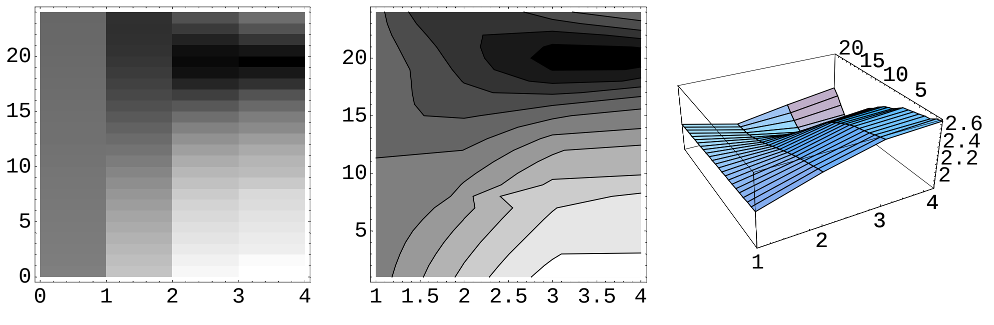
now LSTAR is used instead of TAR computed above (*d*: 1, transition smoothness parameter γ: {0.5, 2, 10, 50}, other options are left to default values)

```
Timing[tmp = fConditionalRegimeSwitching[dMo, dEx, {1, 0}, dTh, 2, {1, rrange, γlist}];]
```

```
{12.875 Second, Null}
```

sublist specification points to matrix: threshold (2nd) against smooth (3rd) parameter

```
fMatrixPlots[tmp⟦1⟧⟦1, All, All⟧];
fArgMinTensor[tmp⟦1⟧, tmp⟦2⟧] // Chop
```



```
{1, 0, 50}
```

■ minimizing AIC and BIC information criteria

Akaike and Schwarz (bayesian) information criteria computed for various orders of AR models in regimes
(options:
310 = (3) output is {AIC, BIC}, (1) transition function is set to logistic, (0) no aggregation operator is applied to threshold variable
Last = default type of aggregation operator, when only dTh_{t-d} is considered as treshold variable z_t
{1} = generate several sets of lags in AR model, such that maximum order *p*=4 is being decreased jointly for all regimes)

```
tmp = fConditionalRegimeSwitching[dMo, dEx, {4, 0}, dTh, 2, {1, 0, 50}, {310, Last, {1}}];
```

parameters that minimized either AIC (1st row) or BIC (now order of AR is also a parameter under estimation)
note : the order is in full form, which expresses all lags for every variable in regression per each regime

```
(tmp1 = fArgMinTensor[tmp⟦1⟧, tmp⟦2⟧, #] & /@ {1, 2}) // MatrixForm
```

```
( 1  0  50  {{{{1}, {1}}, {}}, {{{1}, {1}}, {}}} )
( 1  0  50  {{{{1}, {1}}, {}}, {{{1}, {1}}, {}}} )
```

■ other estimates

parameters that minimized BIC are used to compute $(K_1+K_2)\times k$ parameter matrix Φ, $k\times k$ covariance matrix of residuals, $k\times(n\text{-}h)$ residuals

```
tmp1⟦2⟧;
{tΦ, tΣ, tres} = fConditionalRegimeSwitching[dMo, dEx, tord = %⟦-1⟧, dTh, 2, tpar = Drop[%, -1], {211, fAgOp}]⟦1⟧⟦
   Sequence @@ Table[1, {Length[tpar]}]⟧;
```

comparison of estimated parameter matrix with the real one, which - obviously - is unavailable for real data

```
tΦ // Transpose // MatrixForm
fAugment[{{0}, {0}}, {{0.7, 0.0}, {0.3, 0.7}}, {{0}, {0}}, {{-0.7, 0.0}, {-0.3, -0.7}}] // MatrixForm
```

```
(  0.0296685   0.736954  -0.0441415  -0.264911  -0.431952   0.0449293 )
( -0.0441137   0.298378   0.641691    0.167066  -0.405031  -0.713922  )
```

