Takagi-Sugeno Model Identification Toolbox

March 29, 2021

Static LiP model for the 3-dimensional Friedman test function.

V1.0

Axel Dürrbaum (axel.duerrbaum@mrt.uni-kassel.de)

Department of Measurement and Control (MRT)

Institute for System Analytics and Control (ISAC)

University of Kassel, Germany (http://www.uni-kassel.de/go/mrt)

$Id: Static_Friedman3D.m | Fri Feb 26 16:25:05 2021 +0100 | Axel Dürrbaum $

Contents

1 Identification data

Use the 3-dimensional Friedman function:

```
nu = 3;
```

$$y = 10 \cdot \sin(\pi \cdot u_1 \cdot u_2) + 20 \cdot (u_3 - 0.5)^2$$

Choose the fuzziness parameter $\nu = 1.2$

```
nue = 1.2;
```

Choose the input matrix u as random data with N data-points: $u_{1,2} \in [0, 1]$

```
N = 500;
u = rand( N, nu );
```

Compute the output vector y:

```
y = Friedman_fct( u, nu );
```

2 Structural parameters

Number of clusters n_v = number of local models

```
nv = 5;
```

Membership function Type: FCM

```
MSF = 'FCM';
```

3 Estimation of the static LiP TS model

```
addpath( '../TSModel' );  % Path to TSModel class
ts = TSModel( 'Static', nv, nu, 'comment', 'Friedman 3D' );
```

Set the identification data: u, y

```
ts.setData( u, y );
```

Clustering:

- FCM: fuzziness parameter $\nu = 1.2$ with Euclidean norm (default)
- clustering in product-space
- Multi-Start: 5 tries

```
ts.clustering( MSF, 'nue', nue, 'productspace', true, 'tries', 5 );
```

Initialisation of local models: global least squares estimation

```
ts.initialize( MSF, 'nue', nue, 'method', 'global' );
```

Optimization of both: membership and local model parameters

```
ts.optimize( 'Both' );
```

```
                                        Norm of      First-order
 Iteration  Func-count     f(x)          step          optimality
     0         36         829.803                     1.22e+03
     1         72         654.531            10            478
     2        108         654.531       22.9875            478
     3        144         512.447             5            254
     4        180          416.58       11.8577            197
     5        216          259.76       7.44467            106
     6        252         224.981       4.14896           49.1
     7        288           210.9       1.84324           28.3
     8        324         196.735       1.47128           46.3
     9        360         180.724       1.74153           51.2
    10        396         171.051       1.97235           14.4
    11        432         166.681       1.07093           16.3
    12        468         164.948      0.645502           13.5
    13        504         164.016      0.598966           9.84
    14        540         163.542      0.539842           7.73
    15        576         163.341      0.473414           5.28
    16        612         163.268      0.332755           3.63

Local minimum possible.

lsqnonlin stopped because the final change in the sum of squares relative to
its initial value is less than the value of the function tolerance.
```

Show the resultiung TS model parameters:

