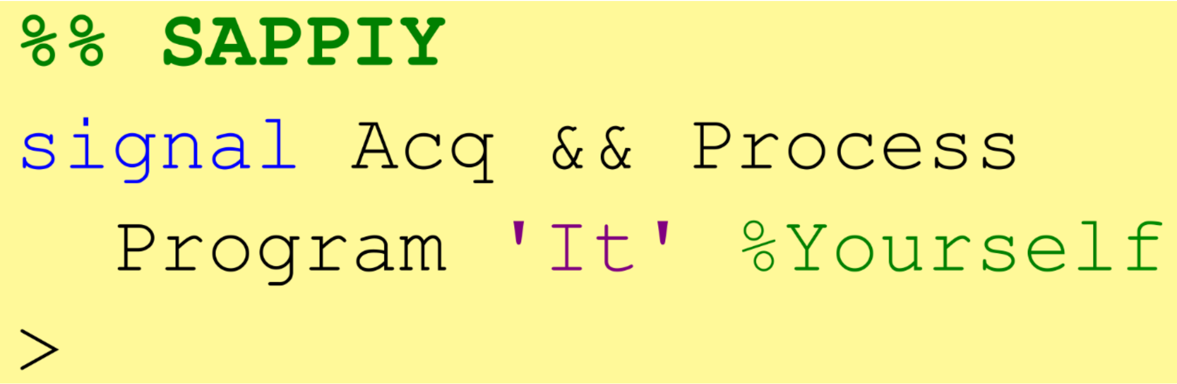
SIGNAL ACQUISITION AND PROCESSING: PROGRAM IT YOURSELF

USER GUIDE



```
%% SAPPIY
signal Acq && Process
  Program 'It' %Yourself
>
```

Author: V. Doguet (valentin.doguet@univ-nantes.fr)

CONTENTS

1. Download, installation and function management 4

1.1. Download, install and run the software 4

1.2. Management of user functions 4

2. Live mode 5

2.1. Session creation and management 5

2.1.1. Create a session 5

2.1.2. Save / Load a session 6

2.2. Analog channel settings 7

2.2.1. Analog channels settings 7

2.2.2. Add arithmetic channels 7

2.3. Acquisition settings and visualization 8

2.3.1. Set acquisition mode / duration 8

2.3.2. Associate acquisition triggering to input and output TTL 8

2.3.3. Add / Remove guideline 9

2.3.4. Visualize and store data 9

2.3.5. Review and analyze stored data 10

2.3.6. Save or clear live data 10

2.4. Input and output TTL settings 11

2.4.1. Input TTL sequencing options 12

2.4.2. Manual input TTL 13

2.4.3. Analog channel-based input TTL 13

2.4.4. Manual output TTL 14

2.4.5. Analog channel-based output TTL 14

3. Analyze mode 15

3.1. Data importation and preparation 15

3.1.1. From live mode (during acquisition session) 15

3.1.2. From Live mode (post-acquisition session) 16

3.1.3. From other source files 17

3.2. Define signal source an treats 18

3.2.1. Choose abscissa and ordinate sources 18

3.2.2. Select / Unselect sequences 18

3.2.3. Set sequence visualization options 19

3.2.4. Apply signal treat on sequences 19

3.3. Process data 19

3.3.1. Associate / Disassociate user functions 20

3.3.2. Run user functions ... 20
3.4. Review processing ... 20
3.5. Export results ... 22
4. User functions formatting ... 23
4.1. Guidelines ... 24
4.2. PreviewTreats ... 25
4.3. ProcessFcns ... 26
4.4. Read_ttl ... 28
4.5. Send_ttl ... 29
4.6. SignalTreats ... 30

1. Download, installation and function management

1.1. Download, install and run the software

The software package is downloadable at the following links:

After downloading, user must extract the "SAPPIY.rar" archive and locate all extracted contents in its preferred directory, labeled ***{ROOT}/*** thereafter. The "SAPPIY.rar" archive contains the following files/folders:

Content	**Type**	**Description**
SAPPIY.m	File	SAPPIY main Matlab® script.
lib	Folder	Folder containing all internal software functions.
licenses	Folder	Folder containing all licenses associated with the software.
resources	Folder	Folder containing all resources files used by the software.
userFcn	Folder	Folder containing editable user functions (see **section 1.2**).

To run the SAPPIY software, open and run ***{ROOT}/SAPPIY.m*** file in Matlab® software[1]. There is no need to add any file or folder to Matlab® path, but the above ***{ROOT}/lib/***, ***{ROOT}/resources/*** and ***{ROOT}/userFcn/*** folders must be located in the same directory than the ***{ROOT}/SAPPIY.m*** main script being run.

1.2. Management of user functions

The ***{ROOT}/userFcn/*** folder (and subfolders) should contain all user functions that the user intends to use in SAPPIY software. None of these functions is necessary to use the software, but they allow user customizing its use according to its needs.

It is essential not to edit folder and subfolder labels of ***{ROOT}/userFcn/*** directory since SAPPIY software uses the same labelling throughout all of its internal functions. However, user is welcome to add or edit files in these subfolders. Each user function type – i.e., each subfolder of the ***{ROOT}/userFcn/*** directory – has a specific function in the SAPPIY software. The following list describes these functions:

