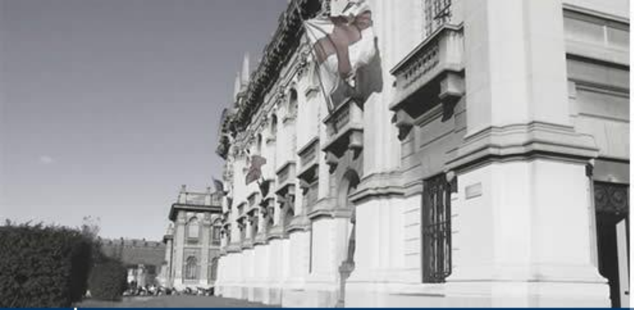
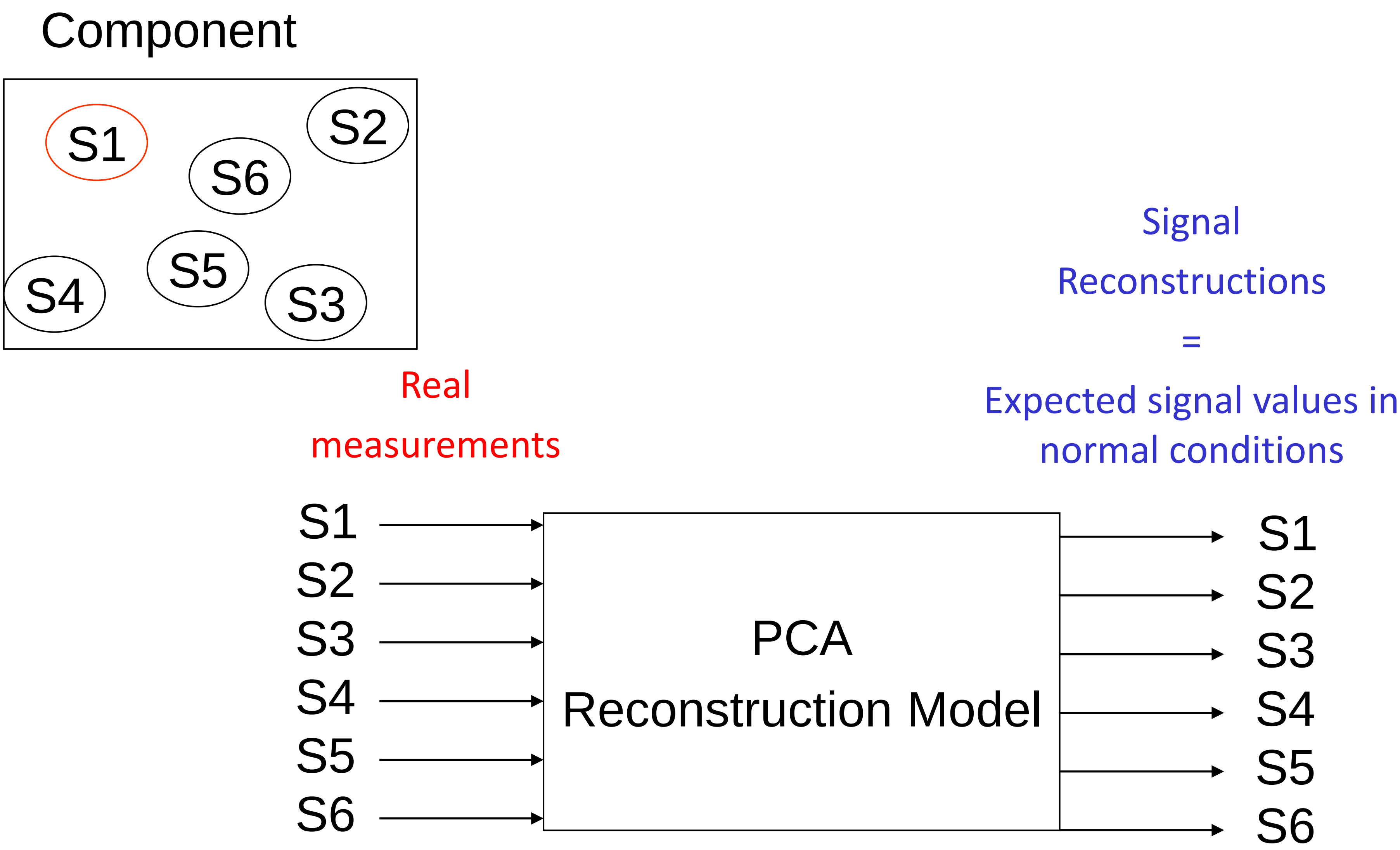


