VTool

A “Validation Toolbox”

by Pierino Bonanni

Why VTool?

- Efficient import/analysis/plotting of simulation and field data.
  - Extracting just what you need
  - Making desired signals easy to access
- Managing signal name variations.
- Managing diversity of sampling rates, including non-uniform sampling.
- Managing file formats, units conversions, etc.
- Performing data clean-up before re-sampling.
- Handling both simulation (i.e., elapsed) time and absolute time.
  - Synchronizing real and simulation time
  - Capturing “events” that don’t start/end at data file boundaries
- Easy concatenation and segmenting of data files.
- Documenting dataset “recipes” for future use, and efficient communication across teams.

The VTool Paradigm

Conventional Approach

- Write script for reading data.
- Write code for analyzing data, given the specific signal names and application format.
- Write customized codes for viewing, plotting, storing.

No re-use.

VTool Approach

- Plan your dataset architecture
- Assemble signals into that architecture, independent of data source
- Devise algorithm code to use the standardized I/O data structures
- Use generalized tools for plotting, analyzing, comparing, documenting, storing, etc.

Re-use utilities again and again.

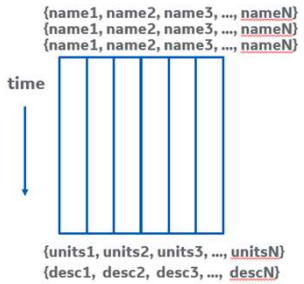
Focus on algorithms only.

The data model …

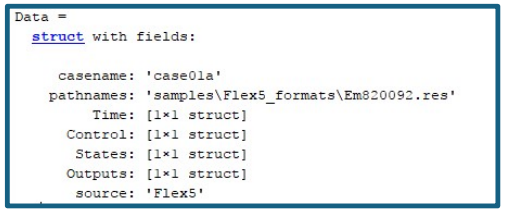
Four VTool Data Structures

1) “Signal Group”
2) “Dataset”
3) “Signal Group Array”
4) “Dataset Array”

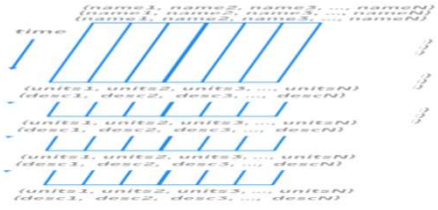
1)



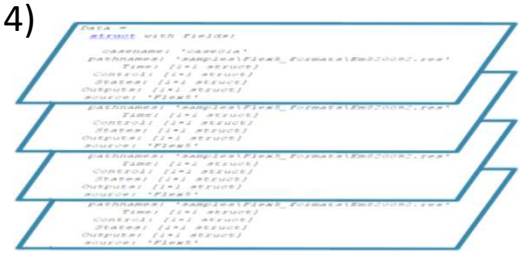
2)



3)



4)



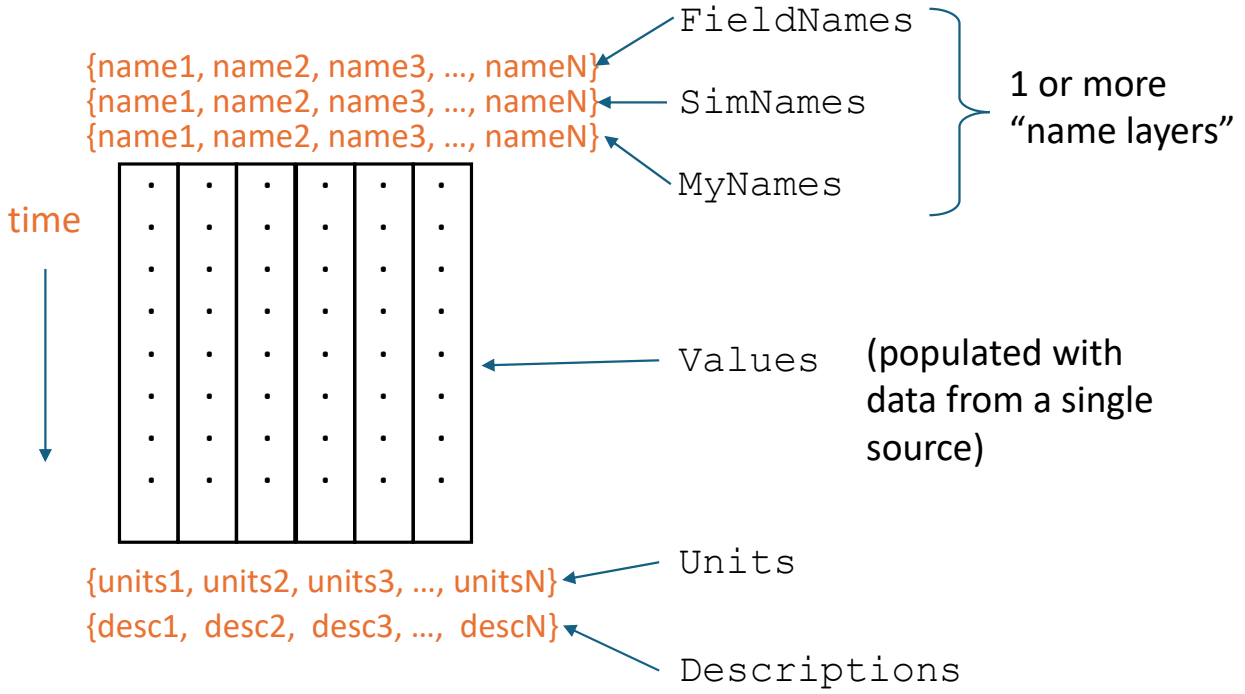
What does their use buy you?

- Annotated plotting
  - Time series
  - Spectra
  - Difference Spectra
  - Coherence plots
- Comparison
  - Equality validation
  - Anomaly detection
- Statistical Analysis
  - Static (DC) statistics
  - Frequency domain
  - Specialized (DEL, LDD, LRD, …)
  - Segmentation by category

What is a “Signal Group”

A block of time-synchronized signals, referenced by name.

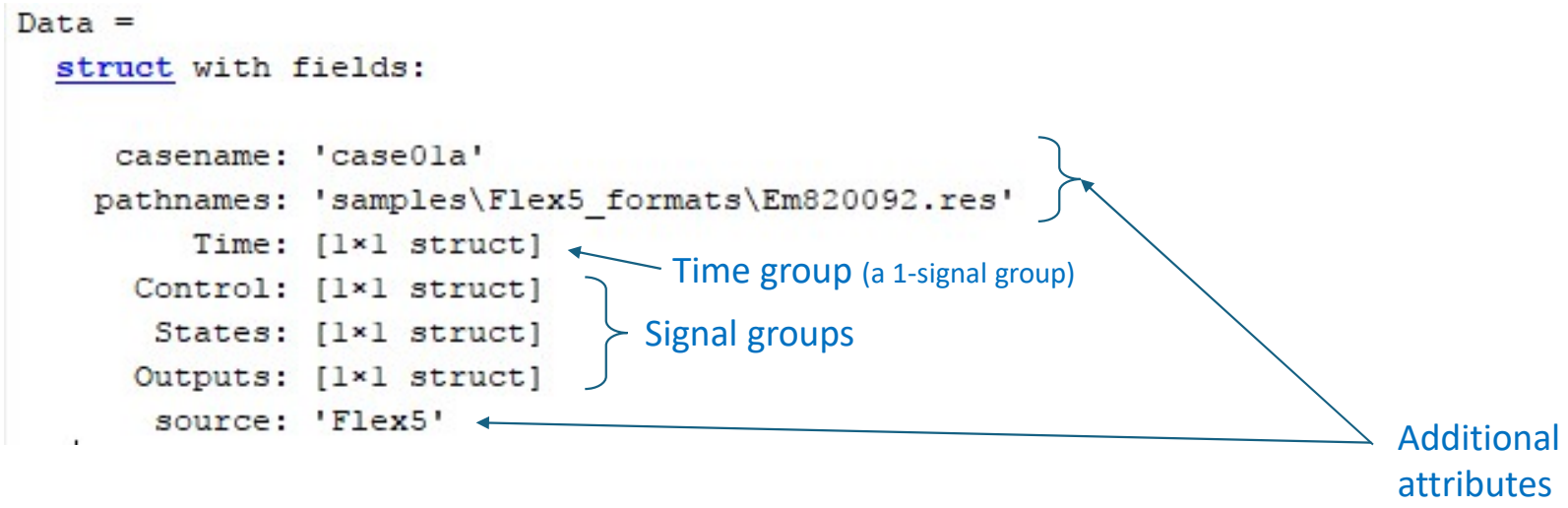
```
group1 =
  struct with fields:

      Flex5Names: {13×1 cell}
      WTLinNames: {13×1 cell}
        MBCNames: {13×1 cell}
          Values: [686×13 single]
           Units: {13×1 cell}
    Descriptions: {13×1 cell}
```



Every signal group carries a complete cross reference of signal names

What is a “Dataset”

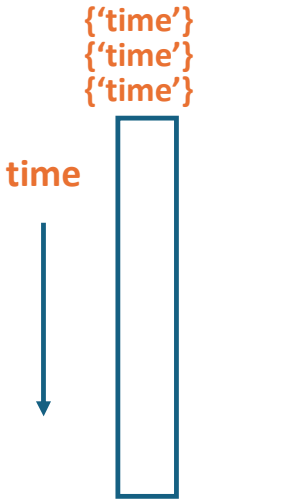
A collection of signal groups with a common Time signal, plus identifying attributes.