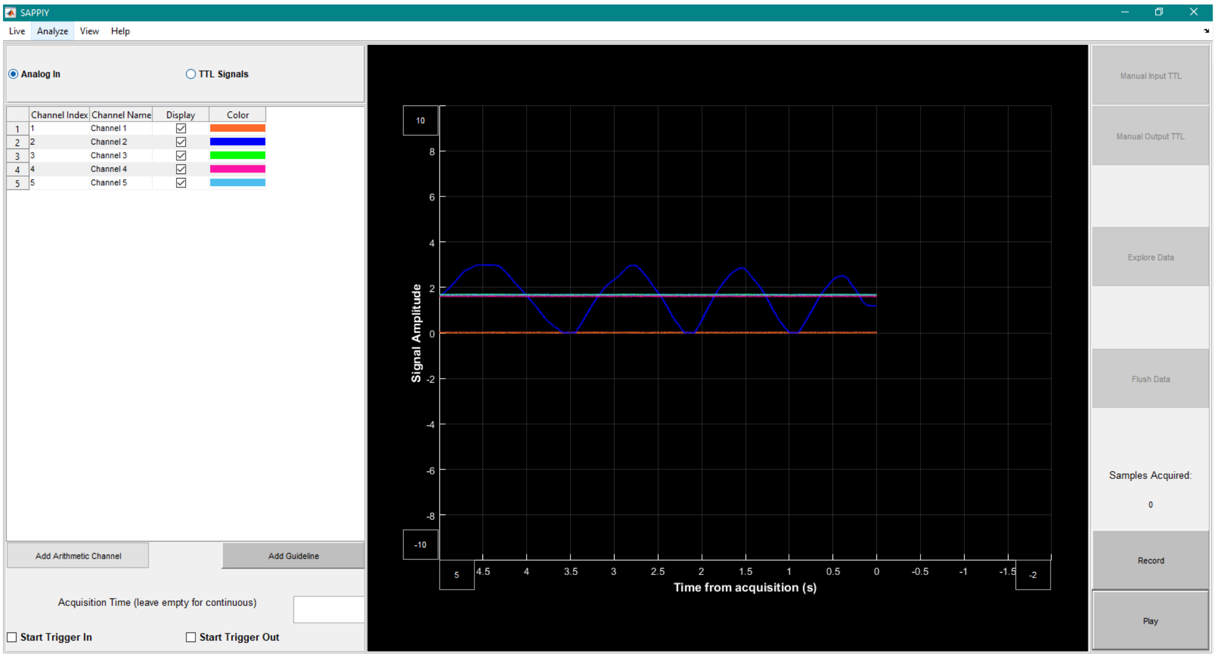
Subfolder	**Description of content**
guidelines/	Functions used to add guidelines to live chart.
previewTreats/	Functions used to apply treatments to live data in the data explorer window.
processFcns/	Functions used to process signal sequences in analyze mode.
read_ttl/	Functions used to mark TTL (i.e., sequence indexing) in a source signal while importing data in analyze mode.
send_ttl/	Functions used to send output TTL, through a defined digital output channel of the connected Daq device in live mode.
signalTreats/	Functions used to apply treatments to a signal source while adding arithmetic channels in live mode or for pre-processing signal sequences in analyze mode.

The above subfolders already contain some default functions, but user should add extra functions to satisfy its needs. Use default functions as examples when writing new user functions. All the default functions contain a header describing how to write a new functions of the same category (i.e., in the same ***{ROOT}/userFcn/*** subfolder), and what are input arguments and what should be the output arguments (see **section 4**).

[1] SAPPIY software was developed with the R2018b Matlab® version. Due to internal function updates between Matlab® versions, it is possible that the SAPPIY software cannot be fully usable with older or newer versions.

2. Live mode

The live mode requires Data Acquisition Matlab® Toolbox and a data acquisition device (Daq) to be installed (both Daq drivers and associated Matlab® add-on) on target computer. Drivers are generally available on the Daq constructor web platform. After Daq drivers have been installed, to install Daq Matlab® add-ons, user must connect a Daq device to the target computer and either navigate to "Get Hardware Support Package" in the Matlab add-on menu or type "daq.getDevices()" in the Matlab® command window to install compatible package (visit Mathworks® website to get started with Data Acquisition Toolbox).



	Channel Index	Channel Name	Display	Color
1	1	Channel 1	☑	🟧
2	2	Channel 2	☑	🟦
3	3	Channel 3	☑	🟩
4	4	Channel 4	☑	🟪
5	5	Channel 5	☑	🟦

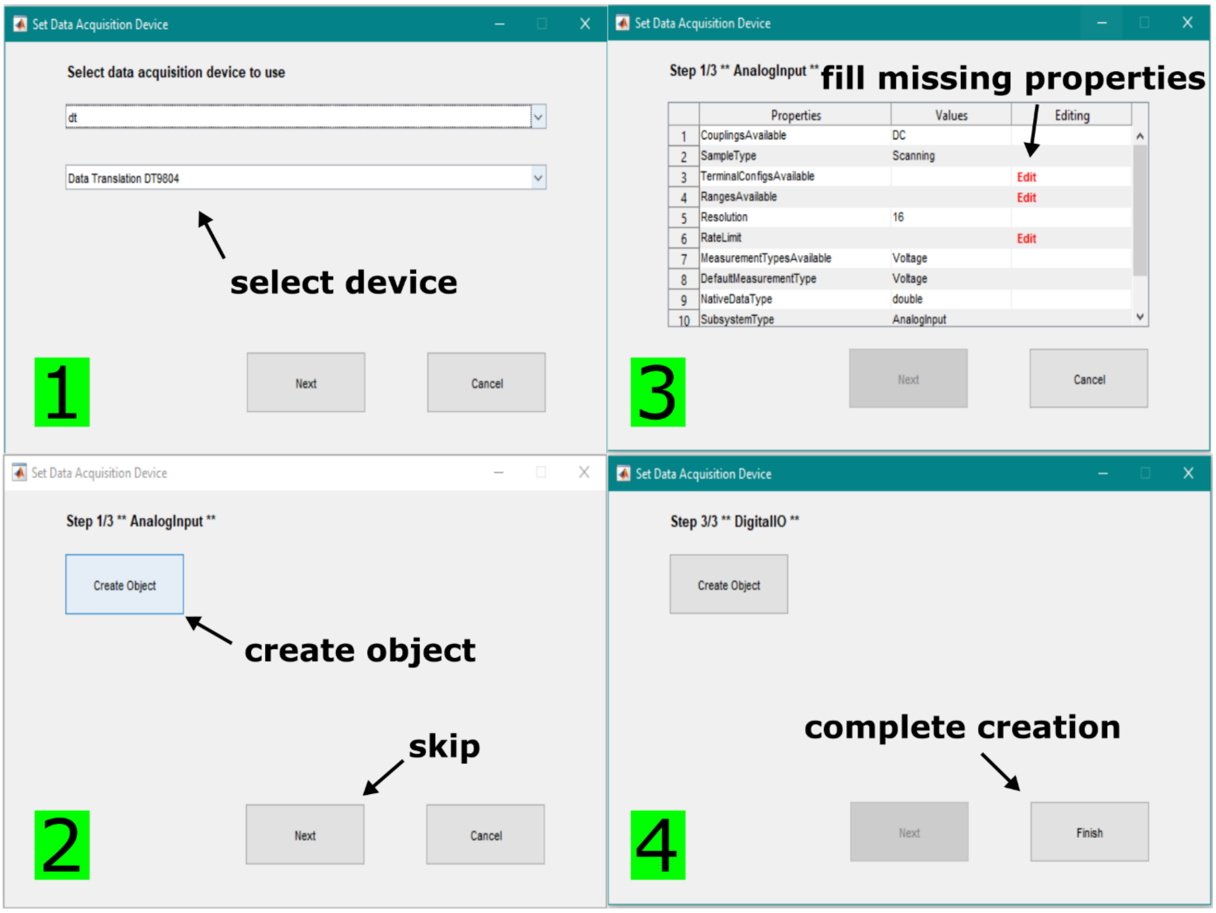
Figure 2-1. *Overview of the live layout. Live layout is separated in different panels: a "Live Chart" panel in the middle that displays analog signals and guideline; A "Live Controls" panel on the right that contains different controllers for live data and TTL management; And either an "Analog Channels" panel or a "TTL Settings" panel (depending which one is selected) on the left to manage acquisition and TTL settings.*