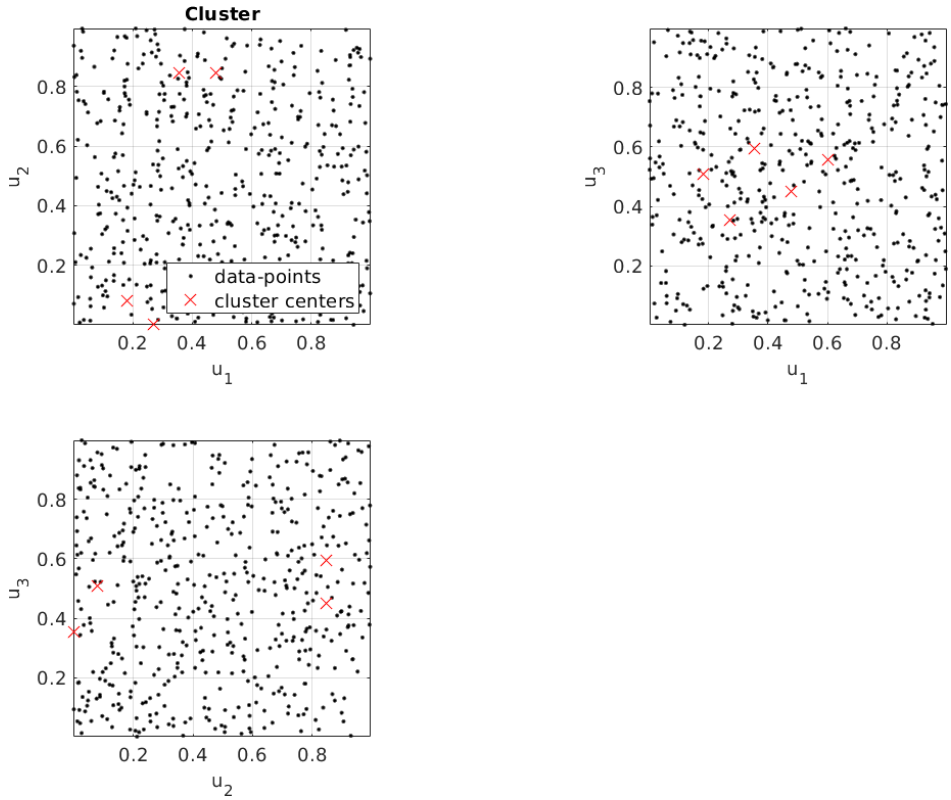
```
disp( ts )
```

```
TS-Model: Type=Static
Name: 'undefined'
 Type: 'TSModel'
 Date: '29-Mar-2021 15:49:58'
 Comments:
  'Friedman 3D'
 Structural parameters: nu = 3, ny = 1, nv = 5
 Identification data: N=500
 Initial model estimation:
  Clustering: FCM, nue=1.2 norm=Euclidean in product space
 Estimation of local models:
  Initialization of local models: global
  Optimization of model parameters: MF&LM
```

Plot the cluster centers: v

```
v = getCluster( ts )
ts.plotCluster( v, 'figure',1);
```

```
v =
    0.3541    0.8471    0.5957
    0.5996    0.9983    0.5581
    0.1811    0.0795    0.5085
    0.4779    0.8468    0.4498
    0.2704    0.0013    0.3554
```