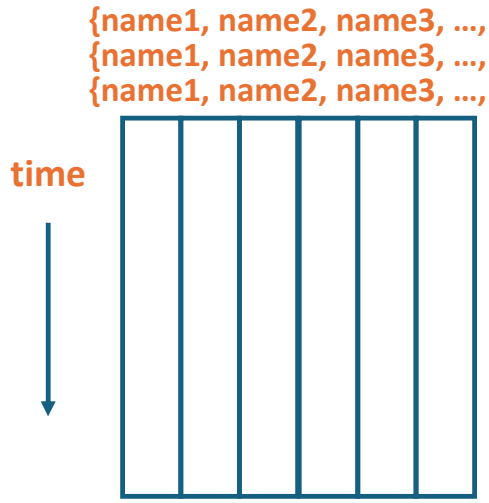
Signal Groups and Datasets can be Stacked

Going from these *scalar* structures…

a) Time Group



b) Signal Group

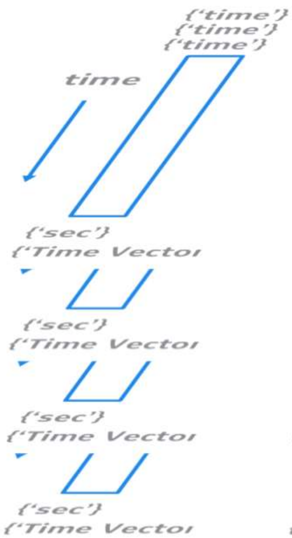


c) Dataset

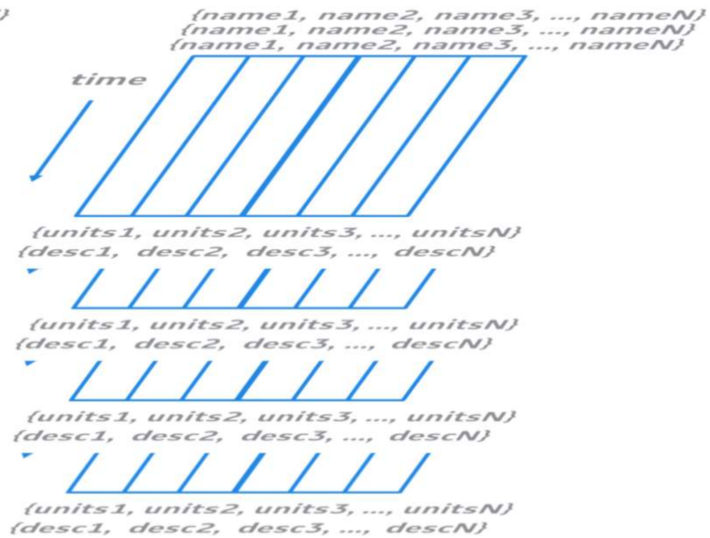
```
Data =
  struct with fields:

      casename: 'case01a'
     pathnames: 'samples\Flex5_formats\Em820092.res'
          Time: [1×1 struct]
       Control: [1×1 struct]
        States: [1×1 struct]
       Outputs: [1×1 struct]
        source: 'Flex5'
```

…to these *structure arrays*.

a) Time Array
TIMES(i)

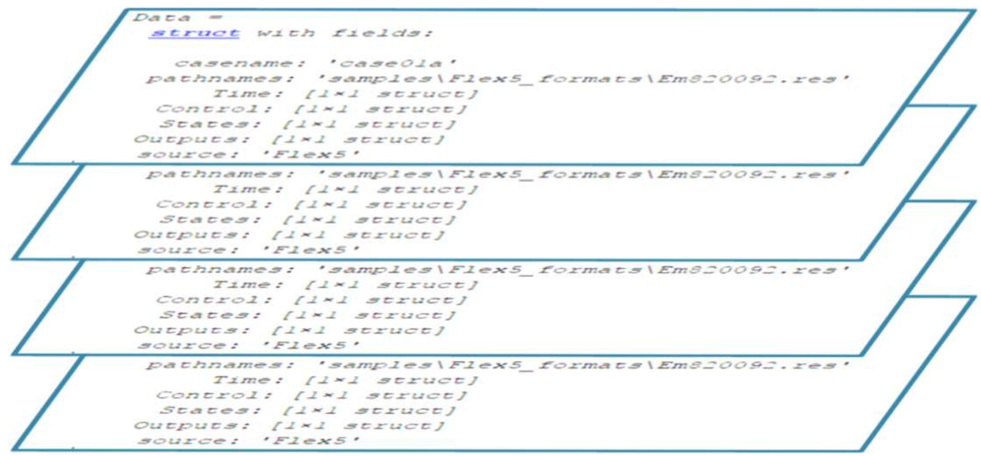


b) Signal Group Array
SIGNALS(i)



Models analogous signals from different sources (e.g., for statistical analysis, coherence calculation, plotting, etc.)

c) Dataset Array
DATA(i)



Stores comparable signal sets from different sources.

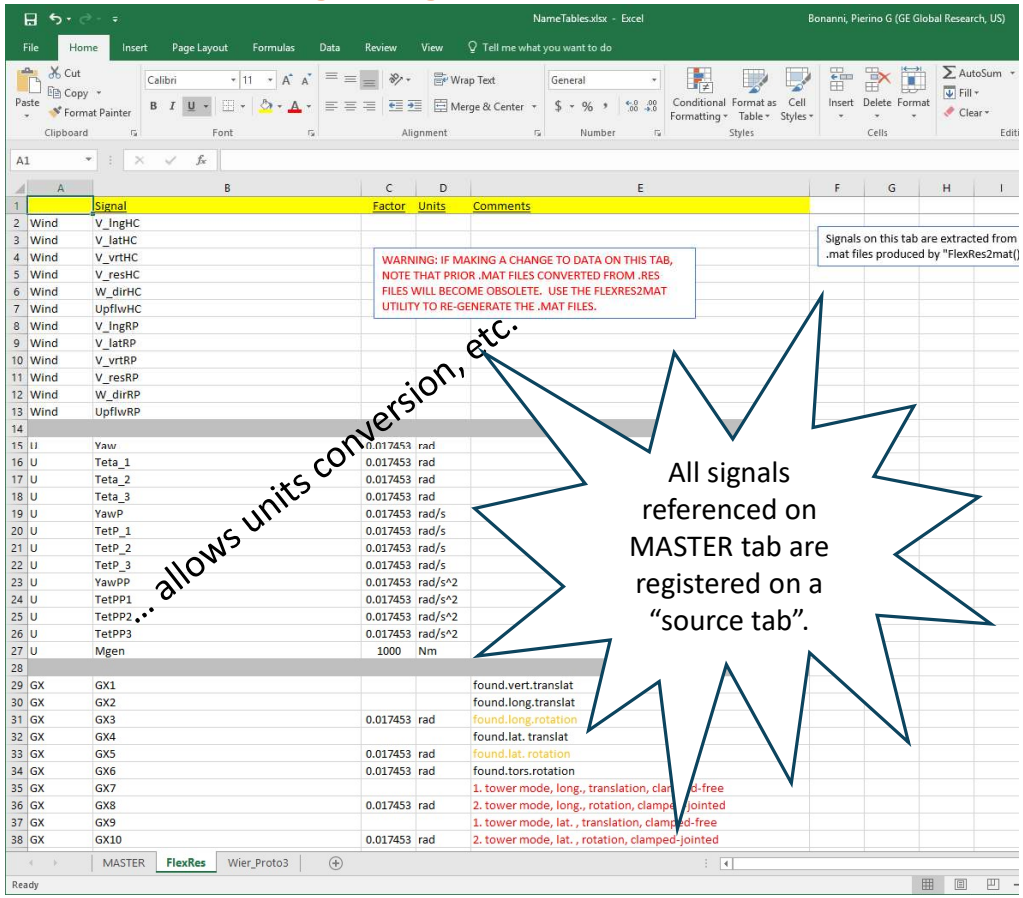
Organizes results from different test cases.

