Takagi-Sugeno Model Identification Toolbox

March 29, 2021

Example of a NOE TS model for the Narendra 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: NOE_Narendra.m | Fri Feb 26 16:25:05 2021 +0100 | Axel Dürrbaum $%

Contents

Example of the identification of a NOE MISO TS model for given multiple inputs u and single output y.

Determine the NOE TS model

$$\hat{y}_{k+1} = \sum_{i=1}^{n_v} \phi_i(z) \cdot \left(\sum_{l=1}^{l_y} A_i \cdot \hat{y}_{k-l} + \sum_{j=1}^{n_u} \sum_{l=0}^{l_u} B_{i,j} \cdot u_{k-l} + c_i \right) + e_k$$

- for given $u_j, j = 1, \ldots, n_u$ of n_u input vectors, output error e and
- input lags x_u with length l_u
- intial vector y of single output,
- output lags x_y with length l_y
- with FCM membership function

$$\mu_i(z) = \left(\sum_{j=1}^{n_v} \left(\frac{||z - v_i||}{||z - v_j||} \right)^{\dfrac{2}{\nu - 1}} \right)^{-1}$$

- or Gauss membership function

$$\mu_i(z) = e^{-\dfrac{||z - v_i||^2}{2 \cdot \sigma_i^2}}$$

- norm

$$||z - v_j|| = (z - v_j)^T \cdot w_j \cdot (z - v_j)$$

- and fuzzy basis functions

$$\phi_i(z) = \frac{\mu_i(z)}{\sum_{j=1}^{n_v} \mu_j(z)}$$

- with the scheduling variable $z = u$ (for input space clustering) or $z = [u, y]$ (for product space clustering), and
- cluster centers $v_i, i = 1, \ldots, n_v$.

1 Structural settings

```
nc = 3;    % number of clusters = local models
nu = 1;    % number of inputs
ny = 1;    % number of outputs
nue = 1.2; % fuzziness parameter
```

Scheduling lags z = regressor lags $x = [\hat{y}_{k-1}, \hat{y}_{k-2}, u_k]^\top$

```
z_lag_u = {0};
z_lag_y = [2];
x_lag_u = {0};
x_lag_y = [2];
```

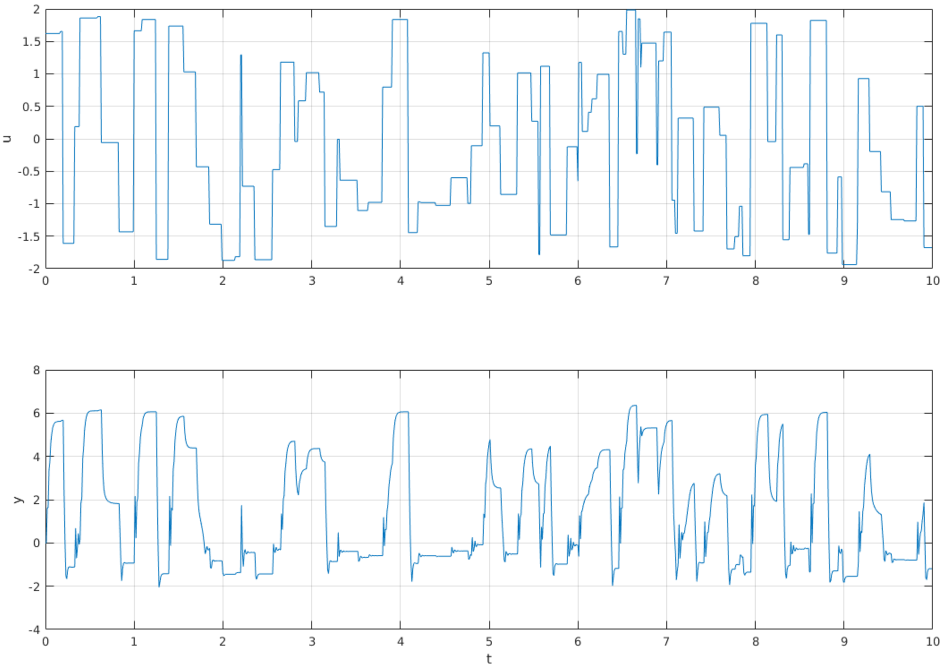
2 Identification data

Create input u as steps with width $l = [1, \ldots, 20]$ for $N = 1000$ time steps (sampling rate is 0.01) and compute the output y from the Narendra function

```
N = 1000;
rng(0);
[u,y] = Narendra_fct( N );
dt = 1e-2;                              % Sampling time
t = dt * transpose( 0:size(u,1)-1 ); % time vector $t$
```