2.1. Session creation and management

2.1.1. Create a session

When running ***{ROOT}/SAPPIY.m*** main script, a graphical user interface pops up on screen. Go to "Live > Session > New Session" menu to launch the Daq session creator tool[2] that helps creating data acquisition sessions (See **Figure 2-2** & ***{ROOT}/resources/videos/live/01.mp4***).

[2] The Daq session creator tool was developed with Data Translation and National Instruments data acquisition devices. This tool might encounters bugs when using different vendors or devices. Contact the author if you encounter any troubles to extend compatibility of the software



Set Data Acquisition Device

Select data acquisition device to use

dt

Data Translation DT9804

Set Data Acquisition Device

Step 1/3 ** AnalogInput **

	Properties	Values	Editing
1	CouplingsAvailable	DC	
2	SampleType	Scanning	
3	TerminalConfigsAvailable		Edit
4	RangesAvailable		Edit
5	Resolution	16	
6	RateLimit		Edit
7	MeasurementTypesAvailable	Voltage	
8	DefaultMeasurementType	Voltage	
9	NativeDataType	double	
10	SubsystemType	AnalogInput	

Set Data Acquisition Device

Step 1/3 ** AnalogInput **

Set Data Acquisition Device

Step 3/3 ** DigitalIO **

Figure 2-2. *Overview of the Daq session creator tool that enables creating data acquisition sessions.*

In the Daq session creator tool, the first step allow user choosing which installed Daq device to use (see panel 1 in **Figure 2-2**). Then, there is as much steps as Daq device gets functionalities (e.g., analog input/output channels, digital input/output channels, counters).

At each step, click **Create Object** button to create the associated functionality in the session object, or skip the step by clicking **Next** button (see panel 2 in **Figure 2-2**). When creating session objects, a table displays all properties of the object being created, and the user is expected to fill all missing properties by clicking the **Edit** text in table (see panel 3 in **Figure 2-2**). Once all missing properties have been filled, click **Next** button to validate the current session object. At the last step of the Daq session creator tool, click **Finish** button to complete session creation (see panel 4 in **Figure 2-2**).

In this software release, only analog input and digital output objects are usable. Therefore, other session objects must be skipped.

If a session is currently running when selecting “Live > Session > New Session” menu, the current session is erased by the new one.

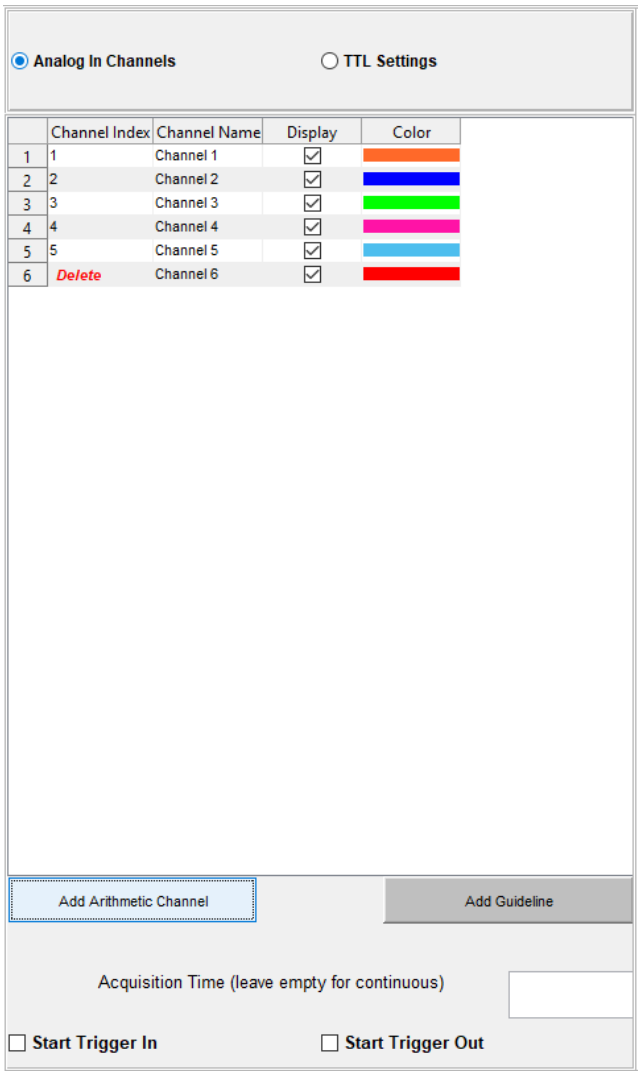
2.1.2. Save / Load a session

After having created a session, the software invites user to save the created session. Session files are Matlab® files (**‘.mat’** extension) that can be saved in any location on the computer, and reopened later using “Live > Session > Load Existent Session” menu (see ***{ROOT}/resources/videos/live/02.mp4***). Session can also be saved later, through “Live > Session > Save Current Session” menu, if some session options (e.g., analog channel labels, arithmetic channels added) have been edited.