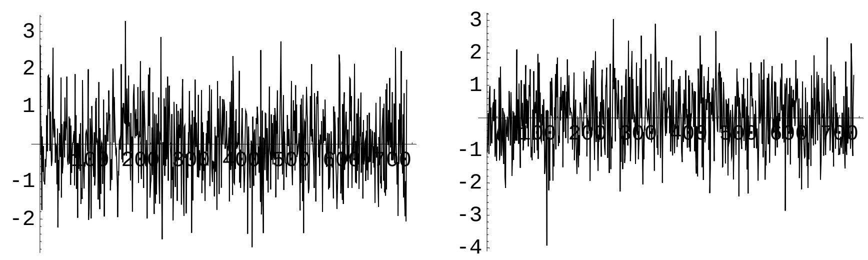
```
( 0  0.7  0.   0  -0.7   0.  )
( 0  0.3  0.7  0  -0.3  -0.7 )
```

■ Diagnostic on residuals

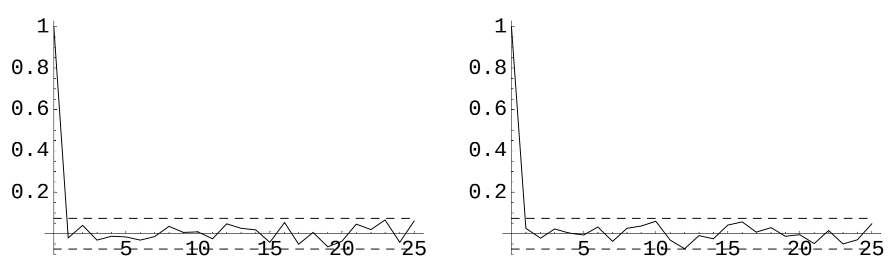
```
fResidualsTest[dMo, dEx, tord, dTh, tpar, tΦ, tres, 1, fAgOp]
```

```
test for:                  p-value:
serial independence        0.0133607
linearity                  0.0983665
```

```
fListPlot[tres];
```



```
fCorPlot[tres, 25];
```



Model selection by out-of-sample fit

■ forecast errors

Forecast errors evaluated at the last 30 time points
(options:
410 = (4) output is forecast errors, (1) /not used/ transition function is set to logistic, (0) no aggregation operator is applied to threshold variable
Last = default type of aggregation operator when only dTh_{t-d} is considered as treshold variable z_t
{30,1,20,{{},{}}} = (30) length of data tail used for comparison, (1)-step-ahead forecast, (20) No of Monte Carlo cycles, ({},{}) out-of sample exogenous & threshold variable data)
Note: Procedure instable if $\gamma\to 0$, already for γ=0.1

```
Timing[tmp = fConditionalRegimeSwitching[dMo, dEx, {1, 0}, dTh, 2, {{1}, rrange}, {410, Last, {30, 1, 20, {{}, {}} }}];]
```

```
{51.562 Second, Null}
```

process the output of fConditionalRegimeSwitching function to be used later for evaluating forecast performance

```
tmp1 = fUnNestAndCollectOutput[tmp];
```

■ Comparing MSPEs

hitparade by simply comparing MSPE (model entering position in the previous list, corresponding MSPE)
parameters (*d*, *r*) of the most accurate model

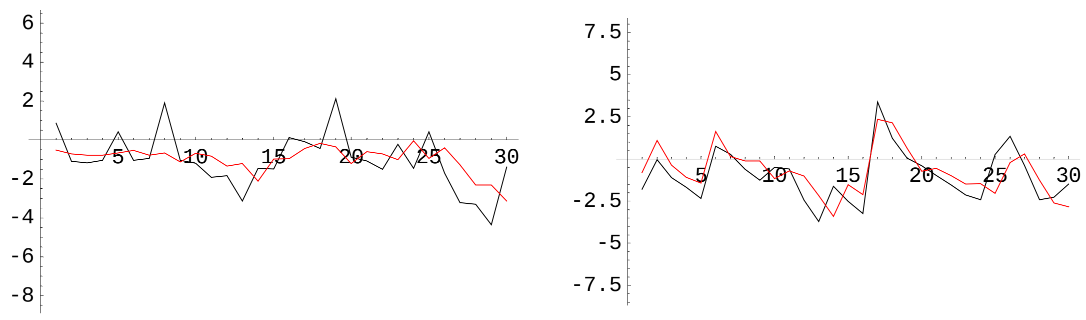
```
fMeanXPredictionErrorHitparade[tmp1[[All, 1]], Dot[#, #] &]
tmp1[[ All, 2 ]][[ %[[1]] ]][[ 1 ]] // Chop
```

```
{{20, 21, 19, 18, 15, 14, 16, 13, 17, 12, 10, 11, 9, 6, 3, 4, 7, 5, 8, 2, 24, 23, 22, 1},
 {2.51479, 2.53694, 2.68401, 2.7494, 2.98473, 3.00212, 3.02595, 3.04676, 3.04762, 3.06327, 3.06394, 3.0914,
  3.175, 3.22193, 3.2232, 3.23055, 3.23262, 3.23794, 3.2414, 3.31844, 3.33795, 3.38015, 3.41454, 3.45174}}
```

```
{1, 0}
```