POLITECNICO DI MILANO

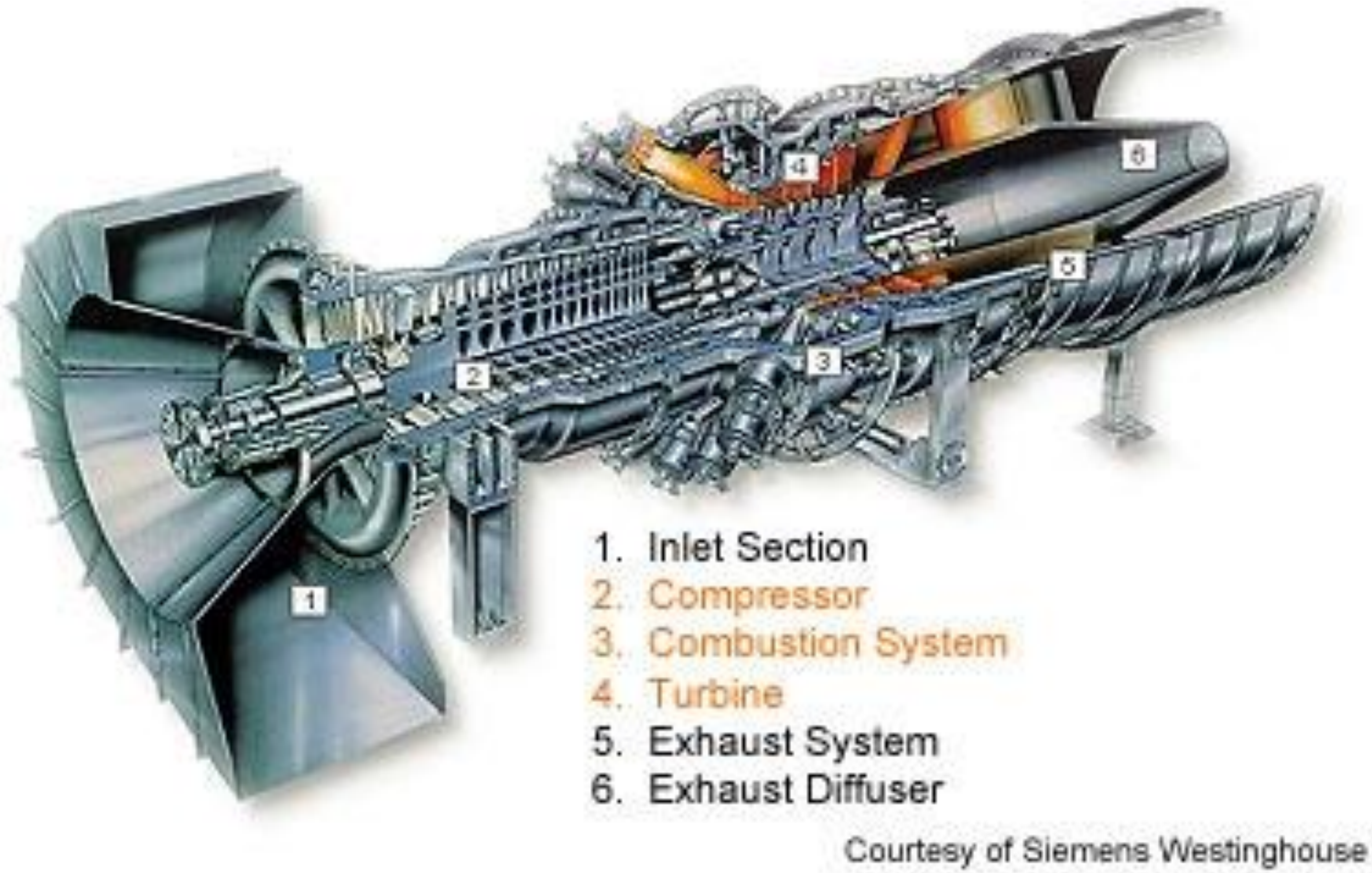
Exercises on Advanced regression techniques of signal analysis for fault detection:

PCA

Federico Antonello
federico.antonello@polimi.it


Component
S1
S2
S6
S5
S4
S3
Real measurements
Signal Reconstructions = Expected signal values in normal conditions
S1
S2
S3
S4
S5
S6
PCA Reconstruction Model
S1
S2
S3
S4
S5
S6

Turbine


1
2
3
4
5
6
1. Inlet Section
2. Compressor
3. Combustion System
4. Turbine
5. Exhaust System
6. Exhaust Diffuser
Courtesy of Siemens Westinghouse

Measured Signals

Temperature location 1 (°C)
Temperature location 2 (°C)
Temperature location 3 (°C)
Temperature location 4 (°C)
Temperature location 5 (°C)
Temperature location 6 (°C)

PCA

Exercise 1

Fill the missing parts of the code PCA_to_be_filled.m

Some useful matlab commands:

- cov
- eig

Exercise 1 TRAINING PHASE: P matrix

%%%

% STEP 1: find the matrix P whose column are the eigenvectors ordered

from the largest (column 1) to the smallest (column n)

% SIZE of P is [n,n]

%%%

Exercise 1 TRAINING PHASE: P matrix Solution


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
V=cov(train_data_n); % Covariance matrix of the normalized data
----------------------------------------------------------------------
%compute eigenvalues and eigenvectors of the Covariance matrix V
[P_rev,D] = eig(V);
eig_val_rev = diag(D); %found eigenvalues are ordered from the smallest to the largest
[a,b]=sort(eig_val_rev);

%change the ordering of eigenvalues and eigenvectors: from the largest to the smallest
for ii=1:n
    P(:,ii)=P_rev(:,b(n-ii+1));
    eig_value(ii)=eig_val_rev(b(n-ii+1));
End

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Exercise 1 TRAINING PHASE: PCA Approximation

%%%

% **STEP 2: perform the PCA approximation**

% keep a number (lambda) of eigenvestors such that they represent at least perc_var of the total data variance