If a session is currently running when selecting “Live > Session > Load Existent Session” menu, the current session is erased by the loaded one.

2.2. Analog channel settings

To define analog channels settings, select “Analog Channels” radio button on the top left panel of the live layout (see **Figure 2-1**).



● Analog In Channels ○ TTL Settings

	Channel Index	Channel Name	Display	Color
1	1	Channel 1	☑	🟧
2	2	Channel 2	☑	🟦
3	3	Channel 3	☑	🟩
4	4	Channel 4	☑	🟪
5	5	Channel 5	☑	🟦
6	*Delete*	Channel 6	☑	🟥

Acquisition Time (leave empty for continuous)

□ Start Trigger In □ Start Trigger Out

Figure 2-3. *”Analog Channels” live panel.*

2.2.1. Analog channels settings

In the “Analog Channels” panel, user can edit label and color as well as display/hide signal from live chart for each analog channel (see **Figure 2-3** & ***{ROOT}/resources/videos/live/03.mp4***). User can also edit chart background color by right clicking in the “Live Chart” panel and selecting “Set Background Color” menu (see ***{ROOT}/resources/videos/live/04.mp4***).

2.2.2. Add arithmetic channels

For calibration or feedback purposes, user can add arithmetic channels from existent analog channel sources by applying an equation or a user function to one or multiple source channel(s).

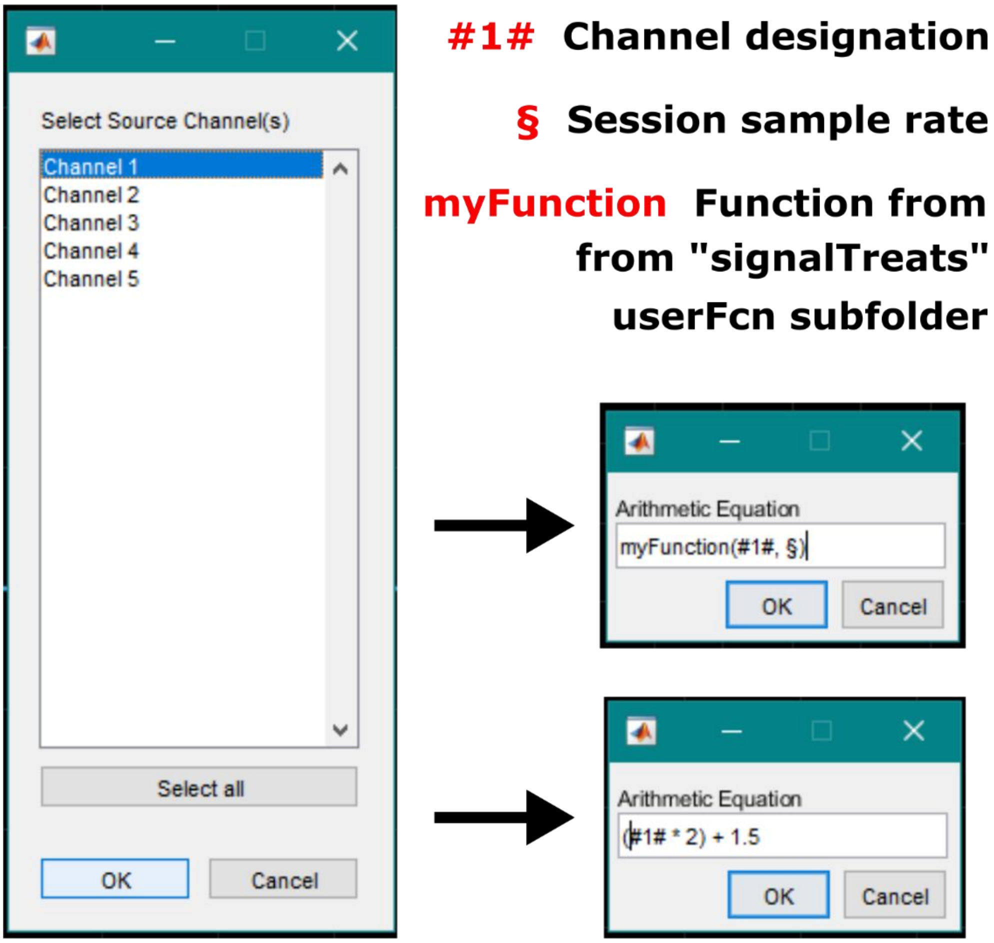
To do so, click Add Arithmetic Channel button in the “Analog Channels” panel (see **Figure 2-3**) and select analog channel(s) to use as source channel(s) in the list that pops up. Then, user must write, in Matlab®

programmatic language, the equation or designate the user function and arguments to use to build arithmetic channel (see **Figure 2-4** & ***{ROOT}/resources/videos/live/05.mp4***).

A source channel must be designated by its channel index number surrounded by hashes (#).

Session sample rate can either be written with number or designed by (§) symbol.

When using functions, user must use function from ***{ROOT}/userFcn/signalTreats/*** folder.



Select Source Channel(s)

Channel 1
Channel 2
Channel 3
Channel 4
Channel 5

Arithmetic Equation

myFunction(#1#, §)

Arithmetic Equation

(#1# * 2) + 1.5

Figure 2-4. *Steps to add arithmetic channel from analog channel source(s).*

To delete an arithmetic channel, click **Delete** text in the analog channel list (see **Figure 2-3**).