More …

Signal Name Management

NameTables.xlsx

Source tab: signal registration



	A	Signal	Factor	Units	Comments
2	Wind	V_lngHC			
3	Wind	V_latHC			
4	Wind	V_vrtHC			
5	Wind	V_resHC			
6	Wind	W_dirHC			
7	Wind	UpflwHC			
8	Wind	V_lngRP			
9	Wind	V_latRP			
10	Wind	V_vrtRP			
11	Wind	V_resRP			
12	Wind	W_dirRP			
13	Wind	UpflwRP			
14					
15	U	Yaw	0.017453	rad	
16	U	Teta_1	0.017453	rad	
17	U	Teta_2	0.017453	rad	
18	U	Teta_3	0.017453	rad	
19	U	YawP	0.017453	rad/s	
20	U	TetP_1	0.017453	rad/s	
21	U	TetP_2	0.017453	rad/s	
22	U	TetP_3	0.017453	rad/s	
23	U	YawPP	0.017453	rad/s^2	
24	U	TetPP1	0.017453	rad/s^2	
25	U	TetPP2	0.017453	rad/s^2	
26	U	TetPP3	0.017453	rad/s^2	
27	U	Mgen	1000	Nm	
28					
29	GX	GX1			found.vert.translat
30	GX	GX2			found.long.translat
31	GX	GX3	0.017453	rad	found.long.rotation
32	GX	GX4			found.lat. translat
33	GX	GX5	0.017453	rad	found.lat. rotation
34	GX	GX6	0.017453	rad	found.tors.rotation
35	GX	GX7			1. tower mode, long., translation, clamped-free
36	GX	GX8	0.017453	rad	2. tower mode, long., rotation, clamped-jointed
37	GX	GX9			1. tower mode, lat. , translation, clamped-free
38	GX	GX10	0.017453	rad	2. tower mode, lat. , rotation, clamped-jointed

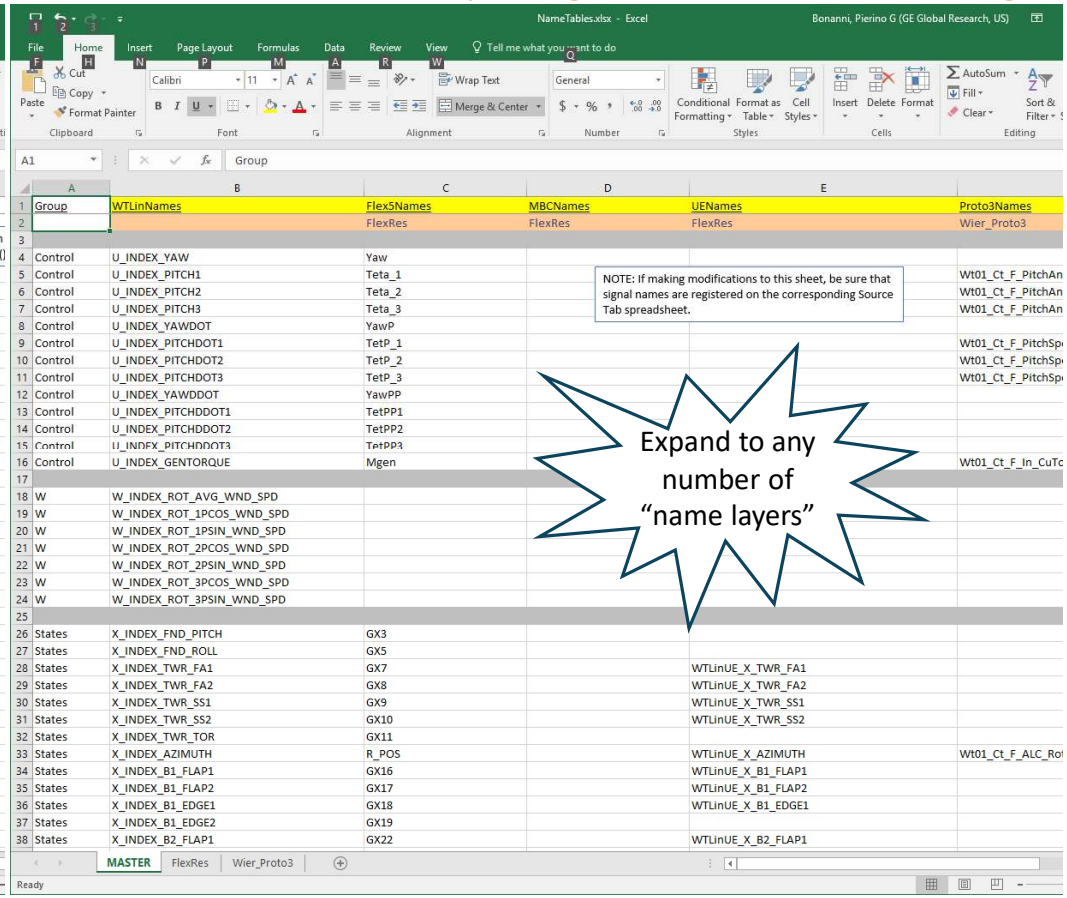
WARNING: IF MAKING A CHANGE TO DATA ON THIS TAB, NOTE THAT PRIOR .MAT FILES CONVERTED FROM .RES FILES WILL BECOME OBSOLETE. USE THE FLEXRES2MAT UTILITY TO RE-GENERATE THE .MAT FILES.

Signals on this tab are extracted from .mat files produced by "FlexRes2mat()

... allows units conversion, etc.

All signals referenced on MASTER tab are registered on a "source tab".