Plot of the identification data

```
h=figure(1);clf

subplot(2,1,1)
plot(t,u)
grid on
ylabel('u')
subplot(2,1,2)
plot(t,y)
grid  on
ylabel('y')
xlabel('t')
```



3 Creation of TS model

```
addpath( '../TSModel' );  % Path to TSModel class
ts = TSModel( 'OE', nc, nu, 'Name','OE Narendra', 'Comment','Narendra function');
ts.setSchedulingLags( z_lag_u, z_lag_y );
ts.setRegressorLags( x_lag_u, x_lag_y );
```

Set the identification data

```
ts.setData( u, y, 'SampleTime',dt, 'Labels', { 'u', 'y' } );
ts.setDataLimits( [-2,2 ; -5,10] );
```

4 Clustering

Clustering in product-space $z = [u, y]$ with FCM membership functions and $\nu = 1.2$ with $s = 3$ multi-start tries and ficed initialized random number generator (seed 0)

```
ts.clustering( 'FCM', 'nue', nue, 'tries',3, 'seed', 0 )
```

```
ans =
TS-Model: Type=OE
Name: 'OE Narendra'
 Type: 'TSModel'
 Date: '29-Mar-2021 15:51:43'
 Comments:
   'Narendra function'
 Structural parameters: nu = 1, ny = 1, nv = 3
 Identification data: N=1000
, ts=0.01 Initial model estimation:
 lags: u_1:0, y = 2
  Membership function type = FCM
  Clustering: FCM, nue=1.2 norm=Euclidean in input space
 Estimation of local models:
 lags:    u_1:0,    y = 2
```

Cluster centers of the inital model

```
v1 = getCluster( ts )
```

```
v1 =
   -0.7733   -0.9077
    2.0289    0.3502
    5.0925    1.0995
```

