Signal Analyzer, the software support for education of signal processing and measurements

Extended by Scripts

Jiří Tůma
Faculty of Mechanical Engineering
Department of Control Systems and Instrumentation
Outline

- Signal processing branches
- Requirements for education software
- Basic principle of handling with Signal Analyzer
- Input signals and instruments
- Output plots
- Teaching support
- Research work
- Conclusion
Signal Processing Branches

- Overall data analysis (A-, B-, C-type filtration, Lin)
- Fourier Transform, Hilbert Transform
- Spectral analysis, averaging in the frequency domain
- Spectral maps evaluation, auto- and cross-correlation analysis
- Frequency responses evaluation
- Envelope analysis and phase demodulation
- Rotational speed evaluation from an impulse signal
- Data resampling, averaging in the time domain
- Order analysis based on FFT or Vold-Kalman filter
- Filtration in the time and frequency domain, FIR filtering, wavelets
- Quadrature mixing for envelope and phase analysis
- Kalman filter for estimating a random constant
- Eigenanalysis (Pisarenko’s and MUSIC methods)
- Spectral analysis based on AR modeling
- Statistics (histograms, rain-flow analysis, fatigue evaluation)
Requirements for Education Software

- Windows based handling
- Data and instrument setups encapsulation enabling capture of both data and setups in project files
- User interface in English
- Simulation of professional signal analyzers (averaging process, multispectra evaluation)
- Replaying signals using a PC sound card
- Plots based on Microsoft Graph component
- Importing data from data files, clipboard or sound cards
- Exporting graphs by Copy and Paste methods into Word documents
Main Window Arranged as a Multiple Document Interface

Program Icon

Organisers

Notepad
Main Menu & Context Menu

Main Menu

Submenu

Context Menus

Input Signal Presentation
Organizers in the Form of Tree Structures

Measurement Organiser (Root)
- Measurements
  - Individual Signals

Instrument Organiser (Root)
- Instruments
  - Input Signals
# Type of Data Sources

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binary Data arranged in Columns</strong></td>
<td>16 bits data in binary file</td>
</tr>
<tr>
<td><strong>ScopeWin binary W-Files</strong></td>
<td>Binary data generated by ScopeWin</td>
</tr>
<tr>
<td><strong>ASCII Data from Clipboard</strong></td>
<td>ASCII data from clipboard</td>
</tr>
<tr>
<td><strong>ASCII Data from Text File</strong></td>
<td>ASCII data text file (arbitrary format, UFF)</td>
</tr>
<tr>
<td><strong>Waveform File</strong></td>
<td>Wave files (8, 16 and 24 bits)</td>
</tr>
<tr>
<td><strong>Waveform Audio Input Device and IO Device</strong></td>
<td>Data recorded directly from sound cards (mono/stereo, 8/16 bits) and NI USB-6009</td>
</tr>
<tr>
<td><strong>BK 2032 Time, BK 3550, PULSE LabShop</strong></td>
<td>Binary &amp; ASCII data from BK 2034/32 and PULSE signal analyzers</td>
</tr>
<tr>
<td><strong>Signal Generator</strong></td>
<td>Data generated manually or automatically</td>
</tr>
</tbody>
</table>
Signal Generator

Sum of components consisting of
- up to 4 harmonic signals differing in frequency, amplitude and initial phase (amplitude and phase of 2 of them can be modulated)
- and/or rectangle
- and/or swept sine
- and/or white and pink noise
FIR Filter Design

Ideal Filters
- Lowpass/Highpass FIR filter
- Lowpass differentiator FIR filter
- Lowpass Hilbert transformer FIR filter
- Bandpass FIR filter

Filters with Kaiser window
- Lowpass/Highpass FIR filter
- Lowpass FIR differentiator
- Lowpass FIR Hilbert transformer
- Bandpass FIR filter
Fast $1/f^\alpha$ Noise Generation

$\alpha = 0$, Filtered White Noise

$0 < \alpha < 2$, Filtered $1/f^\alpha$ Noise

$\alpha = 2$, Random Walk
Processing of Raw Time Histories

ZOOM, Cursor

Individual sample editing

Inserting signal segments into Measurement Organizer
BK Signal Analyzers as a Source of Time History Data