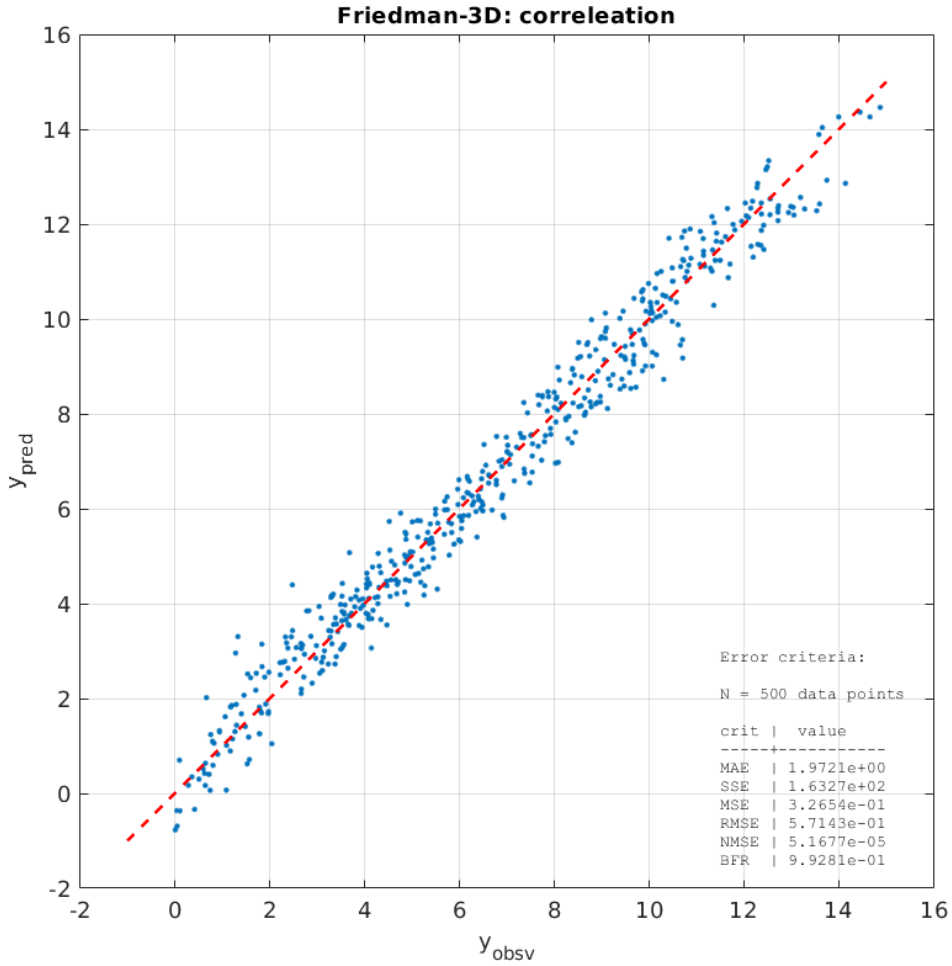


Predict the TS model output: y_{pred}

```
y_pred = ts.predict( u, y );
```

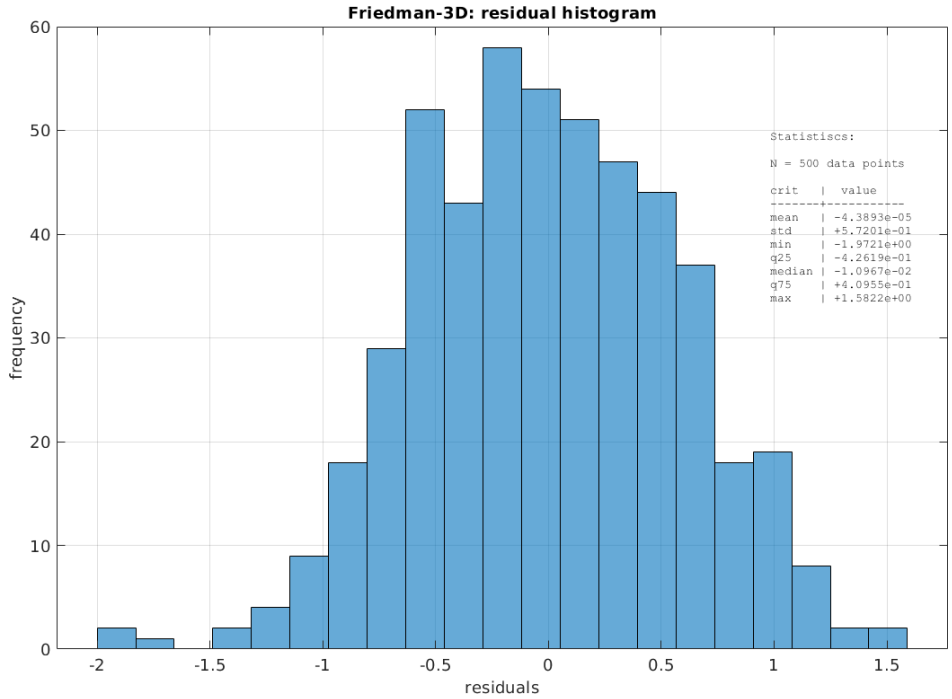
Plot the correlation

```
plotResiduals( y, y_pred, 'figure', 2, 'title', 'Friedman-3D: correleation' );
set(gcf,'WindowState','maximized');
```