MASTER tab: dataset recipes, signal-name cross referencing



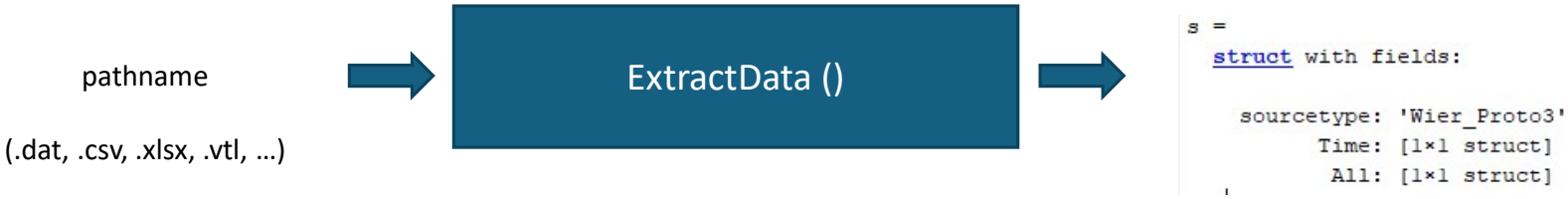
	A	Group	B WTLinNames	C Flex5Names	D MBCNames	E UENames	Proto3Names
1				FlexRes	FlexRes	FlexRes	Wier_Proto3
2							
3							
4	Control		U_INDEX_YAW	Yaw			
5	Control		U_INDEX_PITCH1	Teta_1			Wt01_Ct_F_PitchAn
6	Control		U_INDEX_PITCH2	Teta_2			Wt01_Ct_F_PitchAn
7	Control		U_INDEX_PITCH3	Teta_3			Wt01_Ct_F_PitchAn
8	Control		U_INDEX_YAWDOT	YawP			
9	Control		U_INDEX_PITCHDOT1	TetP_1			Wt01_Ct_F_PitchSp
10	Control		U_INDEX_PITCHDOT2	TetP_2			Wt01_Ct_F_PitchSp
11	Control		U_INDEX_PITCHDOT3	TetP_3			Wt01_Ct_F_PitchSp
12	Control		U_INDEX_YAWDDOT	YawPP			
13	Control		U_INDEX_PITCHDDOT1	TetPP1			
14	Control		U_INDEX_PITCHDDOT2	TetPP2			
15	Control		U_INDEX_PITCHDDOT3	TetPP3			
16	Control		U_INDEX_GENTORQUE	Mgen			Wt01_Ct_F_In_CuTo
17							
18	W		W_INDEX_ROT_AVG_WND_SPD				
19	W		W_INDEX_ROT_1PCOS_WND_SPD				
20	W		W_INDEX_ROT_1PSIN_WND_SPD				
21	W		W_INDEX_ROT_2PCOS_WND_SPD				
22	W		W_INDEX_ROT_2PSIN_WND_SPD				
23	W		W_INDEX_ROT_3PCOS_WND_SPD				
24	W		W_INDEX_ROT_3PSIN_WND_SPD				
25							
26	States		X_INDEX_FND_PITCH	GX3			
27	States		X_INDEX_FND_ROLL	GX5			
28	States		X_INDEX_TWR_FA1	GX7		WTLinUE_X_TWR_FA1	
29	States		X_INDEX_TWR_FA2	GX8		WTLinUE_X_TWR_FA2	
30	States		X_INDEX_TWR_SS1	GX9		WTLinUE_X_TWR_SS1	
31	States		X_INDEX_TWR_SS2	GX10		WTLinUE_X_TWR_SS2	
32	States		X_INDEX_TWR_TOR	GX11			
33	States		X_INDEX_AZIMUTH	R_POS		WTLinUE_X_AZIMUTH	Wt01_Ct_F_ALC_Ro
34	States		X_INDEX_B1_FLAP1	GX16		WTLinUE_X_B1_FLAP1	
35	States		X_INDEX_B1_FLAP2	GX17		WTLinUE_X_B1_FLAP2	
36	States		X_INDEX_B1_EDGE1	GX18		WTLinUE_X_B1_EDGE1	
37	States		X_INDEX_B1_EDGE2	GX19			
38	States		X_INDEX_B2_FLAP1	GX22		WTLinUE_X_B2_FLAP1	

NOTE: If making modifications to this sheet, be sure that signal names are registered on the corresponding Source Tab spreadsheet.

Expand to any number of "name layers"

An under-the-hood

Unified data input function



Read data from any source, independent of format

Currently

```
filetype(*)   requirements(*)
-----------   ---------------
 'vtool'      file extension '.vtl'
 'S-array'    root name beginning with 'S_', file extension .mat
 'xls'        file extension matching .xls or .xlsx
 'csv'        file extension matching .csv
 '.XXX'       file extension matching the characters 'XXX' following
              the '.' character.
(*) The 'filetype' input and file extensions are not case-sensitive.

Type "help formats" for more information on supported file formats.
```

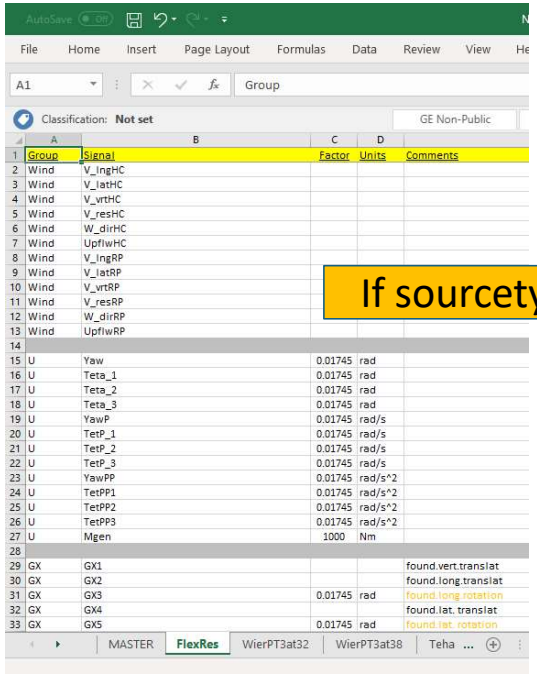
Same function, specifying “sourcetype”

pathname

sourcetype (‘FlexRes’, ‘Lubbock’, ‘Javelina’ …)

ExtractData (…,sourcetype)

```
s =
  struct with fields:

    sourcetype: 'FlexRes'
          Time: [1×1 struct]
       GFEAero: [1×1 struct]
       GFVAero: [1×1 struct]
            GX: [1×1 struct]
           GXP: [1×1 struct]
           MBC: [1×1 struct]
             U: [1×1 struct]
            UE: [1×1 struct]
          Wind: [1×1 struct]
             Y: [1×1 struct]
```



	A	B	C	D	E
1	Group	Signal	Factor	Units	Comments
2	Wind	V_IngHC			
3	Wind	V_latHC			
4	Wind	V_vrtHC			
5	Wind	V_resHC			
6	Wind	W_dirHC			
7	Wind	UpfIwHC			
8	Wind	V_IngRP			
9	Wind	V_latRP			
10	Wind	V_vrtRP			
11	Wind	V_resRP			
12	Wind	W_dirRP			
13	Wind	UpfIwRP			
14					
15	U	Yaw	0.01745	rad	
16	U	Teta_1	0.01745	rad	
17	U	Teta_2	0.01745	rad	
18	U	Teta_3	0.01745	rad	
19	U	YawP	0.01745	rad/s	
20	U	TetP_1	0.01745	rad/s	
21	U	TetP_2	0.01745	rad/s	
22	U	TetP_3	0.01745	rad/s	
23	U	YawPP	0.01745	rad/s^2	
24	U	TetPP1	0.01745	rad/s^2	
25	U	TetPP2	0.01745	rad/s^2	
26	U	TetPP3	0.01745	rad/s^2	
27	U	Mgen	1000	Nm	
28					
29	GX	GX1			found.vert.translat
30	GX	GX2			found.long.translat
31	GX	GX3	0.01745	rad	found.long.rotation
32	GX	GX4			found.lat. translat
33	GX	GX5	0.01745	rad	found.lat. rotation

If sourcetype specified, arrange/modify as specified in NameTables.

Mass Conversion

Rawdata Extractors

.dat, .res, .csv, … files

(any recognized input format)

ConvertToVtl(sourcetype,…)

bulk conversion utility

VTool .vtl files

(signal groups, one time vector)

Use .vtl files for speed efficiency.

.dat, .res, .csv, … files

(any recognized input format)

ConvertToSarray(…)

bulk conversion utility

“S array” .mat files

S(i).name
S(i).data
etc.

Use S_*.mat files for space efficiency.