forecasts chosen as best (according to some loss function, SPE by default) ploted together with original series and with optional deterministic component added.

```
fPlotTheBestForecast[dMo, tmp, dDe, 1.7];
%[[1]] // Chop
```



```
( d,r,γ:    {1, 0}  )
( MSPE:     2.51479 )
```

■ Comparing SPEs via modified Diebold-Mariano test

entering positions from the winner to the worst one
parameters (*d*, *r*) of the most accurate model

```
fDieboldMarianoHitparade[tmp1[[All, 1]], 1, 0.15, Dot[#, #] &]
tmp1[[ All, 2 ]][[ %[[1]] ]] // Chop
```

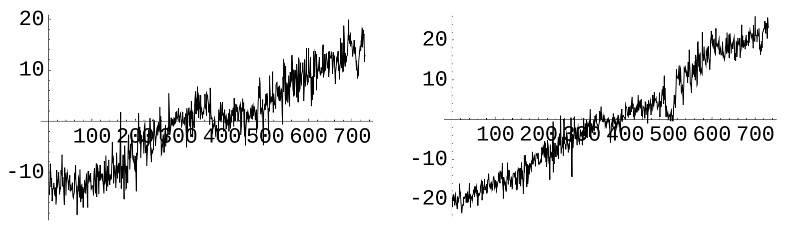
```
{20, 21, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 19, 22, 23, 24}, 17, 16}
```

```
{1, 0}
```

Application to bivariate GPS observations

■ data

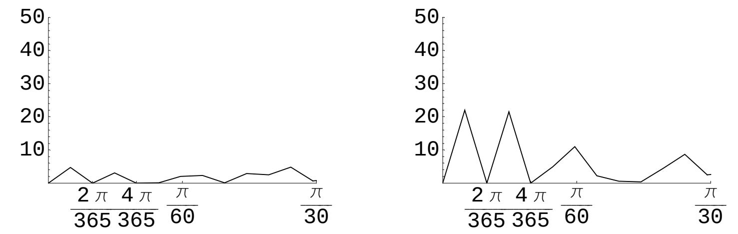
```
In[225]:=
        ReadList["bor1c.dat", Number, RecordLists → True];
        data = Transpose[%];
        {o1, o2, o3} = data;
        n = Length[o1]

Out[228]=
        730

In[229]:=
        fListPlot[{o1, o2}];
```



■ preprocessing

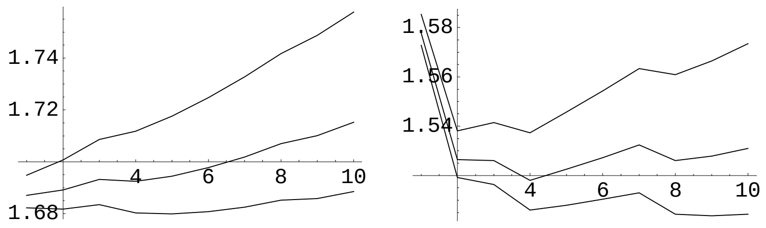
In[502]:=

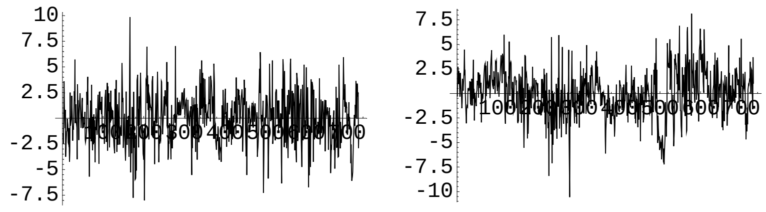
```
fDeterministicsRemoval[#, {0, 1}, {365, 365 / 2}] & /@ {o1, o2};
fSpectrumPlot[%, {365, 120}, PlotRange → {{0, 2 π / 60}, {0, 50}}];
fGreatestPeriods[%%, 5]
fABHQIC[%%%, 10];
fListPlot[%%%%];
```



