Takagi-Sugeno Model Identification Toolbox

May 21, 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         938.137                     1.62e+03
     1         72         782.094            10            803
     2        108         623.957            20            616
     3        144         623.957       23.7711            616
     4        180         514.589       5.94278            267
     5        216         401.823       11.8856            227
     6        252         303.388       10.4834            389
     7        288         256.621        15.789            212
     8        324         221.423       4.20916            105
     9        360          201.17       6.32526           25.6
    10        396         185.071       3.37852             32
    11        432         177.916       1.64542           30.3
    12        468         172.451       1.31403             21
    13        504         169.526        1.1683           12.5
    14        540         167.852       1.07311           12.2
    15        576         166.587      0.699664           10.6
    16        612         166.119      0.230535           9.73
    17        648         165.619      0.248118           8.95
    18        684         165.067      0.283859           8.09
    19        720         164.502       0.31759           7.41
    20        756         163.983      0.342999           6.46
    21        792         163.564      0.350972            5.3
```

```
    22        828         163.267       0.334196          4.22
    23        864         163.079       0.294595          3.34
    24        900         162.969       0.243338          2.65

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:

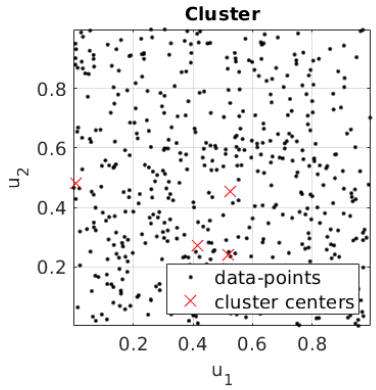
```
disp( ts )
```

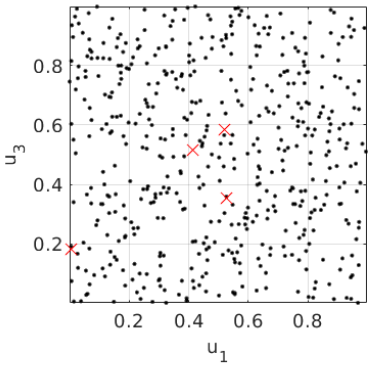
```
TS-Model: Type=Static
Name: 'undefined'
 Type: 'TSModel'
 Date: '21-May-2021 11:37:39'
 Comments:
  'Friedman 3D'
 Structural parameters: nu = 3, ny = 1, nv = 5
 Identification data: N=500
 Initial model estimation:
  Clustering: FCM, nue=1.2 norm=Euclidean in product space
 Estimation of local models:
  Initialization of local models: global
  Optimization of model parameters: MF&LM
```

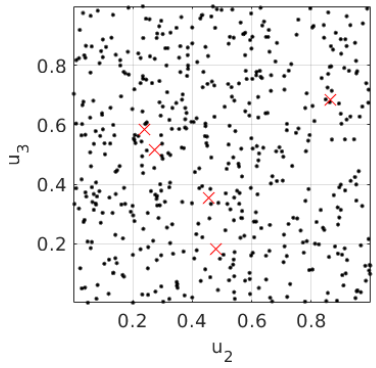
Plot the cluster centers: v

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

```
v =
    0.5188    0.2393    0.5837
    0.0076    0.4809    0.1834
    0.5253    0.4544    0.3524
    0.4137    0.2714    0.5162
    0.9984    0.8658    0.6838
```