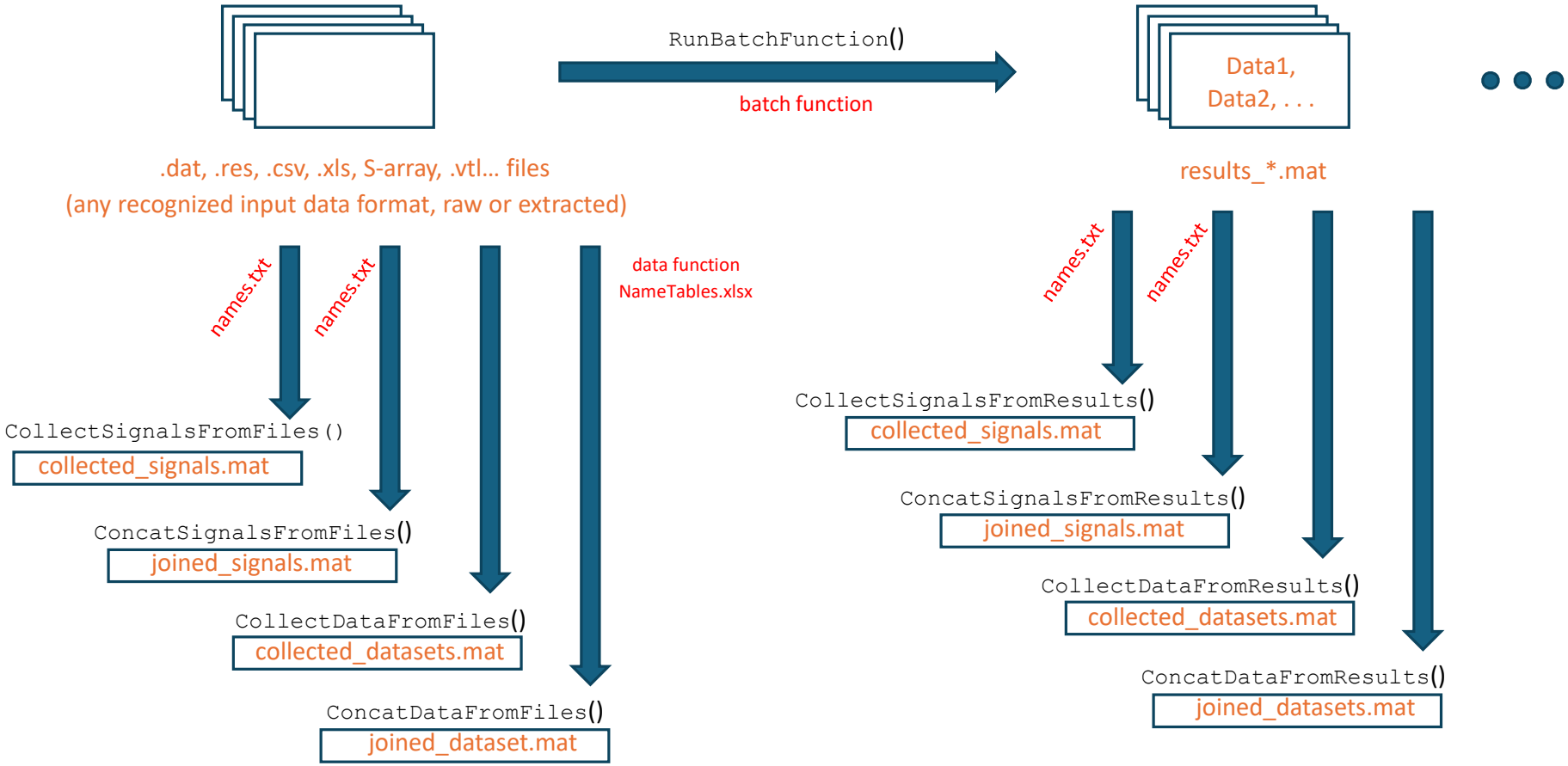
These operations also support:

- Units conversions
- Signal description changes
- Re-sampling

VTool Workflow

Files used: names.txt, include.txt, exclude.txt

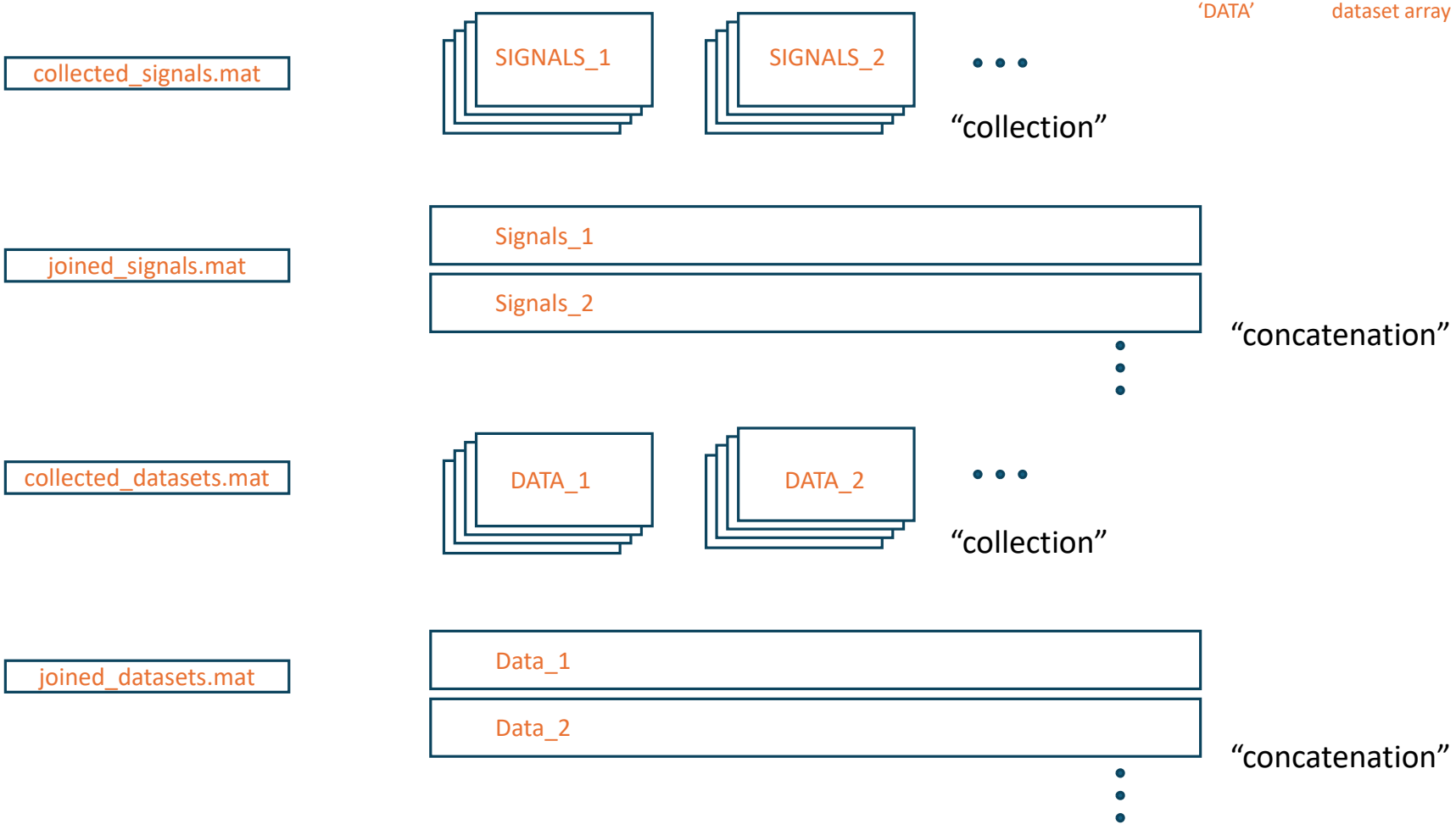
data function: pathname in, dataset out
batch function: pathname in, results_xxx.mat out



Mat-file Contents

NAMING CONVENTION

'Signals'	signal group
'SIGNALS'	signal group array
'Data'	dataset
'DATA'	dataset array



Plotting features …

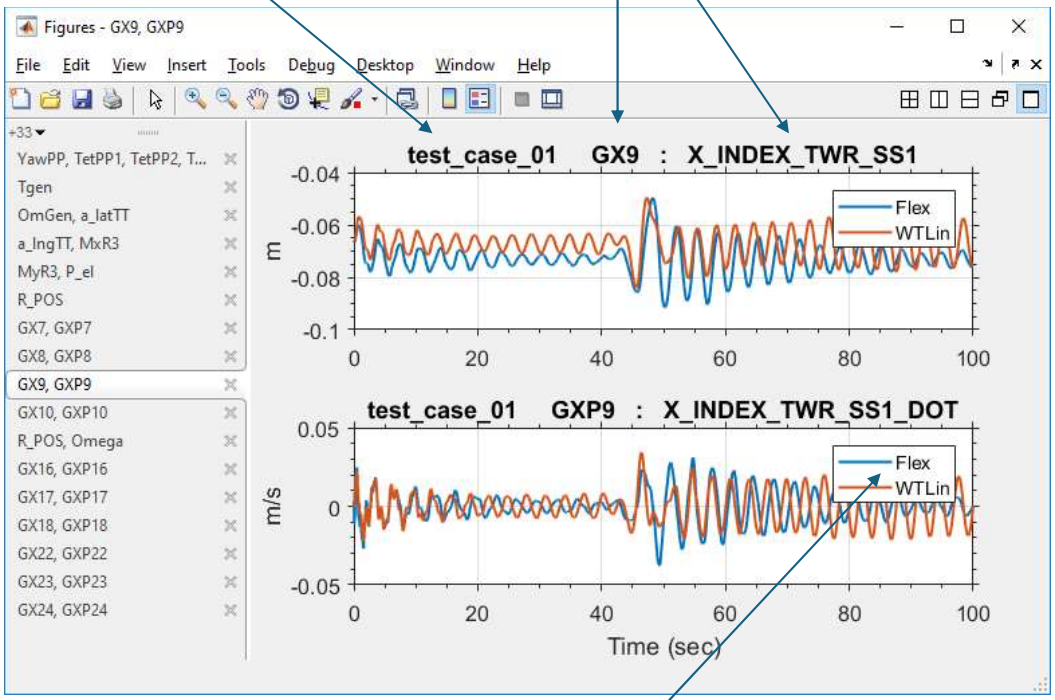
Plotting Time Series

```
>> PlotDataset(resultsFlex,resultsWTLin)
```

resultsFlex
resultsWTLin

From “casename” field

From ‘Flex5Names’
and ‘WTLinNames’