2.3. Acquisition settings and visualization

2.3.1. Set acquisition mode / duration

A Daq device can either be set in continuous or finite mode[3]. To use a Daq device in continuous mode, user must leave the "Acquisition Duration" edit box empty (see **Figure 2-3**). To set a Daq device in finite mode, user must determine the desired acquisition duration (in seconds) in the "Acquisition Duration" edit box (see ***{ROOT}/resources/videos/live/06.mp4***).

2.3.2. Associate acquisition triggering to input and output TTL

User can set acquisition triggering options to either send an output TTL when acquisition is started (Start Trigger Out) or start acquisition from an external input TTL (Start Trigger In). To associate acquisition triggering to a Start Trigger In/Out, tick corresponding ☐ Start Trigger In and ☐ Start Trigger Out checkboxes at the bottom of the "Analog Channels" panel (see **Figure 2-3**).

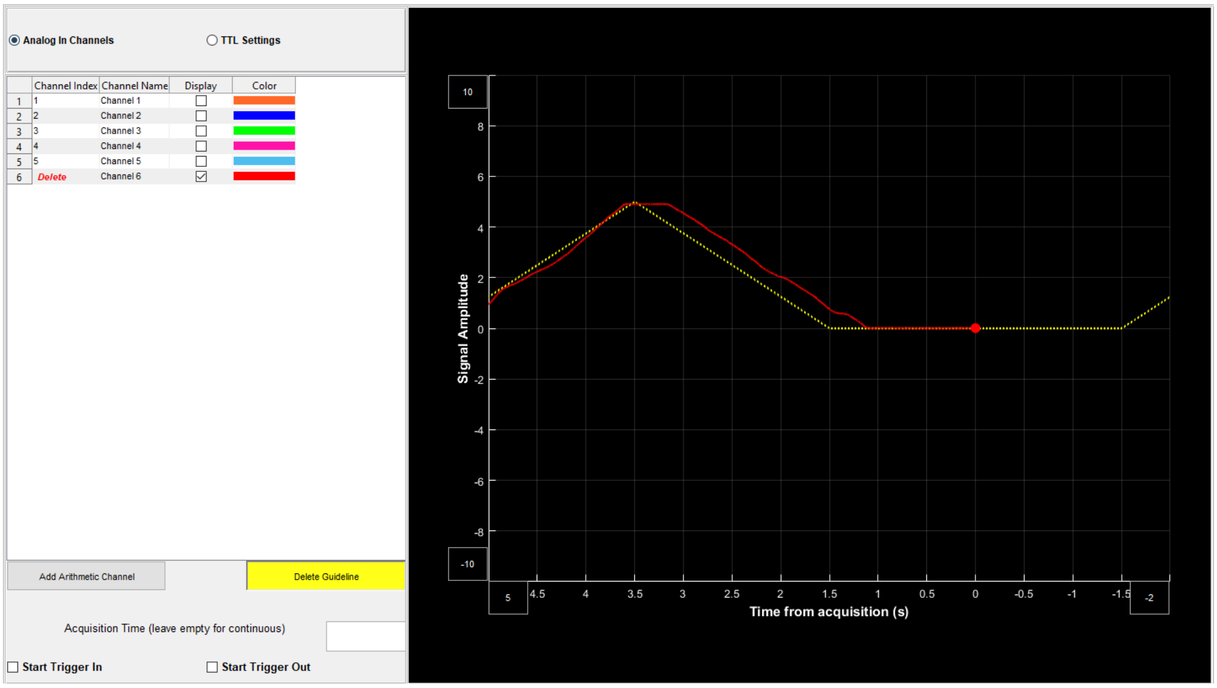
[3] Switching between Daq acquisition modes modify the way signal is displayed on live chart. Beware that this is also likely to modify the way to create guidelines as well (see **section 2.3.3**).

By ticking ☐ Start Trigger In and ☐ Start Trigger Out checkboxes, software ask user to define which connection source to use from the connected Daq device[4]. If no connection is available, or a wrong connection is set, the software throw an error and cancel acquisition-triggering option.

2.3.3. Add / Remove guideline

For feedback purpose, user may want to add a guideline to live chart. To do so, user should first write its own guideline user function (see section 4.1) and store it in ***{ROOT}/userFcn/guidelines/*** folder. By default, some guideline user functions are available in this folder. After having coded a guideline user function, user must click Add Guideline button in the "Analog Channels" panel (see **Figure 2-3**) and select the desired guideline user function in the dialog list. This action automatically runs guideline user function selected and should, if the function is properly written, display the associated guideline on live chart (see **Figure 2-5** & ***{ROOT}/resources/videos/live/07.mp4***).

The software allow displaying only one guideline at a time. To delete a guideline, click Delete Guideline button in the "Analog Channels" panel (see **Figure 2-5**).



● Analog In Channels ○ TTL Settings

	Channel Index	Channel Name	Display	Color
1	1	Channel 1	☐	🟧
2	2	Channel 2	☐	🟦
3	3	Channel 3	☐	🟩
4	4	Channel 4	☐	🟪
5	5	Channel 5	☐	🟦
6	Delete	Channel 6	☑	🟥

Add Arithmetic Channel Delete Guideline

Acquisition Time (leave empty for continuous)

☐ Start Trigger In ☐ Start Trigger Out

Figure 2-5. *Overview of a guideline added to live chart.*

2.3.4. Visualize and store data

To start acquiring data in Daq device, switch Play button to Play state in the "Live Controls" panel (see **Figure 2-1**). This action starts data acquisition and visualization but does not store data. To store data while acquiring, switch Record button to Record state in the "Live Controls" panel. Note that Record button does not start/stop acquisition, it is only a toggle button that activate/deactivate data storing.

When samples are stored, acquired samples count Samples 3300 increases in the "Live Controls" panel.

[4] Note that Start Trigger options are not available for all Daq devices. Further, at the time of this software release, there is no Stop Trigger options available for Data Acquisition Matlab® Toolbox.