Out[504]=

```
{{730., 66.3636, 36.5, 13.5185, 8.58824}, {730., 243.333, 121.667, 66.3636, 29.2}}
```



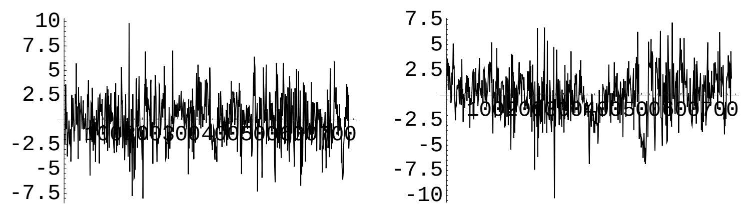


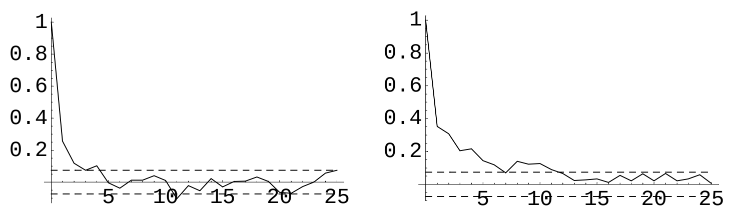
In[354]:=

```
y1 = fDeterministicsRemoval[o1, {0, 1}, {365, 365 / 2}];
y2 = fDeterministicsRemoval[o2, {0, 1}, {730 / 3, 365, 365 / 2}];
```

In[436]:=

```
GraphicsArray[ListPlot /@ {y1, y2}] // fShow;
fCorPlot[{y1, y2}, 25];
```





■ initials

In[469]:=

```
dMo = {y1, y2}; dEx = {}; dTh = y2; dDe = {o1, o2} - dMo;
```

In[449]:=

```
fAgOp = fAggregationOperator[#, 0] &;
γlist = {1, 2, 10, 100};
optF = {36, 1, 20, {{}, {}}};
```

In[447]:=

```
fThresholdRange[dTh, 0.15, 20, 1]
Range @@ (rRange = %) // Length
```

Out[447]=

```
{-2.3, 2.4, 0.2}
```

Out[448]=

```
24
```

■ Testing for Linearity

■ Tsay's test

In[470]:=
```
dlist = Range[1, 8]; plist = Range[1, 4];
Outer[(fChiSquarePValue @@ fTsayLTest[dMo, #2, dEx, 0, dTh, #1, fAgOp]) &, dlist, plist] // MatrixForm
fArgMinTensor[%];
Print["d=", dlist[[%[[1]]]], "   p=", plist[[%[[2]]]], "   p-value= ", %%[[Sequence @@ %]]]
```

Out[471]//MatrixForm=

0.0000203509	0.000884954	0.00115525	0.00121667
7.15318×10^{-7}	0.682728	0.907153	0.973786
0.000158246	0.0255366	0.336675	0.605895
9.77721×10^{-7}	0.000427164	0.000978177	0.00533385
0.00572677	0.0232981	0.0421952	0.0788315
0.0000213808	0.000215344	0.000191788	0.00140842
0.0762632	0.194419	0.0912406	0.14227
0.00139186	0.0721625	0.0701173	0.186451

d=2   p=1   p-value= 7.15318×10^{-7}

■ LM_3 test

In[474]:=
```
dlist = Range[1, 8]; plist = Range[1, 4];
Outer[(fChiSquarePValue @@ fLMtypeLTest[dMo, dEx, {#2, 0}, dTh, #1, fAgOp, 1]) &, dlist, plist] // MatrixForm
fArgMinTensor[%];
Print["d=", dlist[[%[[1]]]], "   p=", plist[[%[[2]]]], "   p-value= ", %%[[Sequence @@ %]]]
```

Out[475]//MatrixForm=

0.0000640628	0.000132296	0.000710505	0.0000518432
0.0394986	0.717146	0.917275	0.833814
0.0266259	0.273775	0.31339	0.41751
0.00457604	0.101799	0.363021	0.329247
0.167573	0.516066	0.324498	0.453824
0.00299904	0.0602739	0.0242313	0.128385
0.513615	0.524667	0.366719	0.411908
0.918698	0.971921	0.505403	0.347705

d=1   p=4   p-value= 0.0000518432

■ In-sample fit procedure

In[478]:=
```
dlist = {1, 2}; p = 2; q = 0;
```

In[479]:=
```
Timing[tmp = fConditionalRegimeSwitching[dMo, dEx, {p, q}, dTh, 2, {dlist, rRange, γlist}, {311, fAgOp, {111}}];]
```