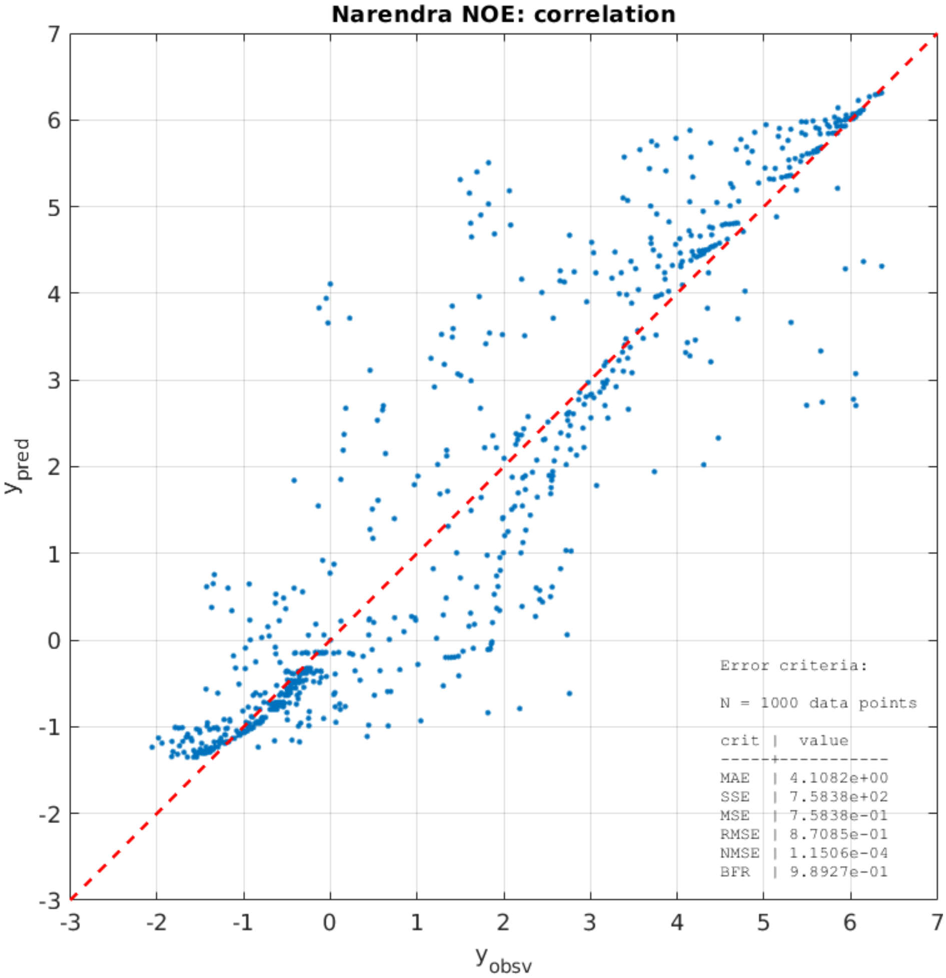
5 Initialization of local models

with global Least-Squares, FCM membership functions and $\nu = 1.2$

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

6 Predicted NOE TS model ouput

```
y_pred = ts.predict( u,y );
plotResiduals( y, y_pred, 'figure', 2, 'title', 'Narendra NOE: correlation' );
set(gcf,'WindowState', 'maximized' );
```

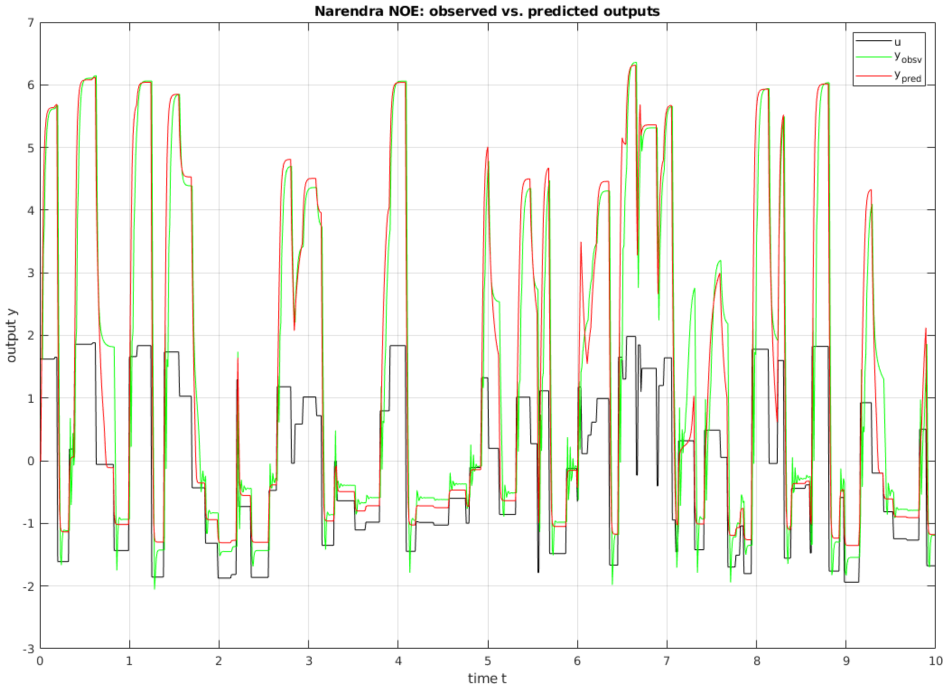


```
Error criteria:

N = 1000 data points

crit |  value
-----+------------
MAE  | 4.1082e+00
SSE  | 7.5838e+02
MSE  | 7.5838e-01
RMSE | 8.7085e-01
NMSE | 1.1506e-04
BFR  | 9.8927e-01
```

Plot of the observed vs. predicted outputs

```
figure(3);clf

plot(t,u,'k-',t,y,'g-',t,y_pred,'r-')
grid on
xlabel( 'time t' )
ylabel( 'output y' )
```

```
title('Narendra NOE: observed vs. predicted outputs')
legend('u','y_{obsv}','y_{pred}')
set(gcf,'WindowState', 'maximized' );
```