2.3.5. Review and analyze stored data

Once samples have been stored, click Explore Data button and select channel(s) to explore. Then, a new window pops up and displays all channels selected in successive panels (see

Figure 2-6). From there, select what source to set as abscissa between Time, Samples or any of the selected channels in the scroll-down list in the top-right corner of data explorer window (see ***{ROOT}/resources/videos/live/08.mp4****)*.

In data explorer window, it is possible to analyze some parameters from signals displayed. To do so, select a user function from the scroll-down list at the top-left corner of each signal panel. Scroll-down lists contain all user functions from the ***{ROOT}/userFcn/previewTreats/*** folder. Therefore, users can write their own functions and store them in the dedicated folder (see section 4.2). It is important to note that only signal comprise between left and right X limits of axis is sent to user functions. This means that zoom in/out can be used before choosing user functions to apply treatment to a specific signal section (e.g., to get max peak value of a limited signal section).

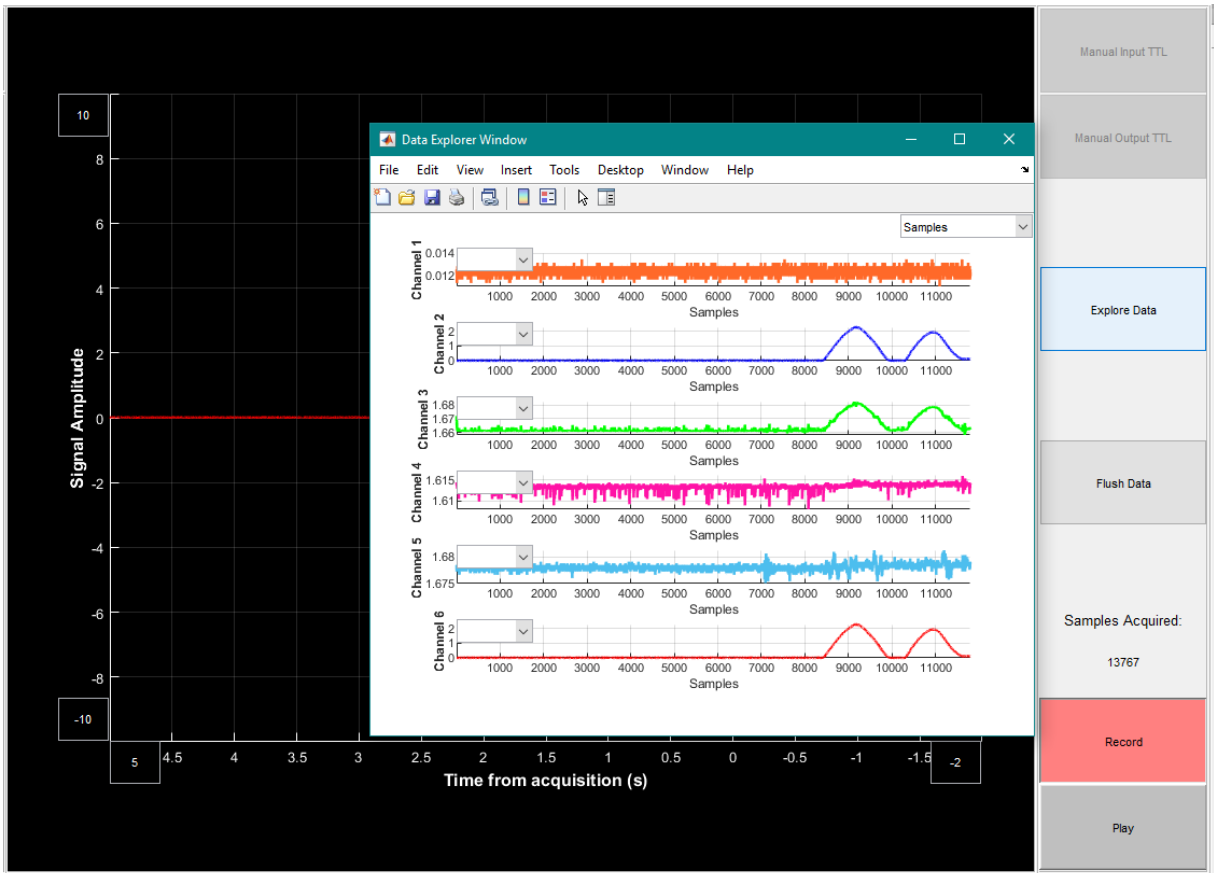


Figure 2-6*. Data explorer window.*

2.3.6. Save or clear live data

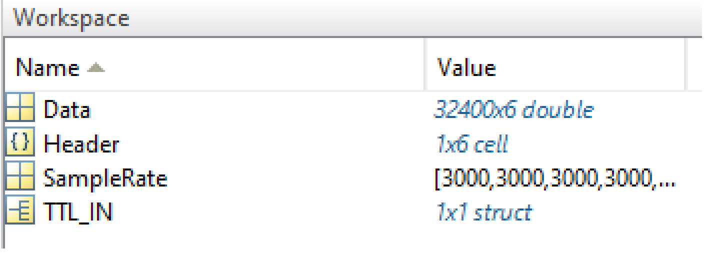
If samples have been stored, select "Live > Save Live Data" menu and define file path, name and extension ('.mat' or '.csv') in the explorer window to save live data (see ***{ROOT}/resources/videos/live/09.mp4****)*.

Conversely, delete live data by clicking Flush Data button in the "Live Controllers" panel (see Figure 2-6

Figure 2-6).

For '.mat' saving format, the saved file 'myFile.mat' contains the following variables that the user can deal with in Matlab® program thereafter (see 'mat_file.mat' example in ***{ROOT}/resources/example_files/*** folder and Figure 2-7):

Variable	Sub-variable	Description
Data		*m-by-n* matrix of double elements, with *m* representing the number of samples and *n* representing the number of channel plus a time channel (the first column). This variable contains data.
Header		*1-by-n* cell vector of char elements, with *n* representing the number of channel plus a time header (the first element). This variable contains data headers.
SampleRate		*1-by-n* vector of double elements, with *n* representing the number of channel plus a time header (the first element). This variable contains data sample rate.
TTL_IN	indexes	*1-by-n* vector of double elements, with *n* representing the number of input TTL marks. This structure field contains input TTL indexes.
	rate	*1-by-1* vector of double element. This structure field contains rate at which input TTL are marked.

