

# Takagi-Sugeno Model Identification Toolbox

March 10, 2021

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

V1.0

Axel Dürrbaum ([axel.duerrbaum@mrt.uni-kassel.de](mailto: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

<b>1</b>	<b>Identification data</b>	<b>1</b>
<b>2</b>	<b>Structural parameters</b>	<b>2</b>
<b>3</b>	<b>Estimation of the static LiP TS model</b>	<b>2</b>
<b>4</b>	<b>Validation of the TS model</b>	<b>6</b>

## 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

```
ts = TSMModel( 'Static', nv, nu, 'comment', 'Friedman 3D' );
```

Set the identification data:  $u, y$

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

Clustering:

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

```
ts.clustering( MSF, 'nue', fuzzy, '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( 'B' );
```

Iteration	Func-count	f(x)	Norm of step	First-order optimality
0	36	964.413		1.39e+03
1	72	729.457	10	774
2	108	546.454	20	638
3	144	546.454	25.8986	638
4	180	442.184	6.47464	281
5	216	348.212	12.9493	126
6	252	269.075	8.44771	103
7	288	220.296	6.48778	80
8	324	188.962	4.63655	88.4
9	360	176.265	1.77473	35.9
10	396	167.294	2.06021	20.5
11	432	160.624	1.89419	16.6
12	468	157.722	0.999579	15.4
13	504	156.268	0.688714	13.4
14	540	155.027	0.756757	14.7
15	576	153.471	0.992621	18.8
16	612	151.077	1.4415	24.6
17	648	147.443	2.23712	24.4
18	684	143.351	3.25085	11.4
19	720	141.379	2.17053	8.54
20	756	140.455	1.15225	7.58
21	792	139.771	0.972443	7.87
22	828	139.078	0.841047	8.63

23	864	138.177	0.758771	10.1
24	900	136.861	0.735133	12.3
25	936	134.915	0.781017	14.5
26	972	132.238	0.986927	14
27	1008	128.816	1.33059	15.5
28	1044	124.901	1.53975	19.7
29	1080	121.352	1.34374	13.9
30	1116	118.96	0.959655	8.49
31	1152	118.11	0.399527	7.09
32	1188	117.433	0.422381	5.99
33	1224	116.934	0.415374	4.92
34	1260	116.605	0.371034	3.93
35	1296	116.407	0.311904	3.05
36	1332	116.298	0.250074	2.28

