Developers Guide

Brief Description of all Functions

plot_pbu_figure

Select a Well Data folder to plot the figures (pressure change and derivative) only.

Usage: plot_pbu_figure.m

Input:

Via user prompt select a Well data folder to plot figures of PBU data.

Output:

Produces figure plots and saves them in pdf.

Other functions required:

remove_nan

interpolate_data

get_limit

Please see the “PBU Figure plot.pdf “ file for elaborate code description.

single_sample_threshold_selection

Select a Well Data and a PBU to empirically select a Deviation Threshold value for that well

Usage: single_sample_threshold_selection.m

Input:

Via user prompt select a Well data folder and one PBU of that well.

Output:

Produces Noise labeled plots for variable Deviation Threshold values for the selected PBU.

Other functions required:

remove_nan

interpolate_data

get_limit

ssa_decomposition

interp_to_original_mapping_single_point

original_to_interp_mapping_start_end_point

Please see the “Single Sample Threshold Selection.pdf “ file for elaborate code description.

automatic_noise_detection

Select a Well Data and set Deviation Threshold value from the single sample learning.

Usage: automatic_noise_detection.m

Input:

Via user prompt select a Well data folder and set the Deviation

Threshold value from single sample learning.

Output:

Produces Noise labeled plots for all the PBU of the well.

Other functions required:

remove_nan

interpolate_data

get_limit

ssa_decomposition

interp_to_original_mapping_single_point

original_to_interp_mapping_start_end_point

Please see the “Automatic Structured Noise Detection.pdf “ file for elaborate code description.

remove_nan

Removes the data points where derivative value is -999.

Usage:

val_mod=remove_nan(val)

Input Arguments:

val A matrix of size 2 by 'PBU-Time Series length'. First row contains Time axis value, Second row contains Signal Derivative value

Output value:

val_mod: Data points that has derivative value = -999, are removed from time and derivative. A matrix of size 2 by 'PBU-Time Series length'. First row contains Time axis value, Second row contains Derivative value.

interpolate_data

Performs linear interpolation on the NaN removed Data to make the data uniformly sampled using given sampling rate

Usage: val_mod = interpolate_data (val,sampling_rate)

Input Arguments:

val A matrix of size 2 by 'PBU-Time Series length'. First row contains Time axis value, Second row contains Derivative value vector.

sampling_rate The time axis value difference between two consecutive data points.

Output value:

val_mod: A matrix of size 2 by 'PBU-Time Series length'. Time axis values has been linearly interpolated using given sampling rate. First row contains Time axis value, Second row contains Derivative value.

get_limit

Calculates the minimum, maximum of time axis value and derivative value for plot axis limit.

Usage:

[xlv, ylv] = get_limit (time, derv)

Input Arguments:

time Time Axis Values.

derv Derivative values

Output value:

xlv: A 1 by 2 vector. [minimum x-axis(time) value, maximum x-axis(time) value]

ylv: A 1 by 2 vector. [minimum y-axis(time) value, maximum y-axis (derivative) value]

ssa_decomposition

Performs SSA decomposition of a given time series

Usage: [ssa_rc]= ssa_decomposition (X, window)

Input Arguments:

X Time series to be decomposed in row vector

window Lag window for building Co-variance Matrix.

Output value:

ssa_rc: The first five SSA reconstructed components in a five by length of (X) matrix. First row of ssa_rc contains first RC, second row contains second RC and so on.

interp_to_original_mapping_start_end_point

Maps interpolated time values to the original data points and returns the corresponding indexes

Usage:

```
[start_ind, end_ind] =
interp_to_original_mapping_start_end_point
        (start_time,end_time,time)
```

Input Arguments:

start_time starting time of a segment from the interpolated values.

end_time ending time of a segment from the interpolated values.

time original time values.

Output value:

start_ind: Index of the matched/mapped original time value of start_time.

end_ind: Index of the matched/mapped original time value of end_time.

original_to_interp_mapping_single_point

Matches/maps a original time axis value with the nearest/closest (based on time axis value) interpolated time axis value and returns the corresponding index.

Usage: [ind] = original_to_interp_mapping_single_point (point_time,time)

Input Arguments:

point_time A time value in the original data points (scaler)

time Interpolated time axis values (vector)

Output value:

ind: Index of the matched/mapped interpolated time value.