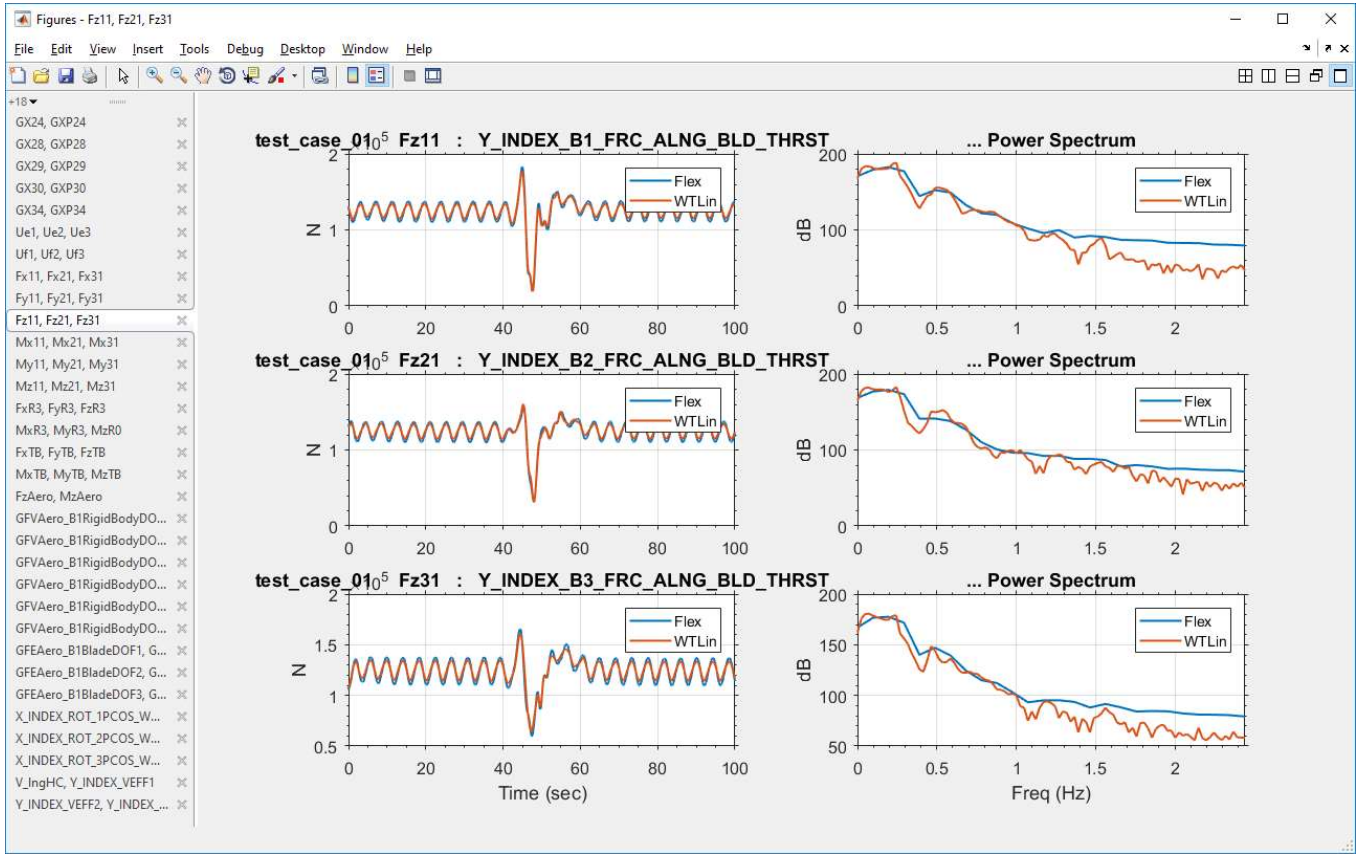


From ‘source’ fields

… Power Spectra

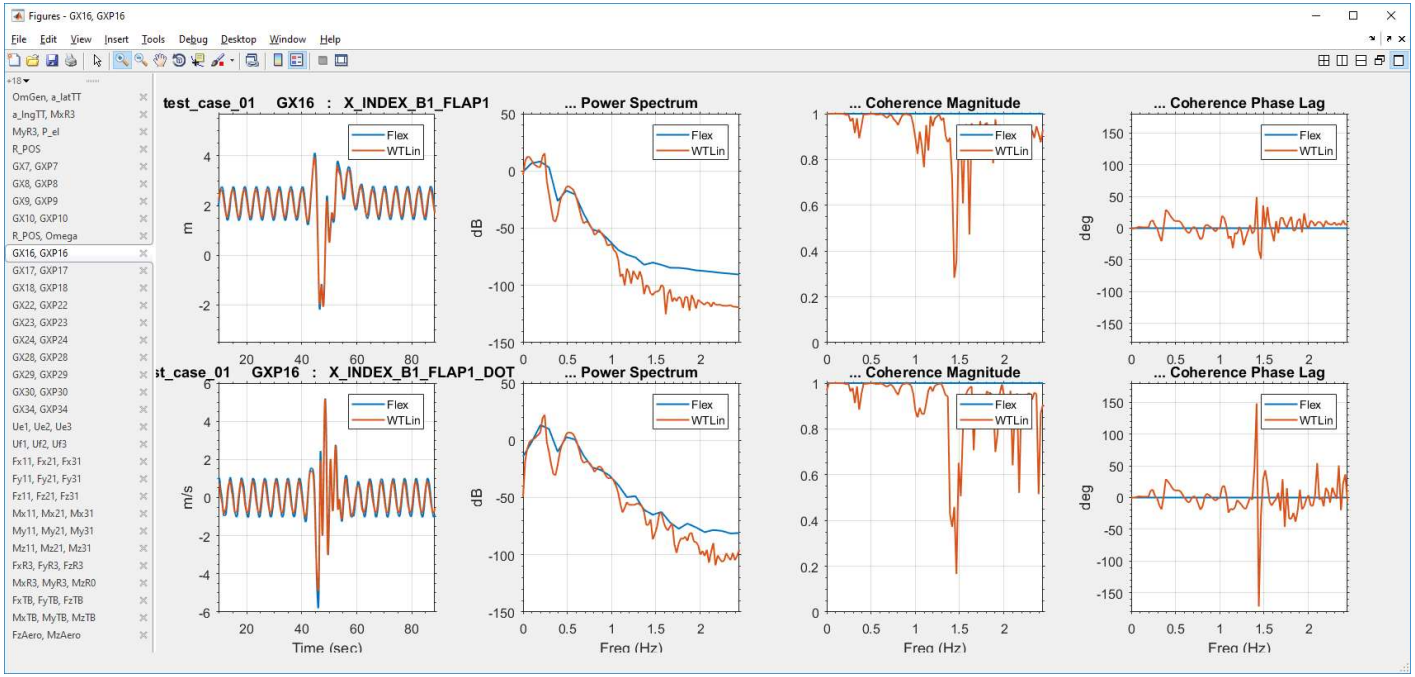
```
>> PlotDataset(DataIn,DataOut,'psd')
```