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 )`

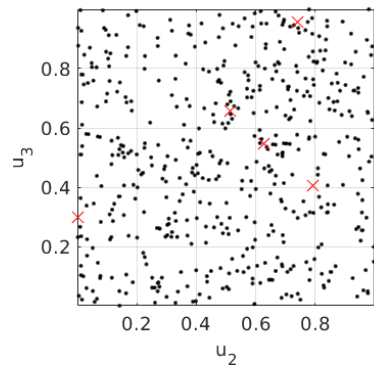
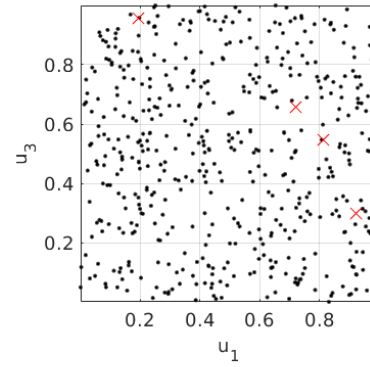
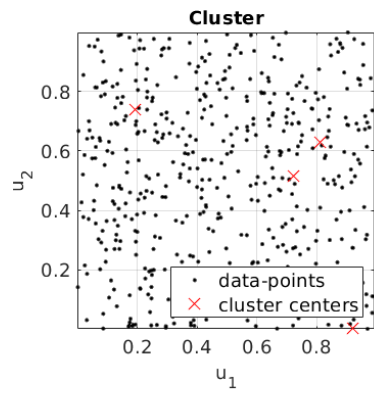
```
TS-Model: Type=Static
Name: 'undefined'
Type: 'TSMModel'
Date: '10-Mar-2021 14:29:49'
Comments:
  'Friedman 3D'
Structural parameters: nu = 3, ny = 1, nv = 5
Identification data: N=500
Initial model estimation:
  Clustering: FCM, nue=1.2 norm=Euclidian in product space
Estimation of local models:
  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* =

0.9201	0.0018	0.2979
0.8109	0.6285	0.5466
0.9988	0.7933	0.4067
0.7206	0.5145	0.6579