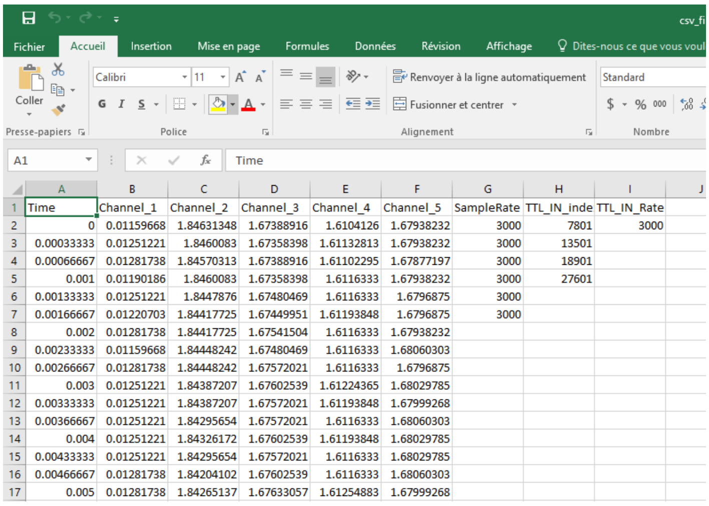


Workspace

Name	Value
Data	32400x6 double
Header	1x6 cell
SampleRate	[3000,3000,3000,3000,...
TTL_IN	1x1 struct

Figure 2-7. *Matlab® workspace view of variables stored in a 'myFile.mat' file saved from live mode.*

For **'.csv'** saving format, the saved 'myFile.csv' is organized in columns as follows (see 'csv_file.csv' example in ***{ROOT}/resources/example_files/*** folder and **Figure 2-8**): the 1st column corresponding to data time stamps; Columns from 2 to *n+1* corresponding to data from the *n* number of analog channels; Column *n+2* corresponding to data rates; Column *n+3* corresponding to input TTL indexes; And column *n+4* corresponding to input TTL rate.



Time

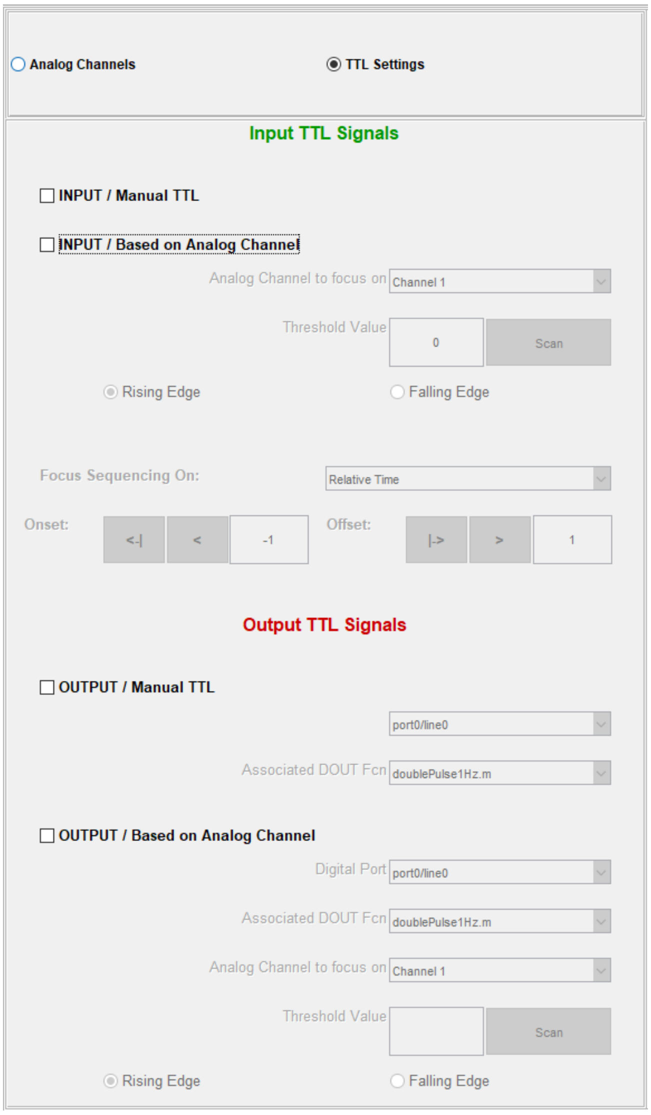
	A	B	C	D	E	F	G	H	I
1	Time	Channel_1	Channel_2	Channel_3	Channel_4	Channel_5	SampleRate	TTL_IN_inde	TTL_IN_Rate
2	0	0.01159668	1.84631348	1.67388916	1.6104126	1.67938232	3000	7801	3000
3	0.00033333	0.01251221	1.8460083	1.67358398	1.61132813	1.67938232	3000	13501	
4	0.00066667	0.01281738	1.84570313	1.67388916	1.61102295	1.67877197	3000	18901	
5	0.001	0.01190186	1.8460083	1.67358398	1.6116333	1.67938232	3000	27601	
6	0.00133333	0.01251221	1.8447876	1.67480469	1.6116333	1.6796875	3000		
7	0.00166667	0.01220703	1.84417725	1.67449951	1.61193848	1.6796875	3000		
8	0.002	0.01281738	1.84417725	1.67541504	1.6116333	1.67938232			
9	0.00233333	0.01159668	1.84448242	1.67480469	1.6116333	1.68060303			
10	0.00266667	0.01281738	1.84448242	1.67572021	1.6116333	1.6796875			
11	0.003	0.01251221	1.84387207	1.67602539	1.61224365	1.68029785			
12	0.00333333	0.01251221	1.84387207	1.67572021	1.61193848	1.67999268			
13	0.00366667	0.01251221	1.84295654	1.67572021	1.6116333	1.68060303			
14	0.004	0.01251221	1.84326172	1.67602539	1.61193848	1.68029785			
15	0.00433333	0.01251221	1.84295654	1.67572021	1.6116333	1.68029785			
16	0.00466667	0.01281738	1.84204102	1.67602539	1.6116333	1.68060303			
17	0.005	0.01281738	1.84265137	1.67633057	1.61254883	1.67999268			

Figure 2-8. *Spreadsheet view of a 'myFile.csv' file saved from live mode.*

2.4.Input and output TTL settings

The software gather a live mode and an analyze mode. While both modes may work separately, it is possible to send signal sequences from live mode to analyze mode for real-time live processing purpose. Signal sequences are sent from live mode by defining input TTL options (see **sections 2.4.2** and **2.4.3**).