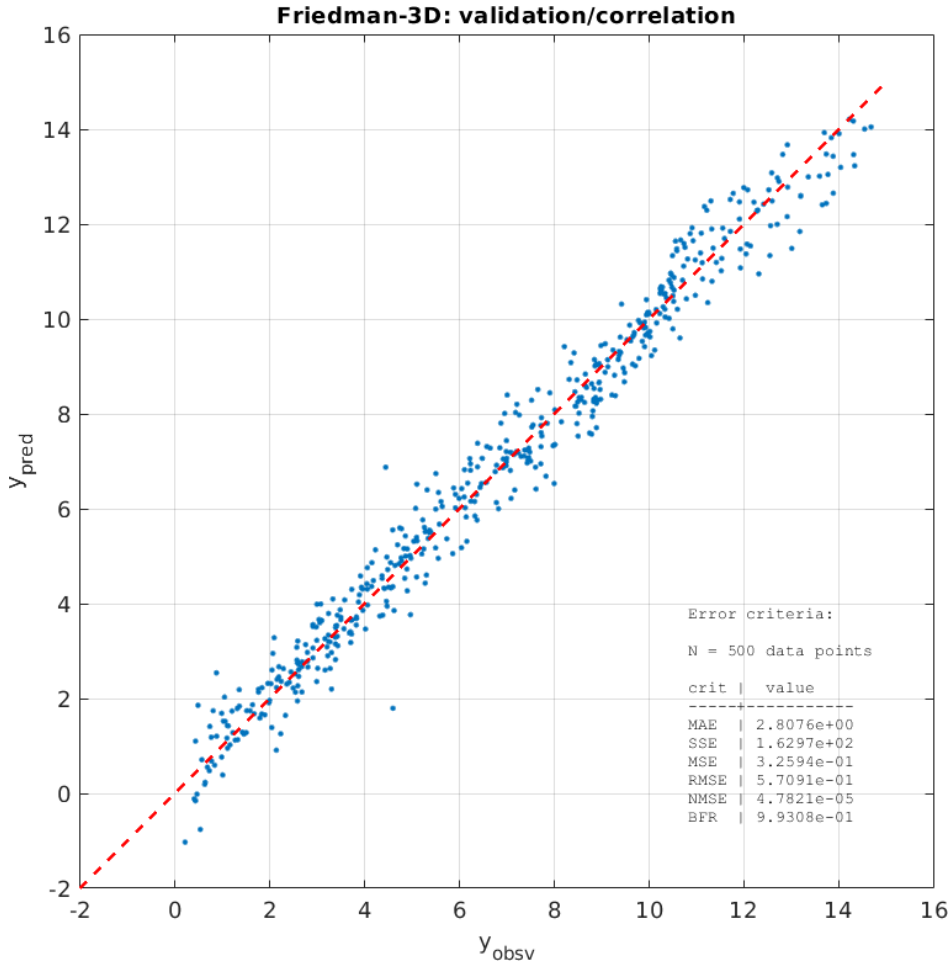


Predict the TS model output: y_{pred}

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

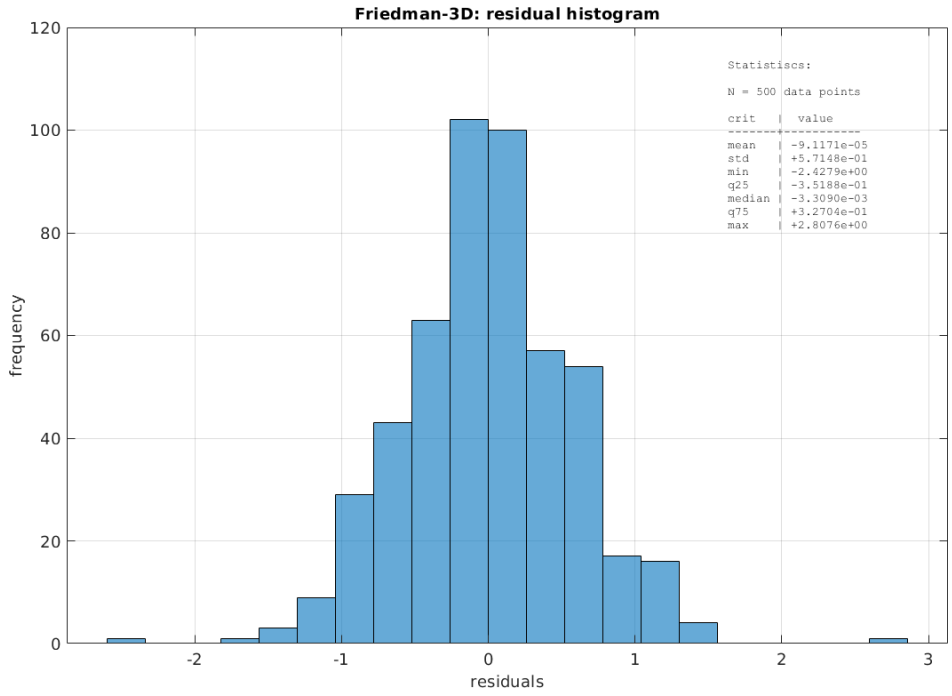
Plot the correlation

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



Plot the residual histogram:

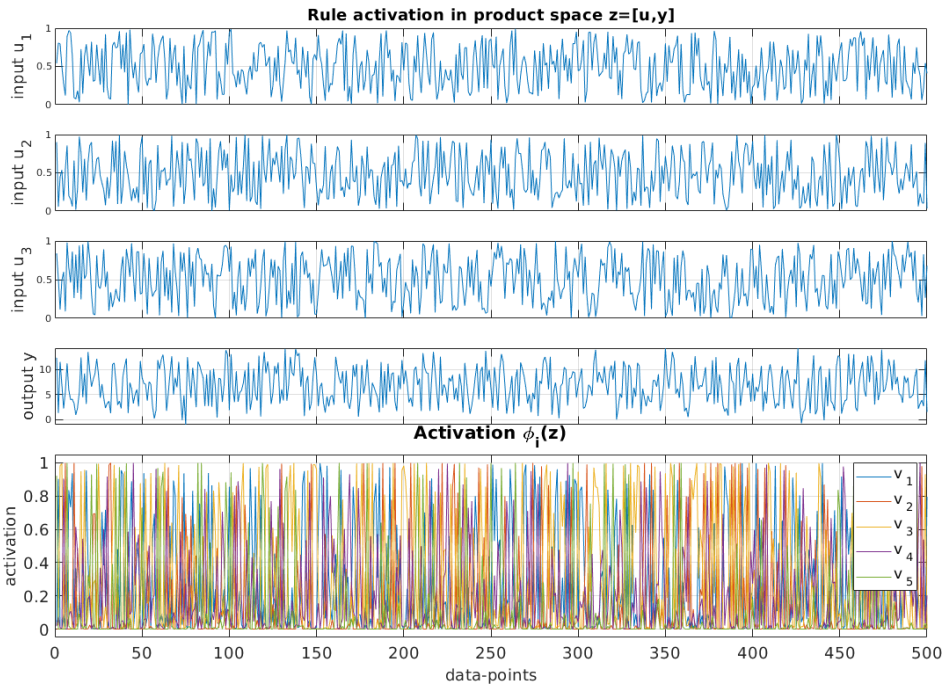
```
plotResidualHist( y, y_pred, 'figure', 3, 'nbins', 21, ...
    'title', 'Friedman-3D: residual histogram' );
set(gcf,'WindowState','maximized');
```



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 $[u_1, u_2, u_3]$

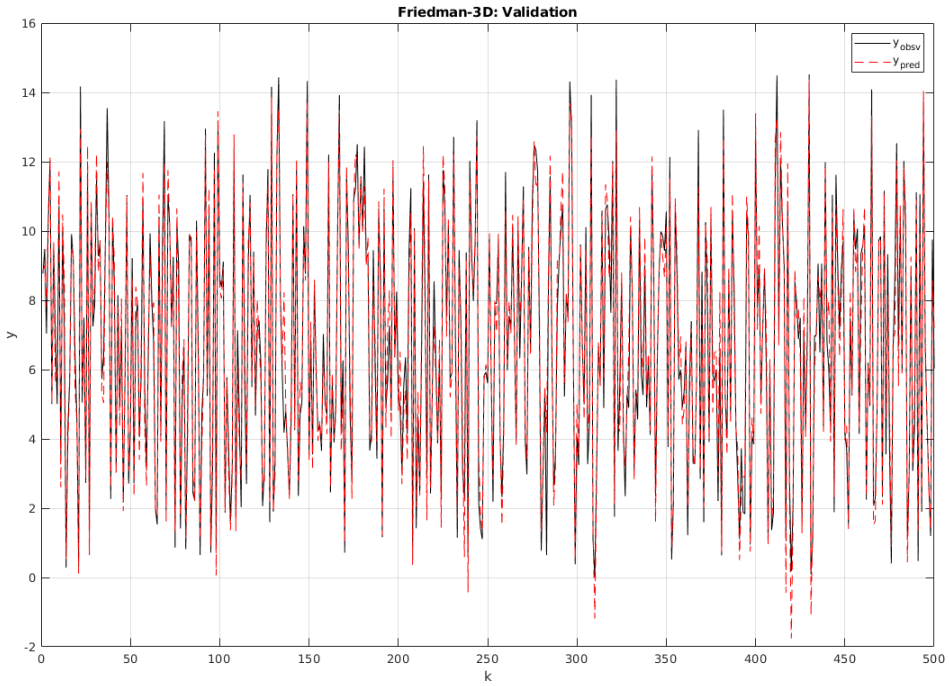
```
u_val = rand( N, nu );
y_obsv = Friedman_fct( u_val, nu );
```