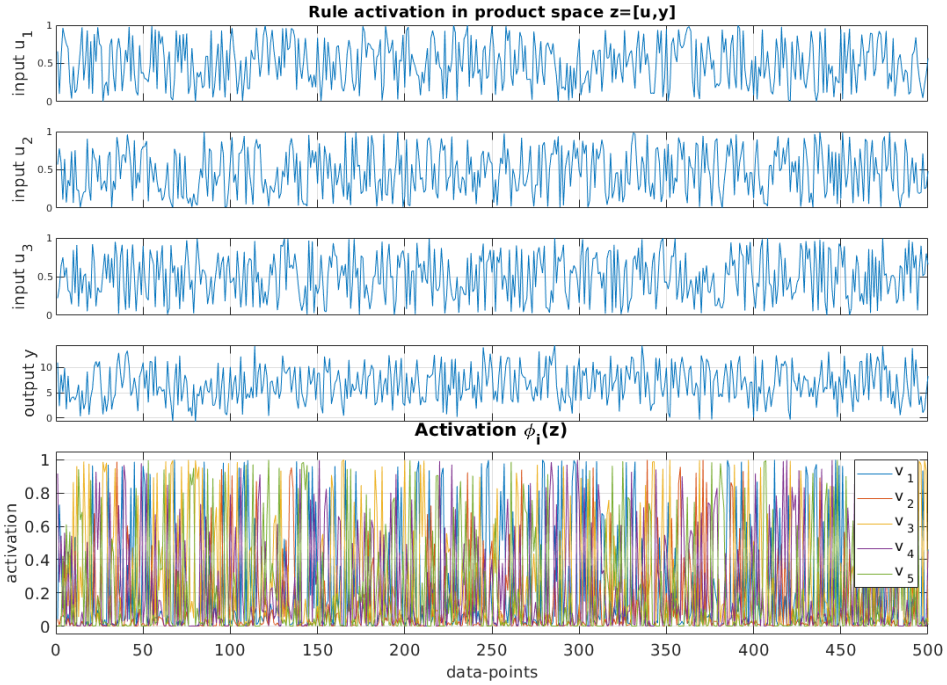
Plot the residual histogram:

```
plotResidualHist( y, y_pred, 'figure', 3, 'nbins', 21, ...
    'title', 'Friedman-3D: residual histogram' );
set(gcf,'WindowState','maximized');
```



Plot the rule activation and input/output data:

```
plotRuleActivation( u,y_pred, ts, 'figure', 4 );
set(gcf,'WindowState','maximized');
```



4 Validation of the TS model

Choose another N random inputs $[u_1, u_2, u_3]$

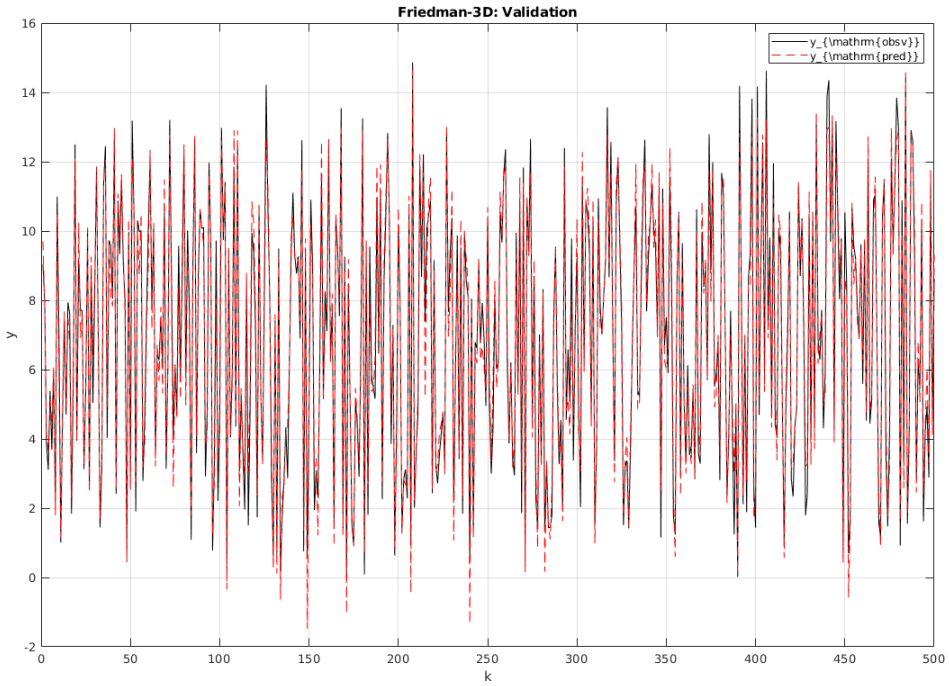
```
u_val = rand( N, nu );
y_obsv = Friedman_fct( u_val, nu );
```