=

% Find the matrix P_lambda whose columns are the first lambda aigenvectors

% Assuming to keep only lambda PCs, size of P_lambda is [n,lambda]

%%%

- How_many principal components should be considered?
  - Untill variance reaches var_th (the variance of a single component is equal to its eigenvalue)

Exercise 1 TRAINING PHASE: PCA Approximation Solution

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    STEP 2: perform the PCA approximation
sum_eig=sum(eig_value);
n_PC=0;
sum_var=0;
while sum_var<var_th;
    n_PC=n_PC+1;
    sum_var=sum_var+eig_value(n_PC)/sum_eig;
    if n_PC==n % to avoid numerical problem from the sum of the eigenvalues
        sum_var=1
    end
end
%---------------------------Keep only the first n_comp principal components
P_lambda=P(:,1:n_PC);
%end training phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Exercise 1 TEST PHASE:
RECONSTRUCT THE TEST PATTERNS

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     STEP 3: RECONSTRUCT THE TEST PATTERNS

%  TO BE FILLED

%     RECONSTRUCT THE TEST PATTERNS

%     Call 'test_data_rec_n' the matrix containing the reconstruction of

%     the test patterns. Size of test_data_rec_n is [N_test,n]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Exercise 1 TEST PHASE:
RECONSTRUCT THE TEST PATTERNS Solution


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    STEP 3: RECONSTRUCT THE TEST PATTERNS

%-----------------------------------------------------------------
%Reconstruct the test patterns by projecting in the trasformed basis, keep only the
% principal components and antitrasform

test_data_rec_n=test_data_n*P_lambda*P_lambda';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Exercise 2:

Consider the patterns in file ‘train.dat’ to develop a PCA reconstruction model. The file contains data collected during normal conditions. They are 6 temperature signals, 5200 measuraments/year.

- Develop the PCA reconstruction model
- Perform the reconstruction of the file ‘validation.dat’. The file contains data collected during normal conditions

Hints:

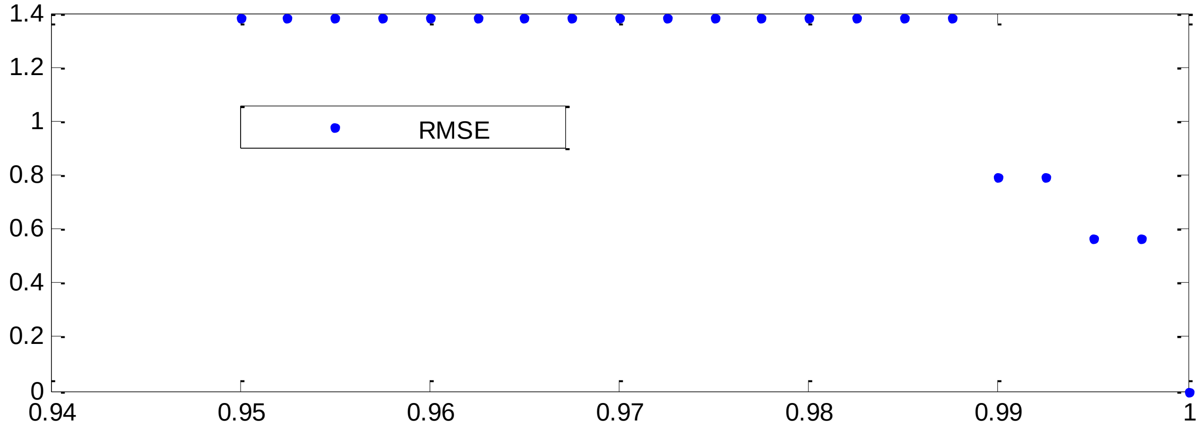
1. Perform the computation considering different numbers of Principal Components
2. Consider the root mean square error as a performance measure. Two tests should be performed:
   - *Input = normal condition signals*
   - *Input = simulated abnormal conditions*

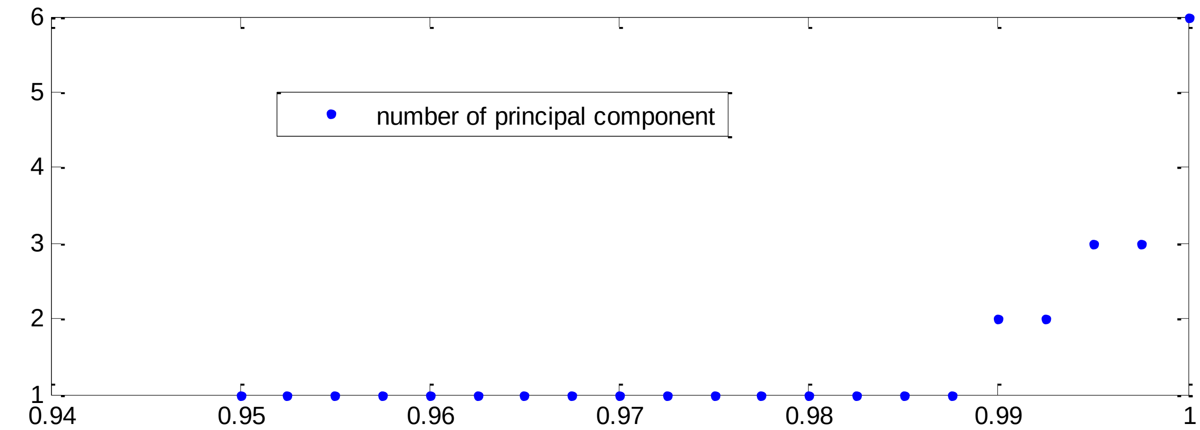
[test_reconstruction,n_PC]=PCA_reconstruction(‘train.dat’,’validation.dat’,0.95)