- **BK 2034/2032 ABF Files**
- **BK 3550 Time History Files**
- **LabShop PULSE Time History ASCII Data**
- **LabShop PULSE Time Capture Instrument**
- **LabShop PULSE Wave Files**

Binary files ➔ Text files & Clipboard Data

Signal Analyzer
Direct Recording Waveform Data

- **Sample Frequency**
- **Mono / Stereo**
- **Bits per Sample**
- **Insert into the tree**
- **MM Control**
- **Left Channel Data**
- **Right Channel Data**
Multifunction I/O Devices

NI USB-6009:
14-Bit, 48 kS/s Low-Cost Multifunction DAQ

NI DAQCard-6036E (for PCMCIA):
200 kS/s, 16-Bit, 16 Analog Input Multifunction DAQ

NI USB-4432
102.4 kS/s, 100 dB, 0.8 Hz AC/DC Coupled, 5-Input Sound and Vibration Device
## Signal Analyzer Instruments 1

Instruments based on the Fast Fourier Transform

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
<td>Hilbert transform, filtration in the frequency domain, amplitude and phase demodulation (full spectrum)</td>
</tr>
<tr>
<td><strong>FFT</strong></td>
<td>FFT, filtration in the frequency domain (full spectrum)</td>
</tr>
<tr>
<td><strong>CPB</strong></td>
<td>Averaged 1/1-octave and 1/3-octave frequency spectrum</td>
</tr>
<tr>
<td><strong>Autospectrum</strong></td>
<td>Averaged autospectrum (full spectrum)</td>
</tr>
<tr>
<td><strong>Cross-Spectrum</strong></td>
<td>Averaged cross-spectrum</td>
</tr>
<tr>
<td><strong>FRF`</strong></td>
<td>Averaged frequency response</td>
</tr>
</tbody>
</table>
## Signal Analyzer Instruments 2

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overall</strong></td>
<td>Overall level of the high passed signal in RMS or PWR</td>
</tr>
<tr>
<td><strong>Tachometer</strong></td>
<td>RPM evaluated from an impulse tachosignal</td>
</tr>
<tr>
<td><strong>Resampling</strong></td>
<td>Resampling of signal</td>
</tr>
<tr>
<td><strong>2nd Resampling</strong></td>
<td>2nd Resampling of signal and time delay</td>
</tr>
<tr>
<td><strong>Correlation</strong></td>
<td>Auto and cross correlation</td>
</tr>
<tr>
<td><strong>FIR Filters</strong></td>
<td>Two FIR Filters, Hilbert transform, quadrature mixing</td>
</tr>
<tr>
<td><strong>FIR Filter FRF</strong></td>
<td>FIR filter FRF including filter zeros</td>
</tr>
</tbody>
</table>
## Signal Analyser Instruments 3

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vold-Kalman</strong></td>
<td>Vold-Kalman order tracking filter</td>
</tr>
<tr>
<td><strong>Kalman Filter</strong></td>
<td>Kalman Filter for Estimating a Random Constant</td>
</tr>
<tr>
<td><strong>Statistics</strong></td>
<td>Histogram, Cumulative Histogram, Empirical distribution function, Expected</td>
</tr>
<tr>
<td></td>
<td>values of normally distributed data</td>
</tr>
<tr>
<td><strong>AR Model</strong></td>
<td>Autoregressive model coefficients evaluation</td>
</tr>
<tr>
<td><strong>AR Spectrum</strong></td>
<td>Frequency spectrum based on autoregressive model including filter poles</td>
</tr>
</tbody>
</table>
## Signal Analyzer Instruments 4

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Combiner</td>
<td>Linear combiner, Prediction, AR Process</td>
</tr>
<tr>
<td>Eigenanalysis</td>
<td>Eigenanalysis of correlation matrix, Pisarenko’s method, Music pseudospectrum</td>
</tr>
<tr>
<td>Difference</td>
<td>Weighted difference of two signals</td>
</tr>
<tr>
<td>Detrend</td>
<td>Detrend of a signal</td>
</tr>
<tr>
<td>Test Unit</td>
<td>Unit is only for testing response of the 2-order system</td>
</tr>
<tr>
<td>Script</td>
<td>Programming language for signals</td>
</tr>
</tbody>
</table>
Saving Output Data