Out[479]=
{121.141 Second, Null}

In[480]:=
```
(isf = fArgMinTensor[tmp[[1]], tmp[[2]], #] & /@ {1, 2}) // MatrixForm  // Chop
```

Out[480]//MatrixForm=

1	−2.3	100	{{{{1, 2}, {1, 2}}, {}}, {{{1, 2}, {1, 2}}, {}}}
1	−2.3	100	{{{{1, 2}, {1, 2}}, {}}, {{{1, 2}, {1, 2}}, {}}}

In[481]:=
```
isf[[2]];
{tΦ, tΣ, tres} = fConditionalRegimeSwitching[dMo, dEx, tord = %[[-1]], dTh, 2, tpar = Drop[%, -1], {211, fAgOp}][[1]][[
   Sequence @@ Table[1, {Length[tpar]}]]];
```

■ Diagnostic on residuals

Ljung-Box multivariate portmanteau test for serial independence

In[485]:=
```
fChiSquarePValue @@ fPortmanteauTest[tres, 2]
```

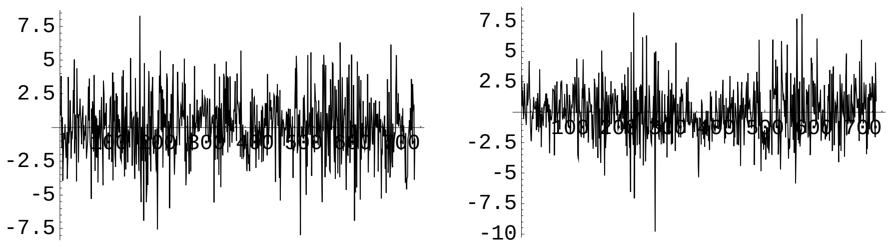
Out[485]=
0.982284

Durbin–Watson test statistic (if about 2, the 1-order serial correlation is zero). Univariate only.

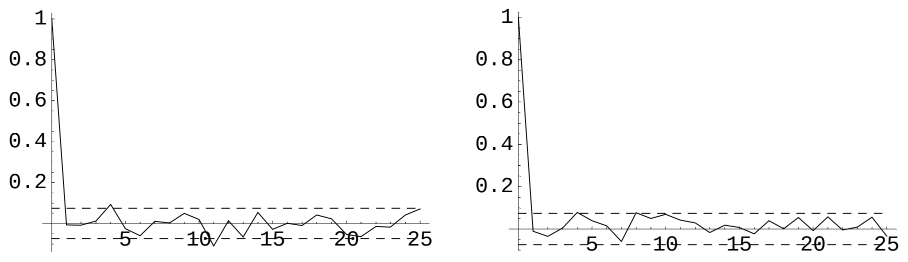
In[486]:=
```
fDurbinWatsonStatistic[#] & /@ tres
```

Out[486]=
{2.01251, 2.02502}

In[487]:=
```
fListPlot[tres];
```



In[488]:=
```
fCorPlot[tres, 25];
```

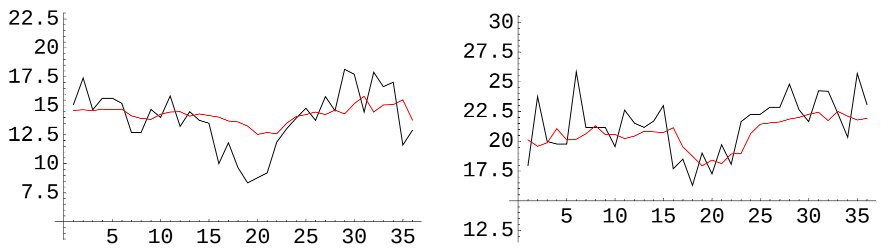


■ Forecasts

In[491]:=
```
Timing[tmp1 = fConditionalRegimeSwitching[dMo, dEx, tord, dTh, 2, {dlist, rRange, γlist}, {411, fAgOp, optF}];]
```

Out[491]=
```
{1738.92 Second, Null}
```

In[493]:=
```
fPlotTheBestForecast[dMo, tmp1, dDe, 1.5];
%[[1]]
```



Out[494]//MatrixForm=
```
( d,r,γ:    {1, 0.7, 100} )
( MSPE:     8.68415       )
```