0.1935	0.7392	0.9542

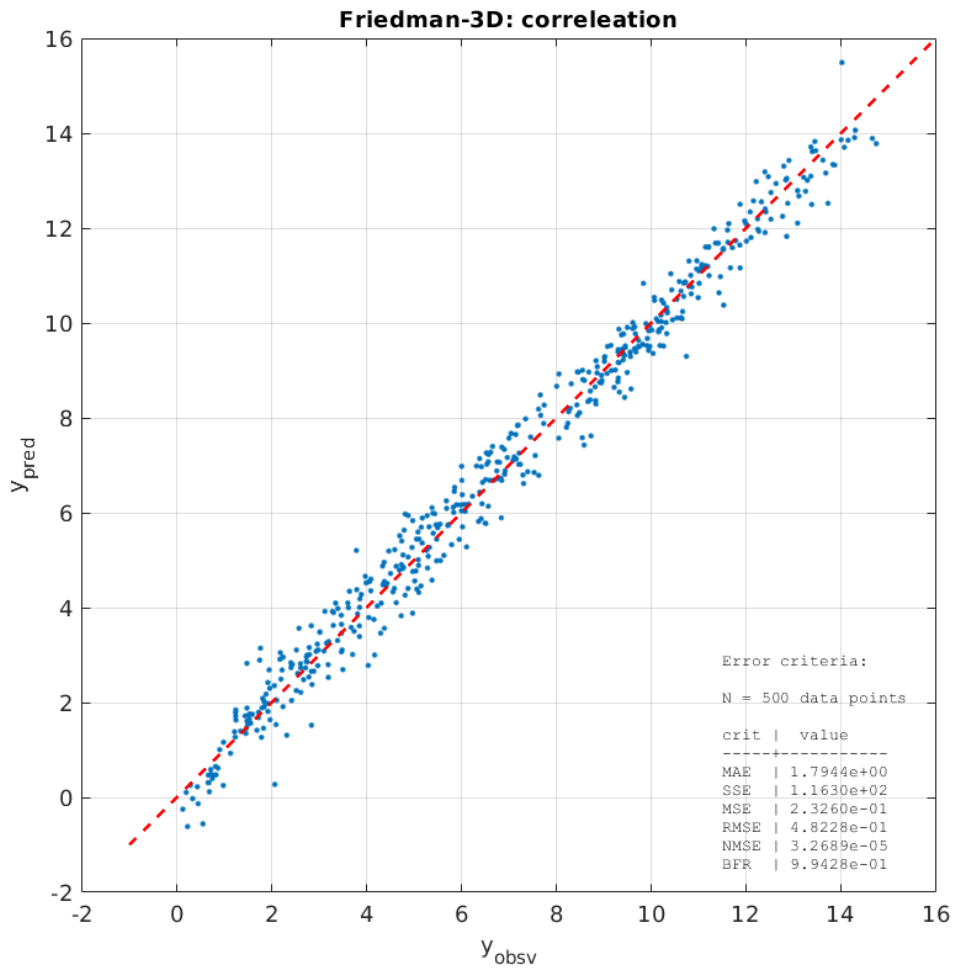


Predict the TS model output:  $y_{pred}$

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

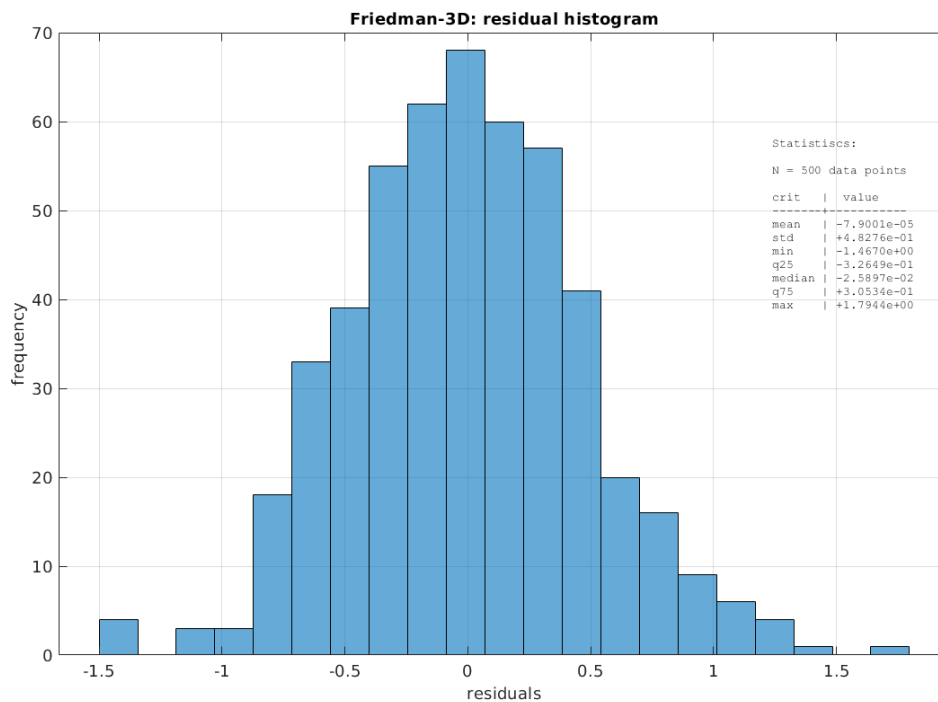
Plot the correlation

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



Plot the residual histogram:

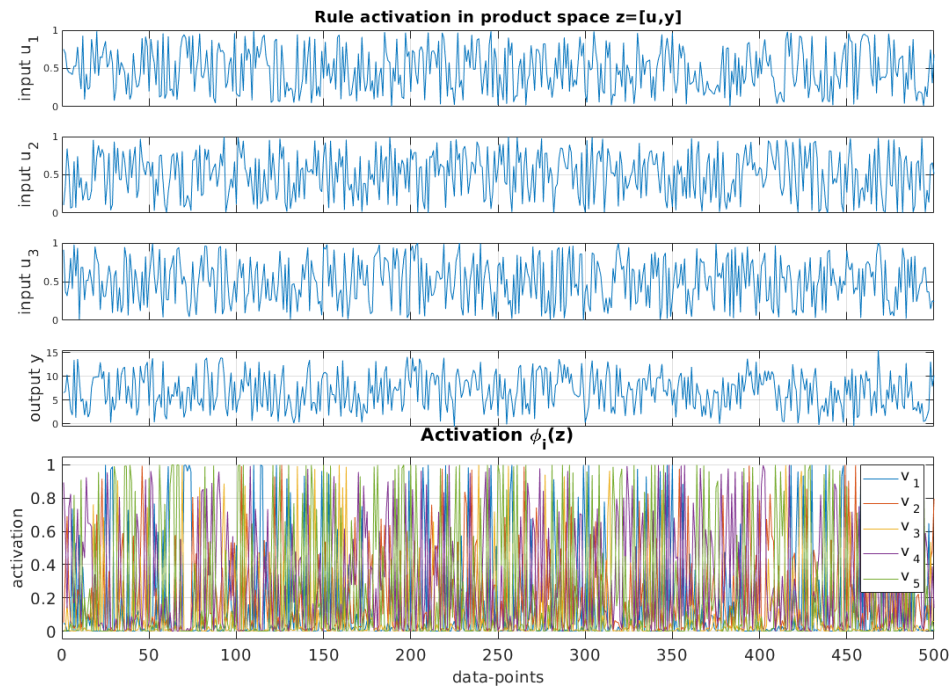
```
plotResidualHist( y, y_pred, 'figure', 3, 'nbins', 21, ...
    'title', 'Friedman-3D: residual histogram' );
set(gcf,'WindowState','maximized');
```



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  $[u_1, u_2, u_3]$

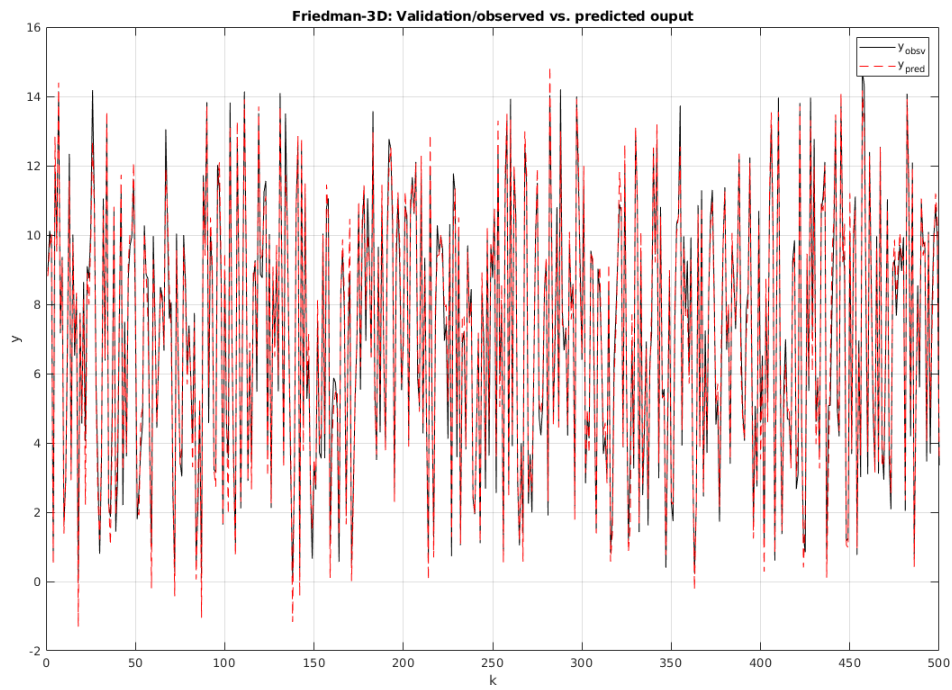