Format of output data files
- ASCII Files
- Waveform Files (16 bits)
- ME’scope spreadsheet files for Operational Deflection Shapes & Modal Analysis
- Binary files for INOVA loading stands

Files containing projects including signals and instrument setting
- ASCII Files (extension *.sga)
- Binary Files (extension *.sgb)

Export charts in the graphic format
- *.gif files
- *.jpeg files
Inserting Signals into Instrument Input

Signal Group

Context Menu

Drag & Drop

Main Menu
Plot & Data Editing
Output Plot 1

Time History: DRIVEBY: Signal

Scatter with Line Plot

3-D Surface Plot

Microsoft Graph Environment
Output Plot 2

Surface (Top View) Plot

3-D Clustered Bar Plot

Microsoft Graph Environment
Plot Export into a Word Document

Click Copy button

Select this form

Click Paste button

Here doubleclick to edit
Setup 1 for Data Processing
Setup 2 for Data Processing
Signal Processing and Machine Diagnostics Teaching Support

- Demo-projects based on both simulated signals and real signals from a research work for industry (http://homel.vsb.cz/~tum52/Projekty/)
- Help file and e-book support (76 pages in Czech)
- Textbook on signals and signal processing methods (126 pages in Czech)
- Student’s homework support
Running Autospectra

Noise excited by run-up of a 8-cylinder and 4-stoke Diesel engine
Full Multispectrum of Signal $x(t) + j \ y(t)$

Two-side spectrum for journal bearing diagnostics

Autospectrum: $X + jY$

RMS $\mu$m

Frequency [Hz]

RPM

RPM

1727
1975
2212
2406
2378
2183
1935

0.475 ord   1.0 ord   2.0 ord

60-70
50-60
40-50
30-40
20-30
10-20
0-10

-100
-90
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
90
100

1693
1935
2183
2378
2406
2212
1975
1727
Test Unit

Frequency Response Function with Resonance & Anti-resonance
CPB, Autospectra and Frequency Response Functions

Ride comfort, truck seat frequency response functions
Frequency Weighting

Sound
Lin (no weighting), A-Type, B-Type, and C-Type

Vibrations
ISO, SAE J1490

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Frequency range</th>
<th>Mezinárodní norma</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total vibration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vertical direction (V-ISO 2631-1)</td>
<td>0,5 až 80 Hz</td>
<td>ISO 2631-1 : 1997</td>
</tr>
<tr>
<td>Horizontal direction (H-ISO 2631-1)</td>
<td>0,5 až 80 Hz</td>
<td>ISO 2631-1 : 1997</td>
</tr>
<tr>
<td>Building, all directions (Bu-ISO 2631-1)</td>
<td>1 až 80 Hz</td>
<td>ISO 2631-2 : 1989</td>
</tr>
<tr>
<td>Vertical direction, motion sickness (TV-ISO 2631-1)</td>
<td>0,1 až 0,5 Hz</td>
<td>ISO 2631-1 : 1997</td>
</tr>
<tr>
<td>Z-SAE J1490</td>
<td></td>
<td>SAE J1490</td>
</tr>
<tr>
<td>X-SAE J1490</td>
<td></td>
<td>SAE J1490</td>
</tr>
<tr>
<td>Hands</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All directions</td>
<td>8 až 1000 Hz</td>
<td>ISO 5349 : 1986</td>
</tr>
</tbody>
</table>
Filtration in the Frequency Domain

Conversion the frequency modulated impulse signal to a harmonic signal
FFT Convolution

Segments \rightarrow FFT \rightarrow Filter \rightarrow IFFT

(FFT Convolution for smoothing discontinuities of consecutive records)
Averaging in the Time or Frequency Domain