Compute the output vector: y_{pred}

```
y_val_pred = ts.predict( u_val );
```

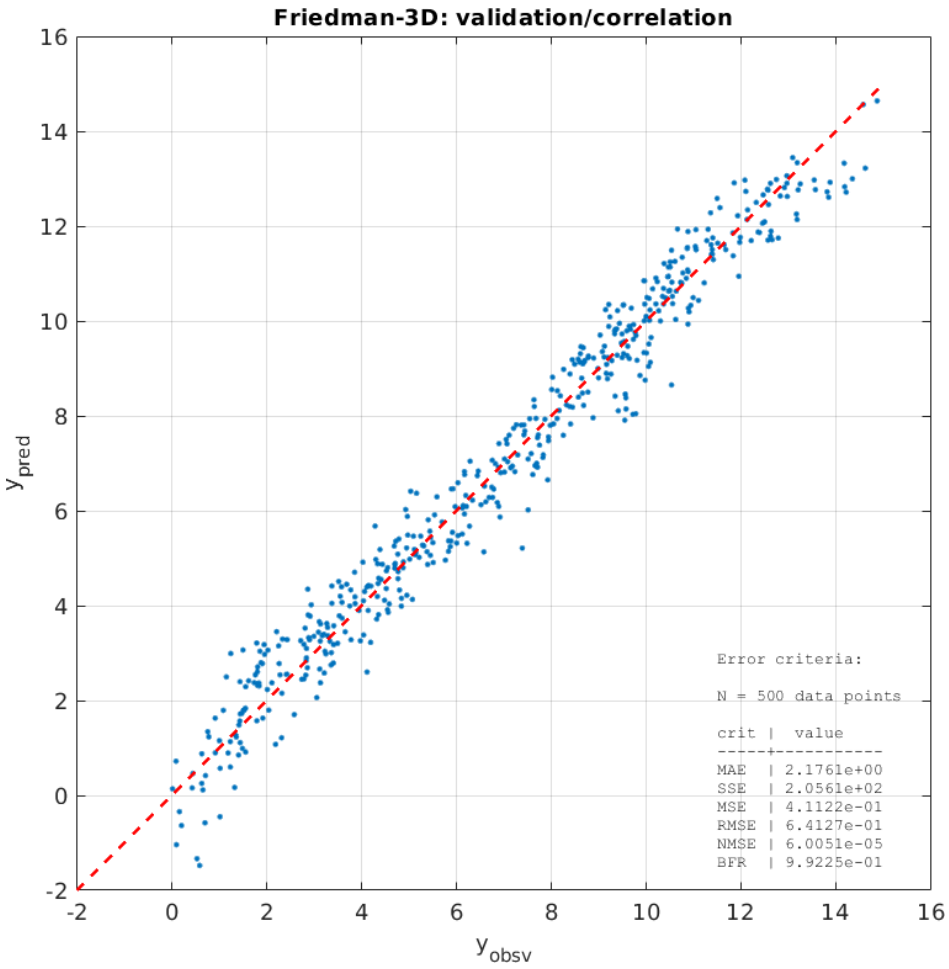
Plot the outputs

```
figure(5);clf
plot( 1:N, y_obsv, 'k-',1:N, y_val_pred, 'r--' )
grid on
xlabel('k')
ylabel('y')
title( 'Friedman-3D: Validation' )
legend( 'y_{\mathrm{obsv}}','y_{\mathrm{pred}}' )
set(gcf,'WindowState','maximized');
```



Plot the correlation

```
plotResiduals( y_obsv, y_val_pred, 'figure', 6, ...
    'title', 'Friedman-3D: validation/correlation' );
set(gcf,'WindowState','maximized');
```



Plot the correlation histogram

```
plotResidualHist( y_obsv, y_val_pred, 'figure', 7, 'nbins', 31, ...
```

```
    'title', 'Friedman-3D: validation/correlation histogram' );
set(gcf,'WindowState','maximized');
```