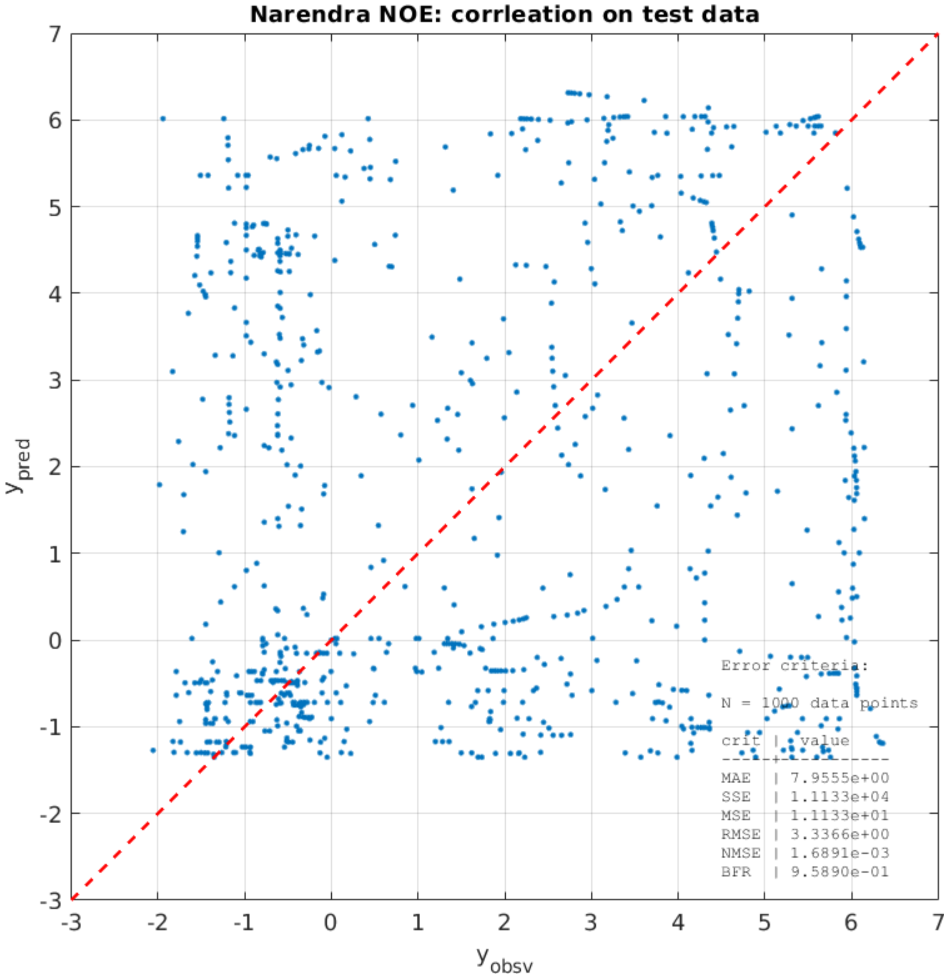


7 Prediction on validation data

```
[u_val,y_val] = Narendra_fct( N );
y_val_pred = ts.predict( u_val,y_val );
```

Plot the correlation

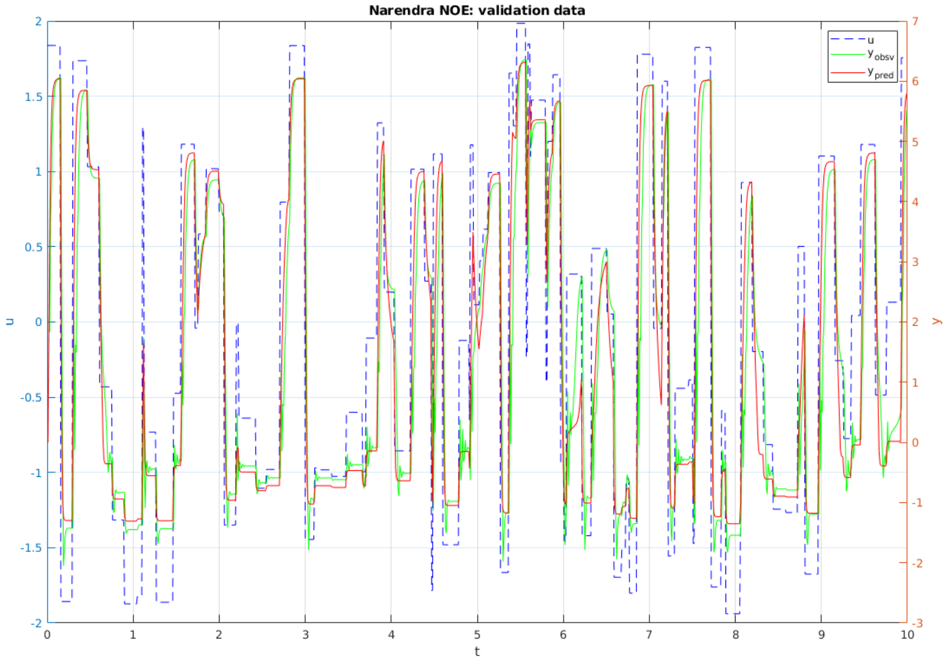
```
plotResiduals( y, y_val_pred, 'figure', 4, 'title', 'Narendra NOE: corrleation on test data' );
set(gcf,'WindowState', 'maximized' );
```



```
Error criteria:

N = 1000 data points

crit | value
-----+-----------
MAE  | 7.9555e+00
SSE  | 1.1133e+04
MSE  | 1.1133e+01
RMSE | 3.3366e+00
NMSE | 1.6891e-03
BFR  | 9.5890e-01
```