Resampled signal

Synchronously averaged signal
Wavelets

Denoising

**FRF of Quadrature mirror filters**

**Approximation**

**Details**

LP filter

HP filter
Structure Vibration Analysis

Signal Analyser
Input acceleration data

ME’scope ODS (Operational Deflection Shape) software

Double integration

Export
Histories

Cab floor acceleration

Seat pan acceleration

Histogram

Normal distribution

Expected Values of Normally Distributed Data

Straight line

Non-linearity effect
Rain-flow Counting Method

Balda’s algorithm

Analysis of fatigue data
Stress-Life Fatigue Analysis

Amplitude vs. Mid Value of Cycles

Stress-Number Curve

Cumulative Damage

Zero Mid Value of Cycles

Non-Zero Mid Value of Cycles
Programming language

- Operators
  +, -, *, /, ^, ==, ~=, <, >, <=, >=
- Signals, variables
  x1, x2[n], a[n1][n2], c, del, …
- Instructions
  for, while, exit, if - else, { }
- Functions
  iif, sin, cos, tan, atn, abs, sqr, mag, angle, unwrap, ones, exp, log, avg, sum, cumsum, length, max, min, filter, …

Text box for inserting a script
VB Code versus Scripts

Example: Computation of the matrix product

Visual Basic code

```
For i = 0 To m - 1
  For j = 0 To n - 1
    S = 0
    For k = 0 To p - 1
      S = S + a(i, k) * b(k, j)
    Next k
    c(i, j) = S
  Next j
Next i
```

Matlab script

```
>> c=a*b;
```

Scripts facilitate programming
Features of Scripts

- Easy to understand and to use
- Based on C++ and Matlab, handling with signals as vectors and matrixes
- Set of built-in functions for signal processing (FFT, IFFT, Hilbert, …)
- Repeating instructions (for, while)
- Saving results as a part of projects
- Graphic output (MS Graphs)
- Code debugging (TraceOn, TraceOff, TraceIn, TraceOut)
Creating the Library of Subroutines

List of subroutines
Parameters
Statements of Script

Build-in statements and functions can be extended by subroutines, which are designed by a software user.

A subroutine call: _SubroutineName(@Ident1, @Ident2, @Ident3, ..)
An example: _norm(x, a);
Example of a Script Code

'Pocet vzorku: '; n = 1024; n; FS = get(x1,'freq'); M = 20;
index = [0;cumsum(ones(n-1))];
hanning = 1- cos(2*pi*index/n);
d = [ ];
for(i=1;i<=M;i=i+1)
{ 
    status(i); rem('poznámka');
    a = extract(x1,round(n*(i-1)/3),n);
    if(i==1) variables;
    a = a * hanning;
    b = (mag(fft(a,'real'),fft(a,'imag'))/n*2/sqr(2))^2;
    c = [sqr(2)*b[0]/2, extract(b,1,n/2-1)];
    d = [d, c]
};
e = transpose(d); pwr = avg(e,M);
rms = sqr(pwr); 'CrLf'; 'CrLf'; 'Autospektrum: '; 'CrLf';
set(rms,n/FS,'freq'); set(rms,'Hz','unit'); format(rms,,0.000')
Additional Functionality in Gear Diagnostics Resulting from Scripts

Rem('Script for polar plots');
a=input1; fi=input2/27;
r=0.3+a;
x=r*cos(fi); y=r*sin(fi);
save(x); save(y)

Rem('Averaging synchronized with hunting frequency');
x=extract(input1,0, 18*27*40*6);
x=set(x,6,'columns');
y=transpose(x); y=avg(y,6);
y=set(y,486/27,'freq');
y=set(y,40,'columns'); matrix(y^2)
Evaluation of the ARX Model using a Script

\[ x=input1; y=input2; z=[y(-1),y(-2),x,x(-1),x(-2)]; c=z\backslash y; \]
\[ a=[c[0];c[1]]; b=[c[2];c[3];c[4]]; \]
'\'Model \',
'y = a1*y(-1) + a2*y(-2) + b0*x + b1*x(-1) + b2*x(-2)';'CrLf';'CrLf';
'Coefficients';'CrLf';'a = ';a;'CrLf';'b = ';b;'CrLf';
sy=get(c,'param1')/len(y);
'Model error standard deviation ';sqr(sy);'CrLf';
zz=diag(invs(prod(tr(z),z)));
'Coefficient error standard deviations';'CrLf';sqr(sy*zz);'CrLf';
freqz(b,a,1000,'mag');