Compute the output vector: y_{pred}

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

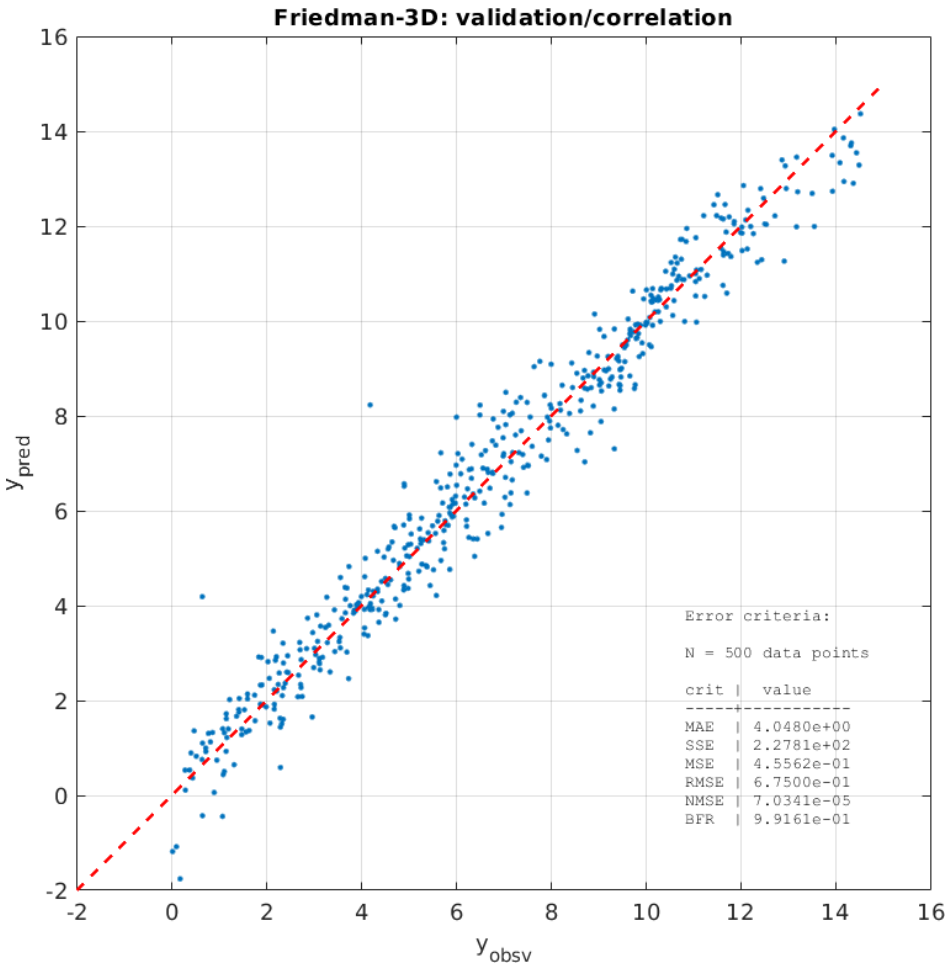
Plot the outputs

```
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_{obsv}','y_{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');
```