Exercise 2: Accuracy

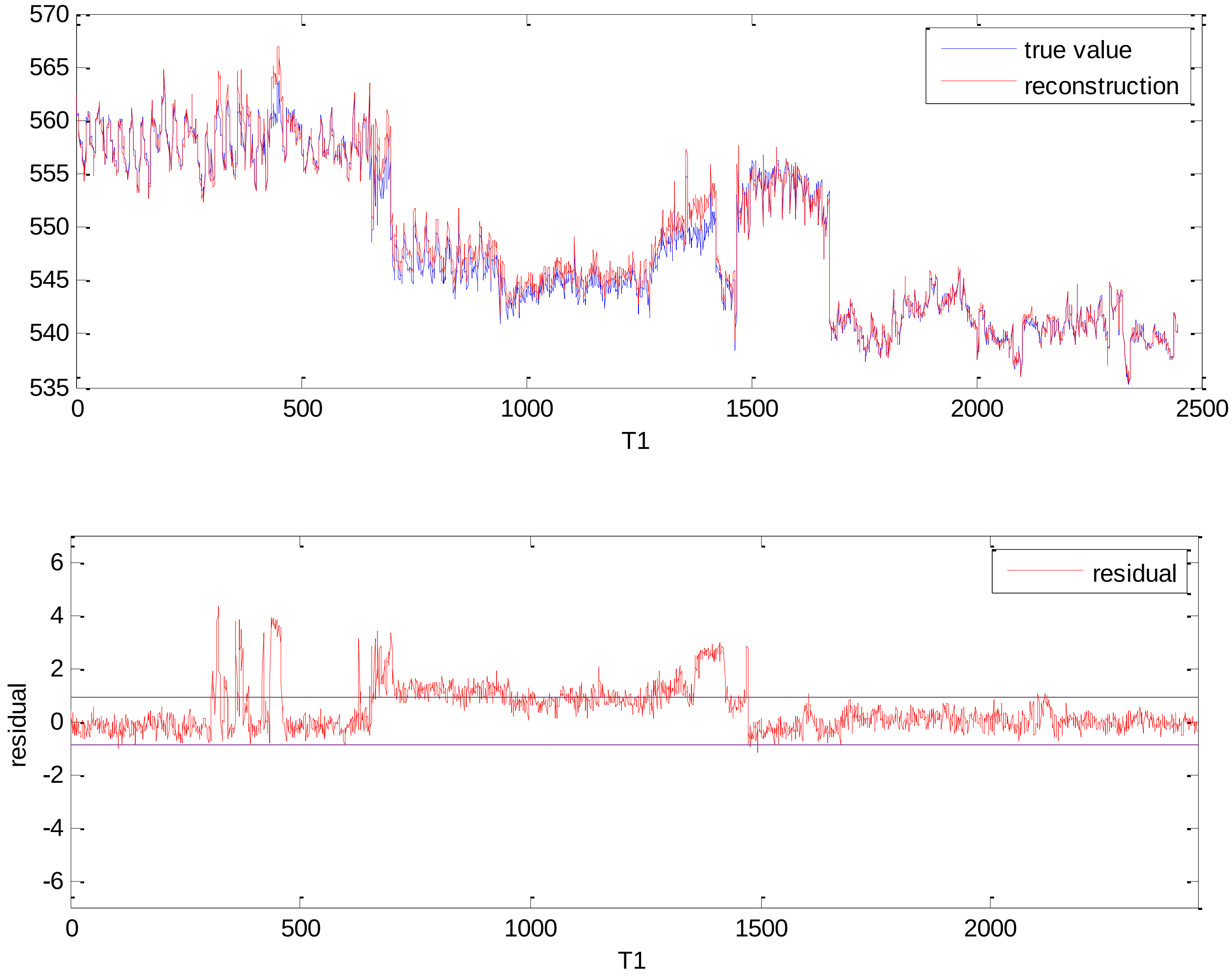
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Clear all
Close all
true_signal=load('validation.dat');
[N,n]=size(true_signal);
v=[0.98,0.99,0.995,1]
for i=1:4;
    [test_reconstruction,n_PC(i)]=PCA_reconstruction('train.dat','validation.dat',v(i));
 % Compute the root mean square error:
    rmse(i)=sqrt(sum(sum((test_reconstruction-true_signals).^2))/(n*N));

close all
end
figure(1)
% subplot(211)
plot(v,n_PC,'.')
legend('number of principal components')
figure(2)
plot(v,rmse,'r.')
legend('RMSE')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Exercise 2: Optimal Choice of the number of Principal components (test for the accuracy)