Results
Model \[ y = a1*y(-1) + a2*y(-2) + b0*x + b1*x(-1) + b2*x(-2) \]
Coefficients
\[ a = [1,9175 \quad -0,9816] \]
\[ b = [0,2436 \quad -0,4130 \quad 0,2421] \]
Model error standard deviation 0,1023
Coefficient error standard deviations
\[ [5,523E-03 \quad 5,521E-03 \quad 3,231E-03 \quad 3,483E-03 \quad 3,427E-03] \]
## Linear Algebra and Signal Processing in Scripts of the Signal Analyzer Software

<table>
<thead>
<tr>
<th>MATLAB</th>
<th>Signal Analyzer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix &amp; vector product</td>
<td>yes</td>
</tr>
<tr>
<td>Array multiplication as the element by element product</td>
<td>yes</td>
</tr>
<tr>
<td>Matrix left division of a column vector</td>
<td>yes</td>
</tr>
<tr>
<td>Matrix inversion and pseudoinversion</td>
<td>yes</td>
</tr>
<tr>
<td>Svd, chol (Cholesky factorization)</td>
<td>yes</td>
</tr>
<tr>
<td>Roots, poly, trace, eig, det, cond</td>
<td>yes</td>
</tr>
<tr>
<td>Filter, conv, deconv, fft, ifft, angel, abs, unwrap, ar, arx, ...</td>
<td>yes</td>
</tr>
<tr>
<td>Sparse matrices</td>
<td>not implemented yet</td>
</tr>
</tbody>
</table>
Research Work

- Vold-Kalman order tracking
- Sound quality
- Kalman filter for estimating a random constant
- Angular vibration
- Quadrature mixing
- Spectral analysis based on autoregressive (AR) modeling
- Pisarenko’s method
- Music pseudospectrum
Vold-Kalman Order Tracking Filter

First generation filter

Second generation filter

Filter frequency response

$Abs(H)$

$2f/f_s$
Pass-By Noise Analysis Enabling De-Dopplerisation

Sound pressure time history

RPM profile

Doppler factor c/(c-v)

Noise level of RPM orders
Sound Quality Analysis

Filtered signal synthesis

Vold-Kalman : vyfuk_L_mono.wav 2 : Signal

-25000 -20000 -15000 -10000 -5000 0 5000 10000 15000 20000 25000

Time [s]

2 ord 4 ord 6 ord 8 ord
Hilbert Transform & Phase Demodulation

Transmission Error Measurements
Quadrature Mixing

Phase demodulation of amplitude and phase modulated harmonic signals

Modulated harmonic signal

Modulation signal

\[ s(t) \rightarrow \times \rightarrow \text{LPF} \rightarrow x_{\text{Real}}(t) \]
\[ -j \sin(\omega_c t) \]
\[ \cos(\omega_c t) \]
\[ \times \rightarrow \text{LPF} \rightarrow x_{\text{Imag}}(t) \]

Time History : Generator 1 : Sine1/PM - 1

FIR Filters : Generator 1 : Sine1/PM - 1
Kalman Filter for Estimating a Random Constant

ADC wideband noise due to the thermal effect

Time History: data1: Col 1

Filter output

Grounded-input histogram

Statistics: data1: Col 1

Cumulative difference

\[ \sum_{k=0}^{i} (s_k - y_k) \]

Difference 1: data1: Kalman Filter (Col 1)
AR Modeling in Spectral Analysis

AR: \( y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ldots + a_M y_{t-M} + e_t \) \( \Rightarrow \) \( PSD(f) = \sigma^2 T \left| 1 - \sum_{m=1}^{M} a_m \exp(-j2\pi mf) \right|^2 \)

Decaying signal

500-order AR model coefficients

FFT Autospectrum

AR Spectrum
Pisarenko’s and MUSIC Methods

100-sample record of a signal

Pisarenko’s method

FFT Autospectrum

MUSIC method

Sine1: 4 Hz, Amplitude 1

Sine2: 20 Hz, Amplitude 0.5
Conclusion

- The paper describes software that supports signal processing education at the Faculty of Mechanical engineering of the VŠB Technical University of Ostrava.

- The signal processing lectures are extended by a set of exercises based on measurements performed as a result of the research work for industry.

- Students that are working with Signal Analyzer can analyze imported measurement data or make their own noise measurements using a sound card as the first step to become experts.