Full list of plots



… PSDs, Coherences, or both

```
>> PlotDataset(DataIn,DataOut,'psd','coh')
```



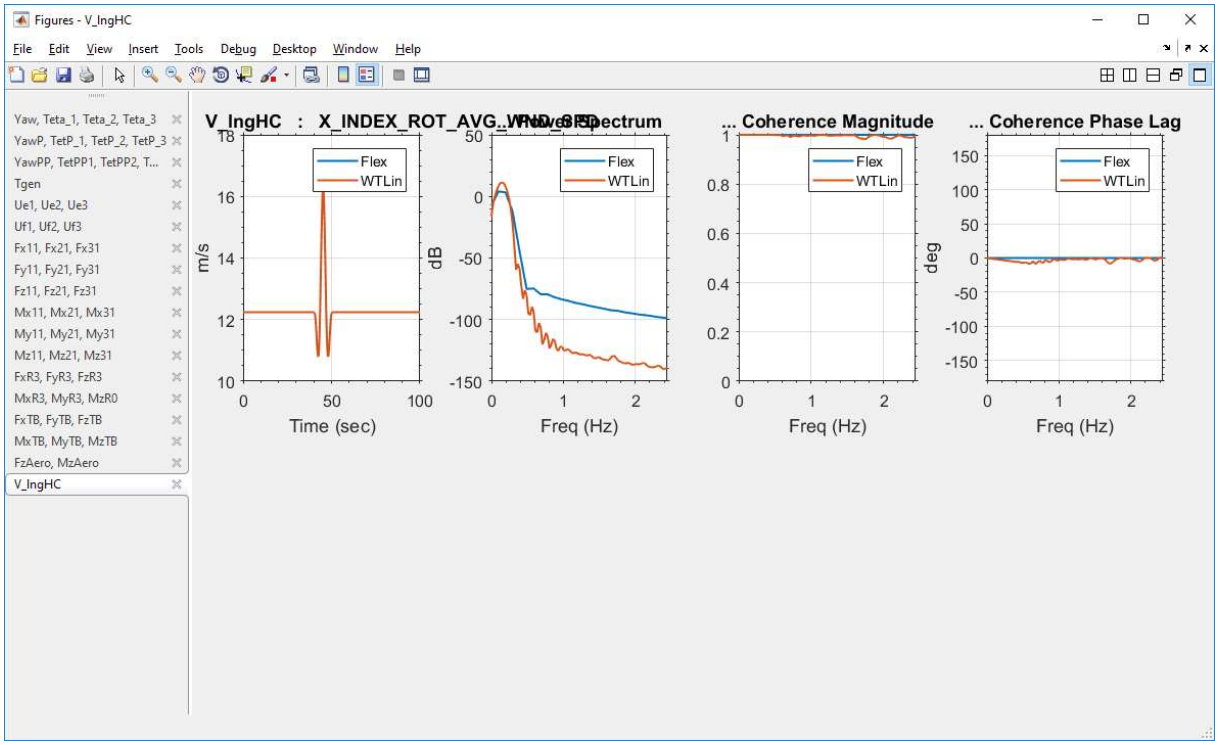
Time axes linked -
within and across all windows.

Frequency axes linked -
within and across all windows.

Can pan/zoom time and frequency independently.

… Specifying signals and groups

```
>> PlotDataset(DataIn,DataOut,'psd','coh',{'Control','Loads','V_lngHC', …})
```

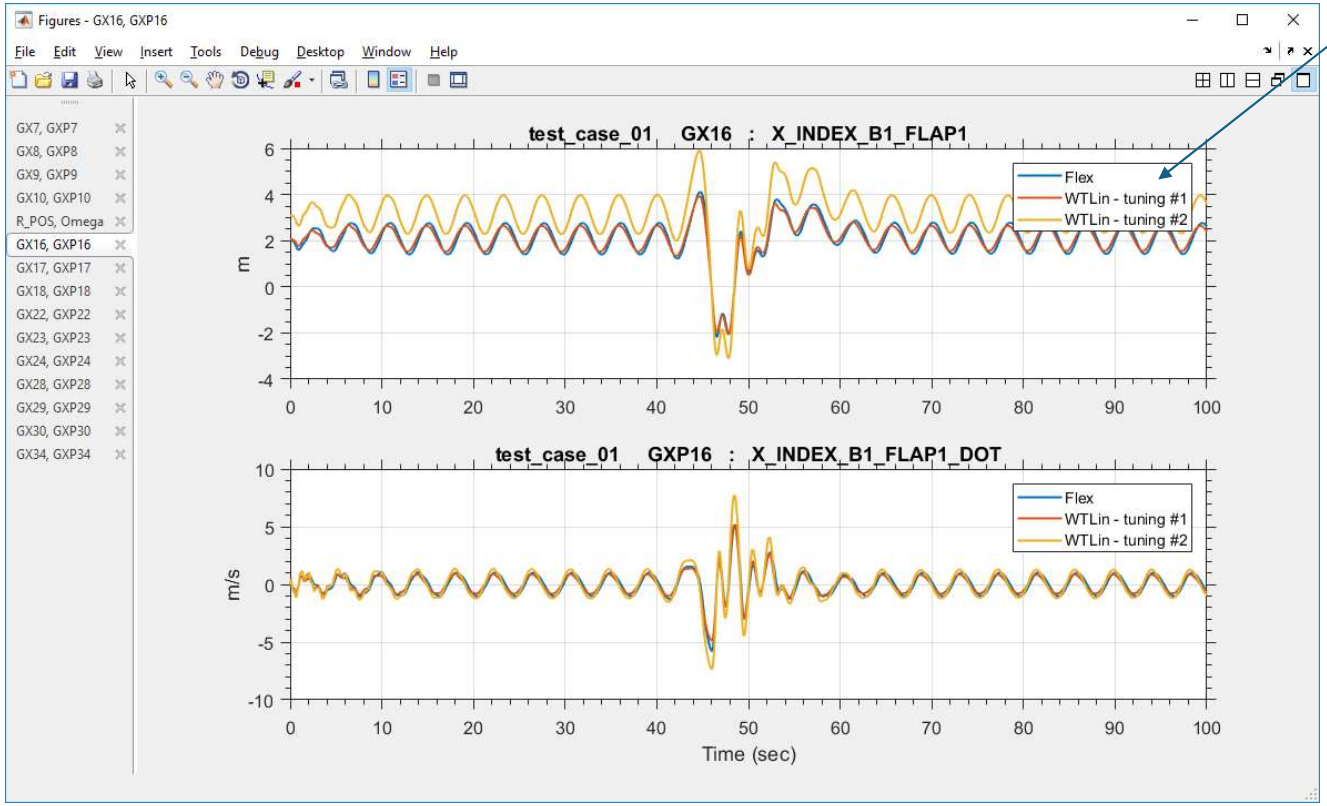


Specify either the Flex5 or WTLin name.

In this example, we requested the “Control” group, the “Loads” group, plus a random signal (‘V_lngHC’)

… Any number of input datasets

```
>> PlotDataset(DataIn,DataOut1,DataOut2,...)
```