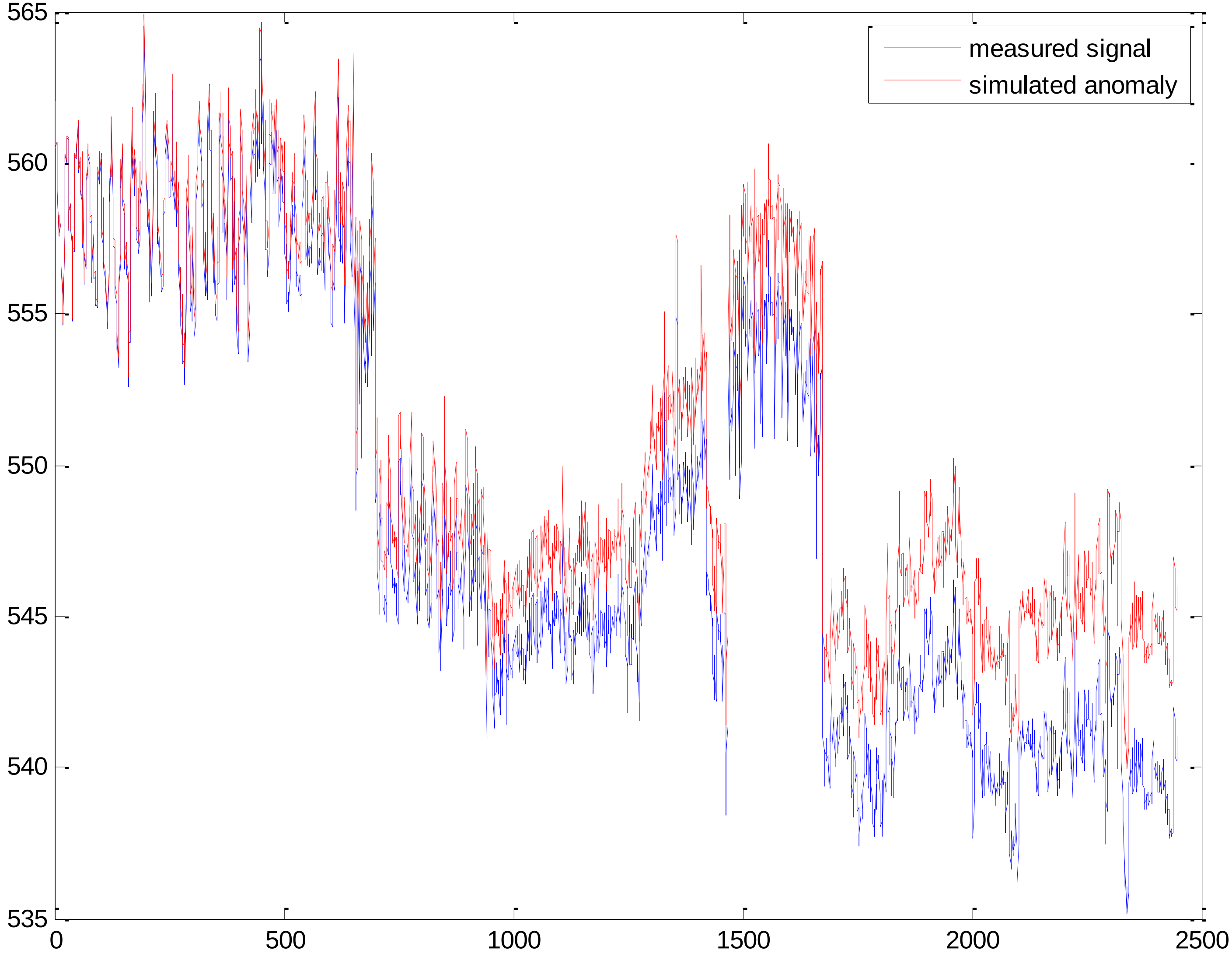




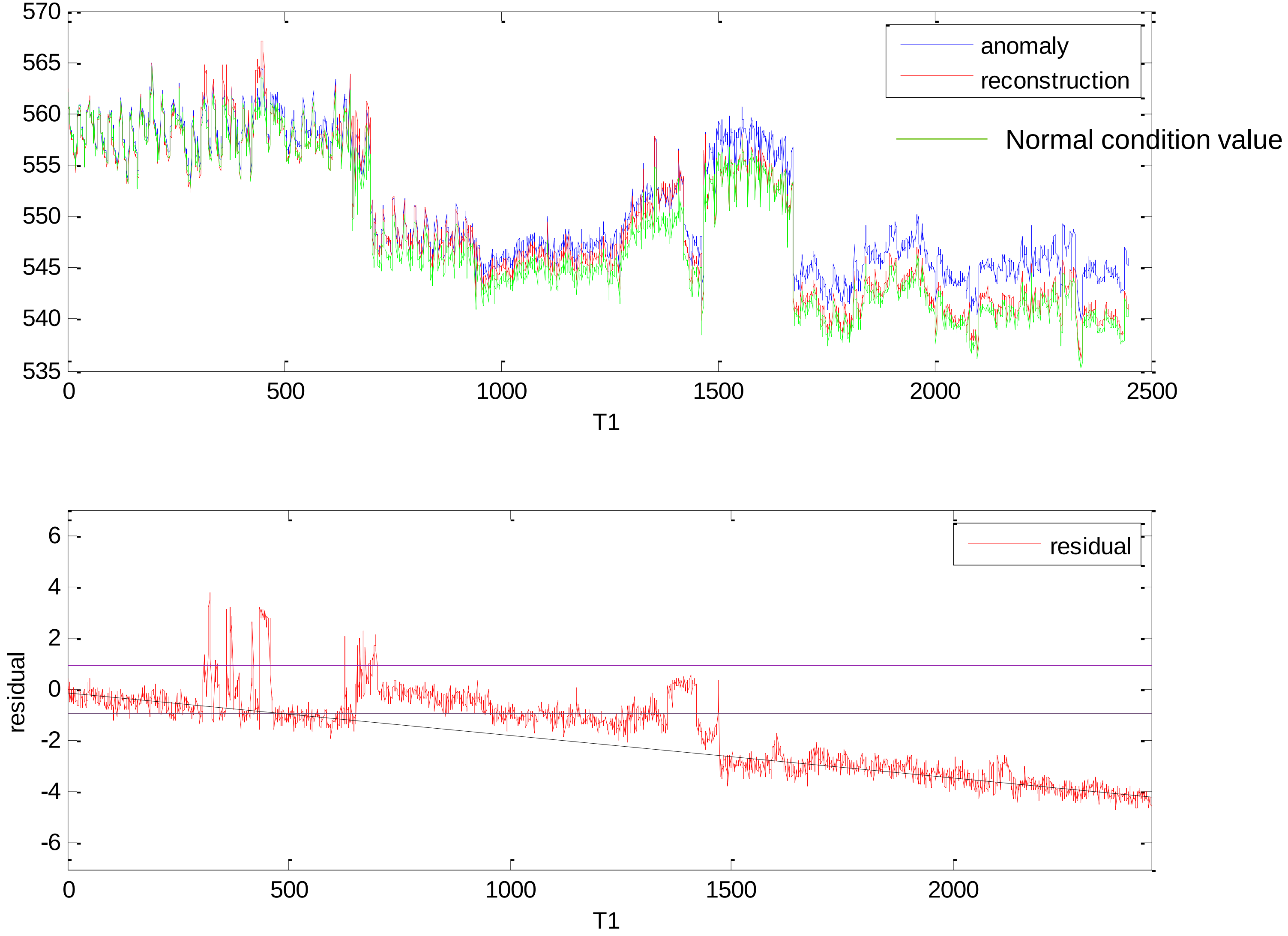
Reconstruction of the