```
u_val = rand( N, nu );
y_obsv = Friedman_fct( u_val, nu );
```

Compute the output vector:  $y_{pred}$

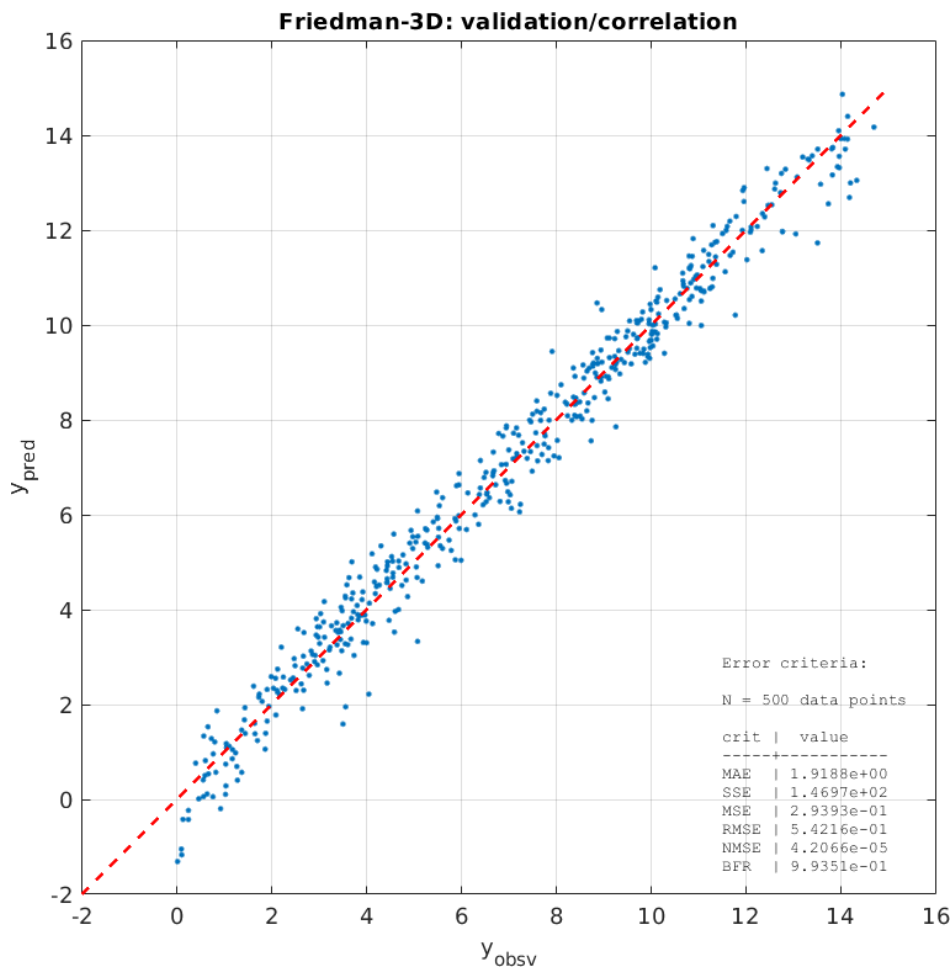
```
y_val_pred = ts.predict( u_val );
```

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/observed vs. predicted ouput' )
legend( 'y_{obsv}', 'y_{pred}' )
set(gcf,'WindowState','maximized');
```



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



```
plotResidualHist( y_obsv, y_val_pred, 'figure', 7, 'nbins', 31, ...
                  'title', 'Friedman-3D: validation/correlation histogram' );
set(gcf,'WindowState','maximized');
```