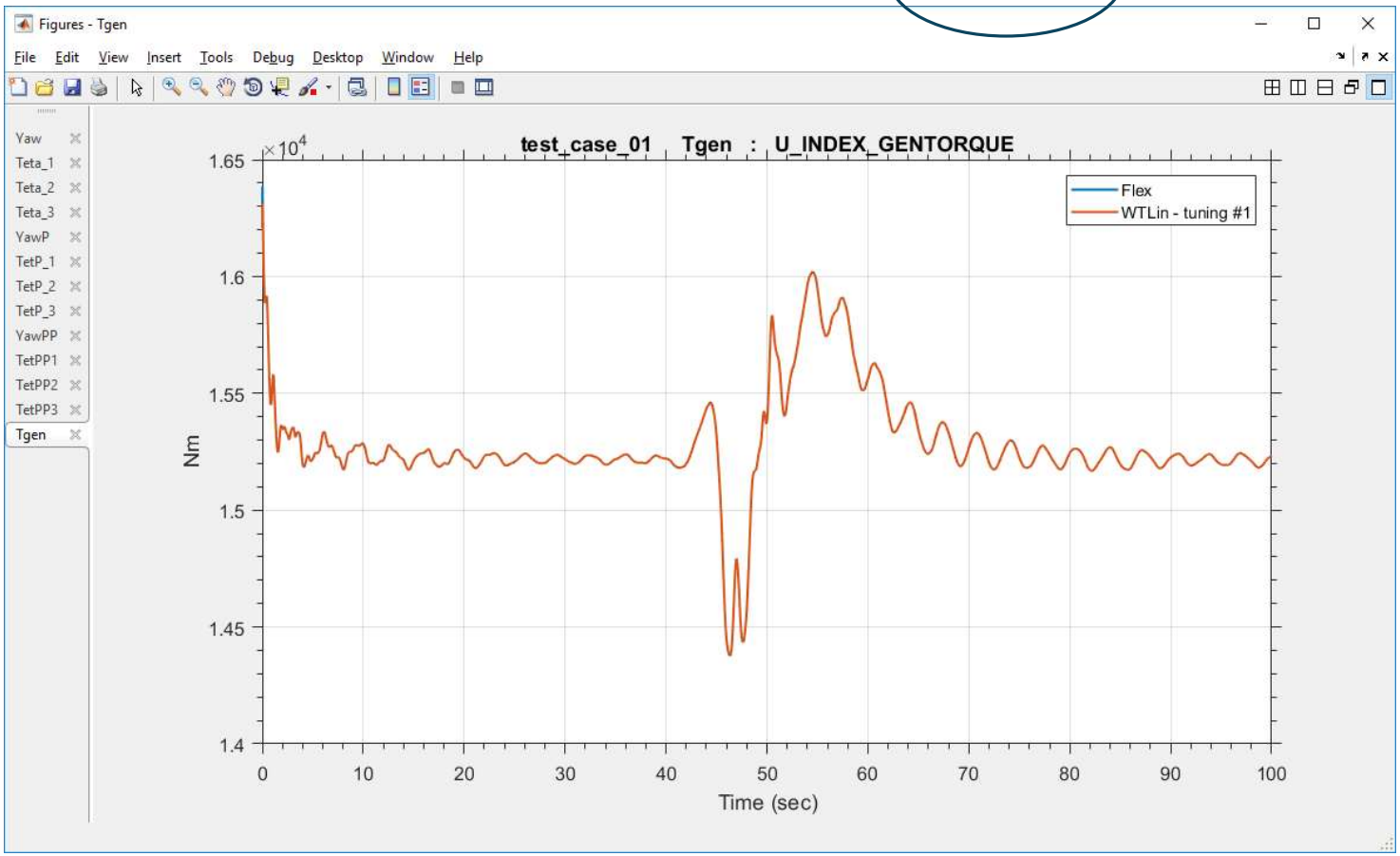


DataIn.source = ‘Flex’
DataOut1.source = ‘WTLin – tuning #1’
DataOut2.source = ‘WTLin – tuning #2’

… Full-window plots

```
>> PlotDataset(DataIn,DataOut1,DataOut2,'single')
```

Only one plot
per plot window



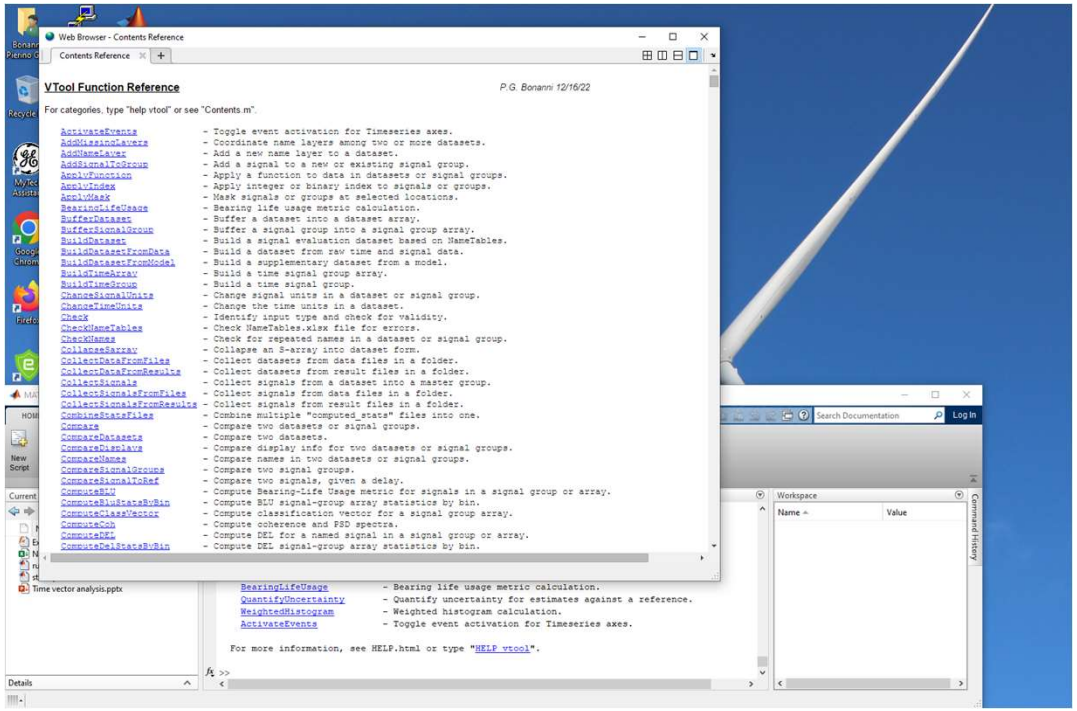
Getting help …

Help Utilities

Categorized List

```
>> help VTool
  VTool              VALIDATION TOOLBOX              Pierino G. Bonanni 7/20/19

  Dataset Construction and Modification
    BuildDataset              - Build a signal evaluation dataset based on NameTables.
    BuildDatasetFromData      - Build a dataset from raw time and signal data.
    BuildDatasetFromModel     - Build a supplementary dataset from a model.
    RebuildDataset            - Rebuild a dataset based on NameTables.
    RebuildDatasetFromModel   - Rebuild a dataset based on a model.
    NullDataset               - Build a null dataset from a model.
    RenameField               - Rename signal groups or fields in a dataset.
    CopySignals               - Copy signals between datasets or signal groups.
    ChangeSignalUnits         - Change signal units in a dataset or signal group.
    ChangeTimeUnits           - Change the time units in a dataset.
    ReconcileUnits            - Reconcile units across datasets or signal groups.
    ReplaceSignalInDataset    - Replace a signal in a dataset.
    ReplaceUnits              - Replace a units string in a dataset or signal group.
    ReplaceDescription        - Replace a description string in a dataset or signal group.
    RemoveGroupsExcept        - Remove extra signal groups from a dataset.
    LimitTimeRange            - Limit time range of a dataset.
    ResampleDataset           - Resample a dataset.
    DownsampleDataset         - Downsample a dataset.
    SampleAndHold             - Sample and hold signals in a dataset or signal group.
    RemoveRepeatedPoints      - Remove repeated time points from a dataset.
    Decimate                  - Decimate signals or groups in a dataset.
    ApplyMask                 - Mask signals or groups at selected locations.
    ApplyIndex                - Apply integer or binary index to signals or groups.
    ApplyFunction             - Apply a function to data in datasets or signal groups.
    BufferDataset             - Buffer a dataset into a dataset array.
    ConcatDatasets            - Concatenate datasets into a single dataset.
    SequenceDatasets          - Sequence datasets contiguously in time.
    MergeDatasets             - Merge two or more datasets into one.
    ConvertToDouble           - Convert data to 'double'.
    ConvertToAbsoluteTime     - Convert to absolute time.
    ConvertToElapsedTime      - Convert to elapsed time.
    SelectFromDataset         - Reduce a dataset to a set of selected signals.
    RegroupByDimension        - Regroup a dataset by signal dimension.

  Signal Group Manipulation
    BuildTimeGroup            - Build a time signal group.
    BuildTimeArray            - Build a time signal group array.
    DefineSignalGroup         - Define a new signal group on a dataset.
    CollectSignals            - Collect signals from a dataset into a master group.
    AddSignalToGroup          - Add a signal to a new or existing signal group.
    MergeSignalGroups         - Merge two or more signal groups into one.
    SelectFromGroup           - Reduce a signal group to a set of selected signals.
    RemoveFromGroup           - Remove one or more signals from a signal group.
```

Command Reference



Topics for Live Demo

- Building structured datasets *generically* – specifying your own *signals groups* and name *layers*
- Matching field data to simulation data, and lining up comparable signals using their names
- Defining new groups (e.g., “Measurements”) based on signals already in the dataset
- Leaving a spot in the signal arrays for post-processed signals
- Reading and mass-conversion of simulation and field data into a common format, *independent of field site and original file format*
- Reading multiple field-data files (spanning a period of absolute time) and concatenating into a single contiguous dataset
- Plotting and comparing signal groups and datasets, both as time series and psd/coherence spectra
- Utility functions: “ResampleDataset”, “LimitTimeRange”, “AddNameLayer”, “GetNamesMatrix”, etc.
- HELP utility

These operations all require only 1 or 2 “code lines”