Plot of observed vs. predicted validation data

```
figure(5);clf

yyaxis left
plot(t,u_val,'b--')
ylabel( 'u' )
yyaxis right

plot(t,y_val,'g-',t,y_val_pred,'r-')
ylabel( 'y' )
xlabel( 't' )

grid on
title('Narendra NOE: validation data')
legend('u','y_{obsv}','y_{pred}')
set(gcf,'WindowState', 'maximized' );
```



8 Optimize the TS model parameters

Set additional parametrs for function `lsqnonlin`

```
optimopts = optimoptions('lsqnonlin');
optimopts.FunctionTolerance = 1e-6;
optim.OptimalityTolerance = 1e-6;
optimopts.StepTolerance = 1e-12;
optimopts.Display = 'iter-detailed';
```

Optimize both, the cluster centers v (MF) and the local model parameters A_i, B_i, c_i

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

```
                                              Norm of      First-order
 Iteration  Func-count     f(x)              step          optimality
     0           16      758.382                               802
     1           32      758.382            1.9881             802
     2           48      526.799          0.497024             413
     3           64      361.411          0.960259             359
     4           80       277.15          0.425853             497
     5           96      271.183          0.114441            99.4
     6          112      270.479         0.0512704            42.5
     7          128      270.316         0.0405038              39
     8          144      270.222         0.0271255            22.5
     9          160      270.187          0.021413            18.5
    10          176      270.121        0.00535325            10.5
    11          192      270.114        0.00574168            3.98
    12          208      270.113        0.00411673             3.4
    13          224       270.11        0.00102918            2.73
    14          240       270.11        0.00136231            1.06
    15          256      270.109       0.000957319           0.898
    16          272      270.109        0.00023933           0.819

Optimization stopped because the relative sum of squares (r) is changing
by less than options.FunctionTolerance = 1.000000e-06.
```

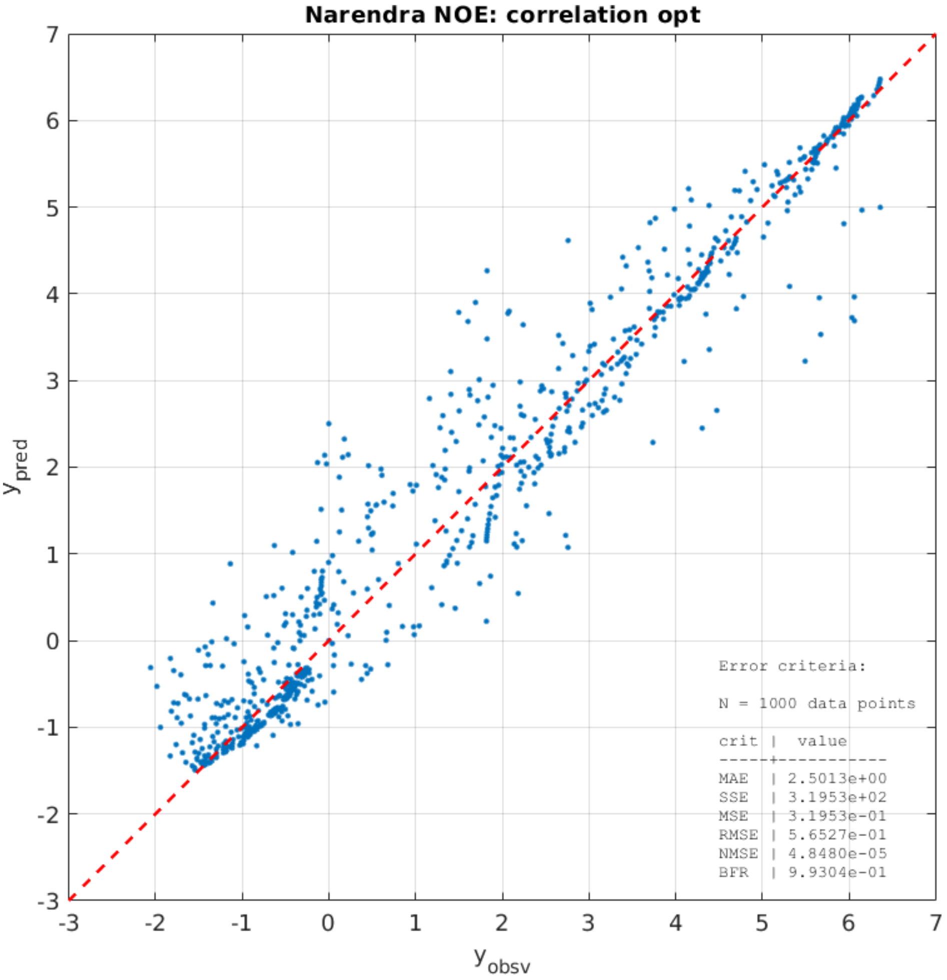
Get the cluster centers of the optimized NOE TS model

```
v2 = getCluster( ts )
```

```
v2 =
   -0.7733   -0.9077
         0    0.3502
    5.0925    1.0995
```