normal condition: 1 PC



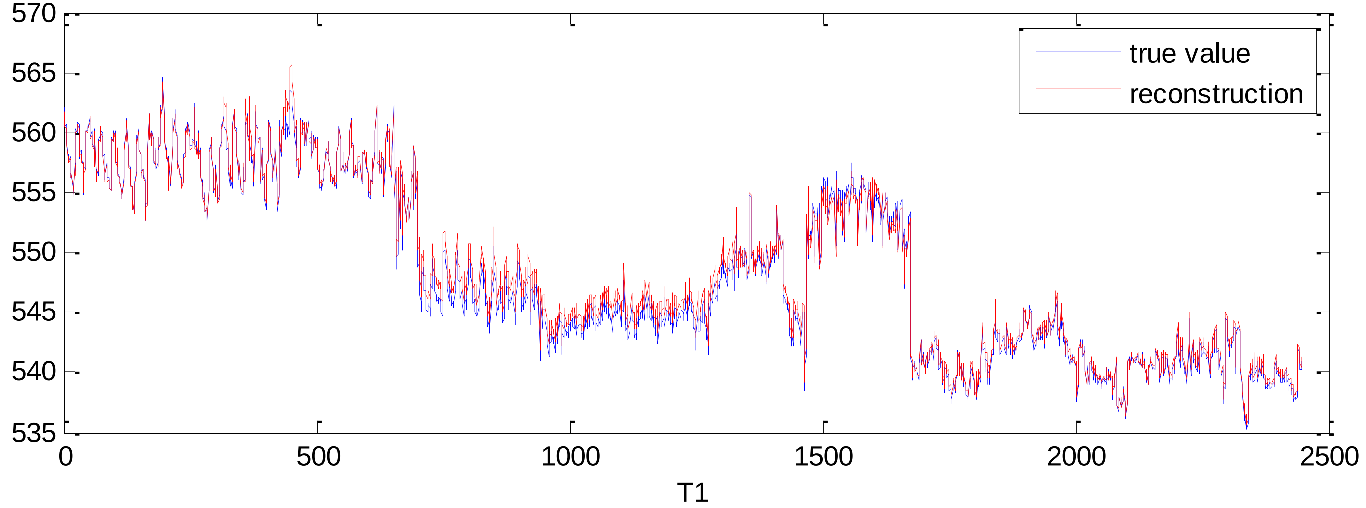
Exercise 2: simulated abnormal conditions

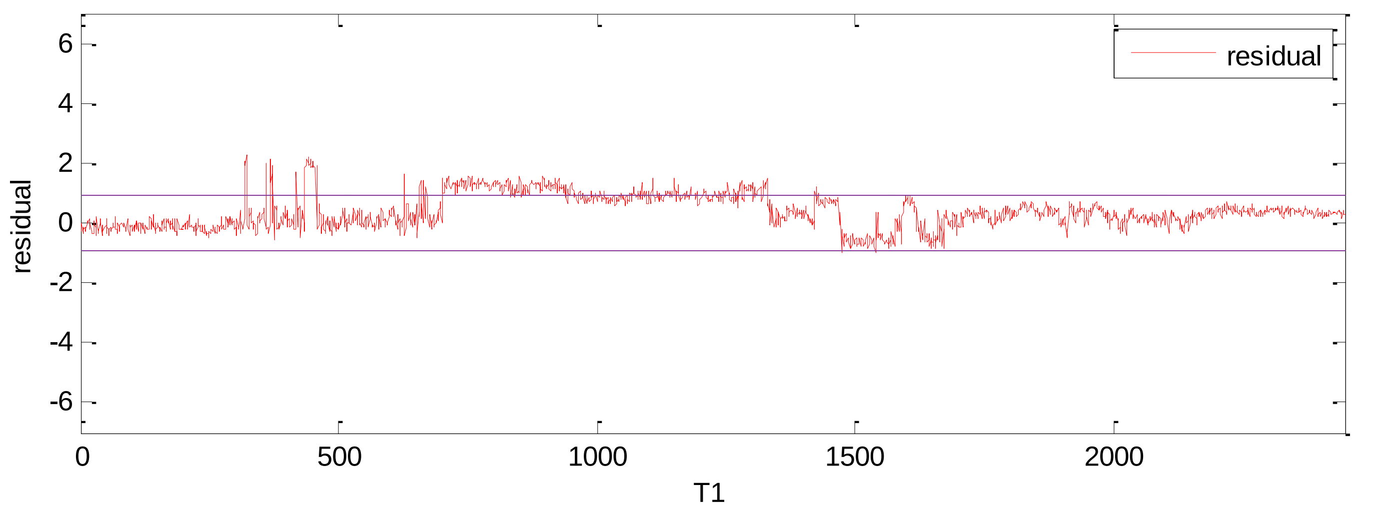


Reconstruction of the anomaly: 1 PC

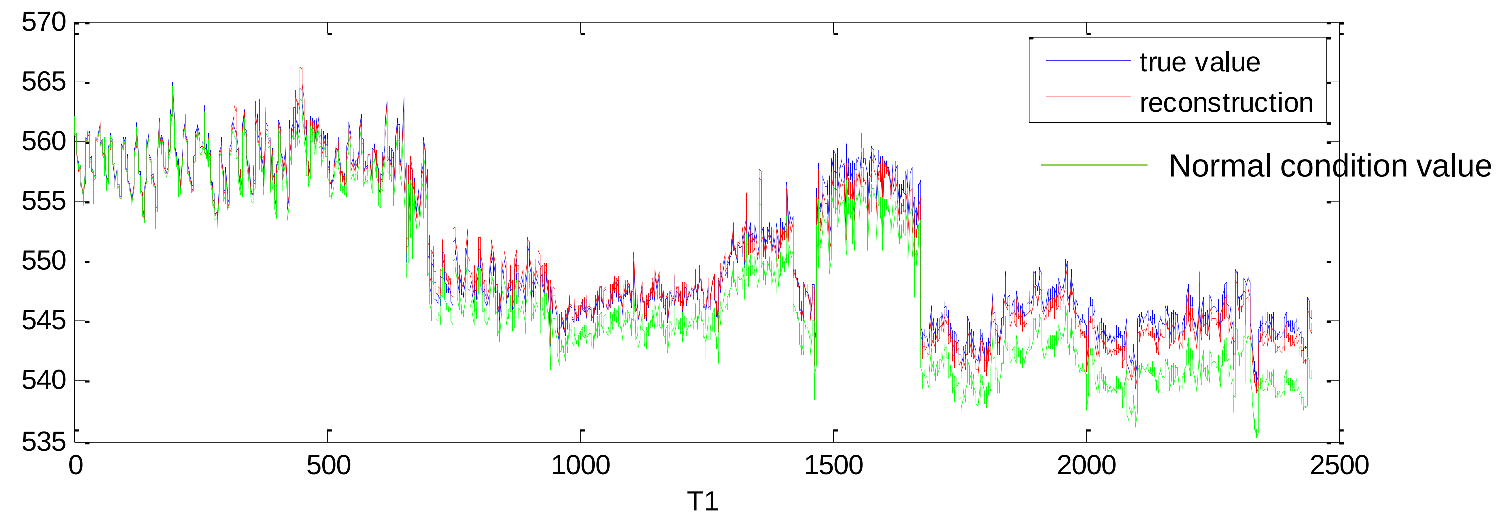


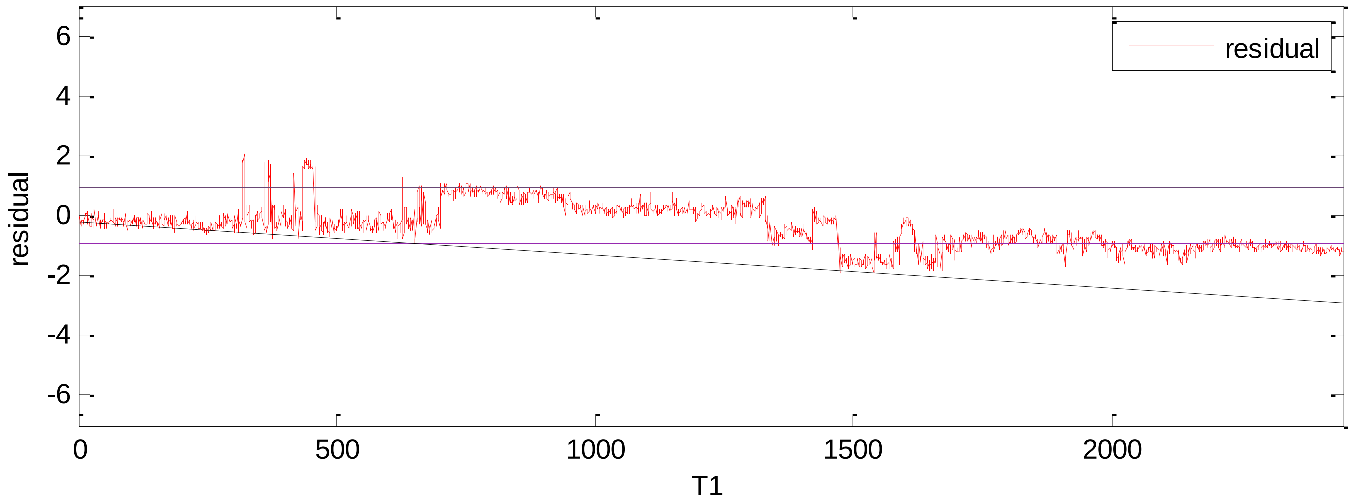
Reconstruction of the normal condition: 3 PC





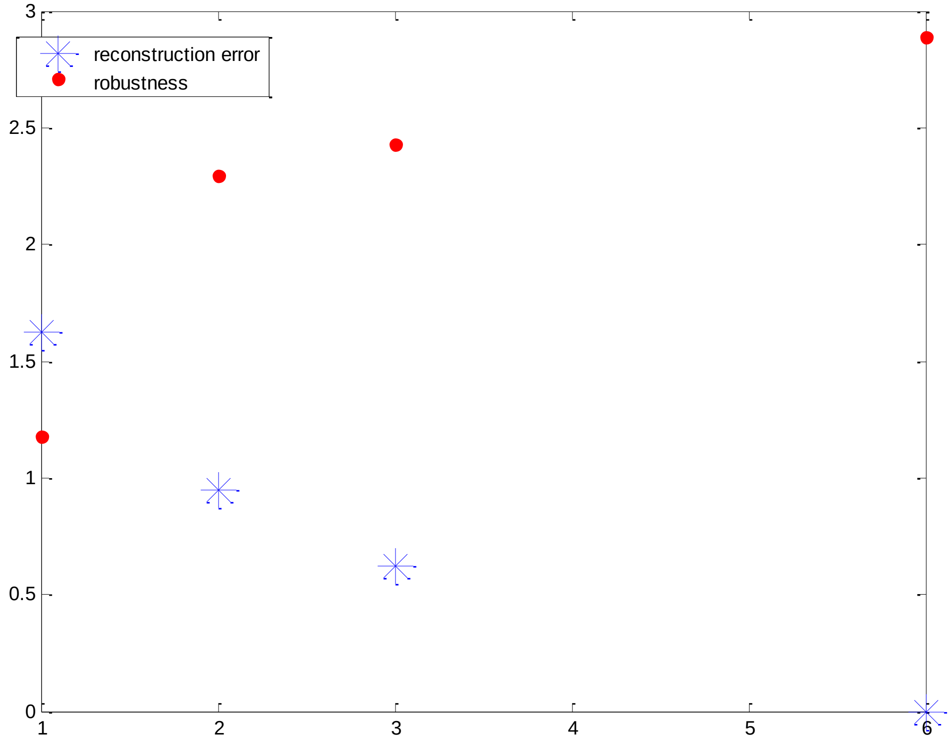
Reconstruction of the anomaly: 3 PC





Measure for robustness:

$$\frac{\sqrt{\sum_{signals}\sum_{samples}(Recontruction_{value} - true_{value})}}{N_{samples}}$$



number of PCs

Exercise: try to plot this figure

Accuracy and robustness Versus Number of Principal Components

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
true_signal=load('validation.dat');
[N,n]=size(true_signal);
v=[0.98,0.99,0.995]
for i=1:4;
    [test_reconstruction,n_PC(i)]=PCA_reconstruction('train.dat','validation.dat',v(i));
    rmse(i)=sqrt(sum(sum((test_reconstruction-true_signals).^2))/(n*N));
    [test_reconstruction,n_PC(i)]=PCA_reconstruction('train.dat','val_anomaly.dat',v(i));
    robustness(i)=sqrt(sum(sum((test_reconstruction(:,1)-true_signal(:,1)).^2))/(N));
    close all
end

figure
% subplot(211)
plot(n_PC,rmse,'.')
hold on
plot(n_PC,robustness,'r.')

legend('reconstruction error','robustness')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Exercise 3

- Perform the reconstruction of the signal measurements in the 3 files test_1.dat, test_2.dat, test_3.dat
- In which files can you detect abnormal conditions? Do you have any hypothesis on the type of abnormal condition?
- Draw your conclusions on the possibility of using the developed model for fault detection