Plot the correlation on the validation data

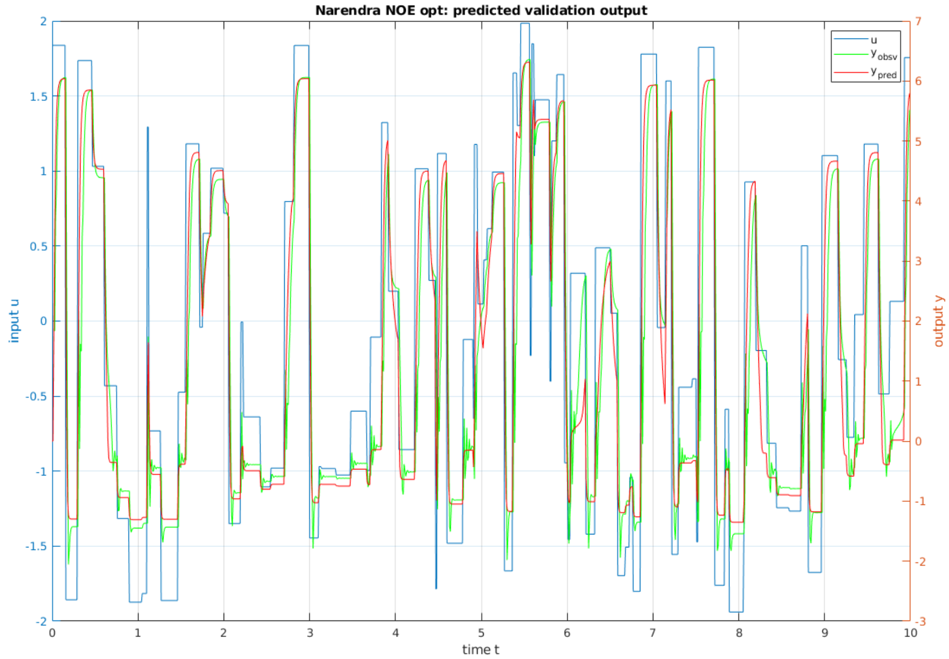
```
y_pred_opt = ts.predict( u,y );
plotResiduals( y, y_pred_opt, 'figure', 5, 'title', 'Narendra NOE: correlation opt' );
set(gcf,'WindowState', 'maximized' );
```



```
Error criteria:

N = 1000 data points

crit |  value
-----+-----------
MAE  | 2.5013e+00
SSE  | 3.1953e+02
MSE  | 3.1953e-01
RMSE | 5.6527e-01
NMSE | 4.8480e-05
BFR  | 9.9304e-01
```

Plot the observed vs. the predicted validation data

```
figure(6);clf
yyaxis left
plot(t,u_val)
ylabel( 'input u' )
yyaxis right
plot(t,y_val,'g-',t,y_val_pred,'r-')
grid on
ylabel( 'output y' )
xlabel( 'time t' )
title('Narendra NOE opt: predicted validation output')
legend('u','y_{obsv}','y_{pred}')
set(gcf,'WindowState', 'maximized' );
```



Error criteria

```
ec_val = ErrorCriteria( y_val_pred,y_val)
```

```
ec_val = 
  struct with fields:

     MAE: 4.1082
     SSE: 718.8470
     MSE: 0.7188
    RMSE: 0.8478
    NMSE: 1.1038e-04
     BFR: 0.9895
     AIC: NaN
     BIC: NaN
```