Further, if a digital output object has been set for the live session (see **section 2.1.1**), specific output TTL sequences can be sent from Daq device digital output channel(s) to external device(s) (see **sections 2.4.4** and **2.4.5**).



○ Analog Channels ● TTL Settings

Input TTL Signals

□ INPUT / Manual TTL

□ INPUT / Based on Analog Channel

Analog Channel to focus on: Channel 1

Threshold Value: 0 Scan

● Rising Edge ○ Falling Edge

Focus Sequencing On: Relative Time

Onset: <-| < -1 Offset: |-> > 1

Output TTL Signals

□ OUTPUT / Manual TTL

port0/line0

Associated DOUT Fcn: doublePulse1Hz.m

□ OUTPUT / Based on Analog Channel

Digital Port: port0/line0

Associated DOUT Fcn: doublePulse1Hz.m

Analog Channel to focus on: Channel 1

Threshold Value: Scan

● Rising Edge ○ Falling Edge

Figure 2-9. *"TTL Settings" live panel.*

2.4.1. Input TTL sequencing options

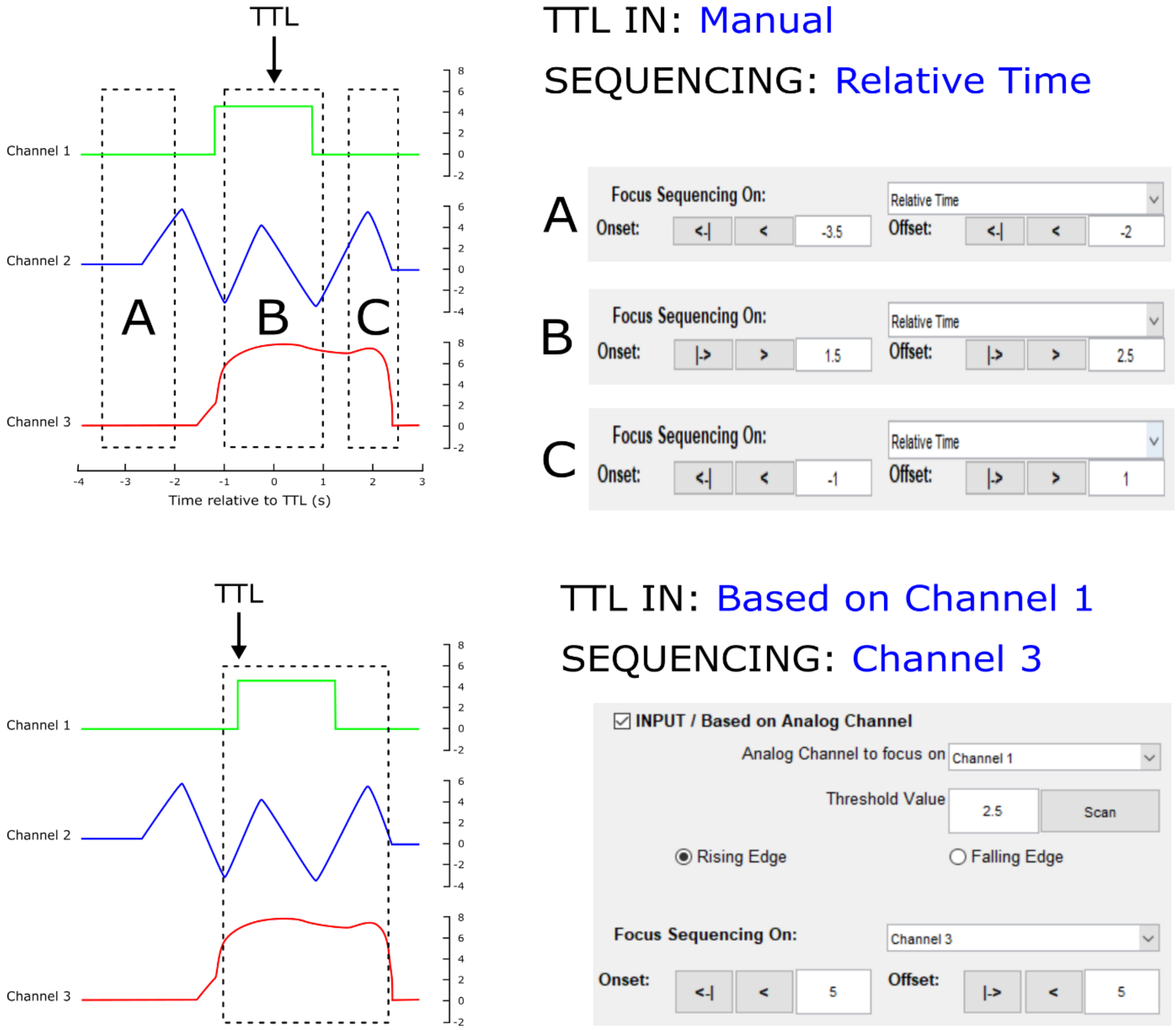
To send data sequences to the analyze layout, an input TTL must be marked in the live mode. A marked input TTL represents a time stamp at which a condition is reached. To mark an input TTL, conditions can either be a manual button click (see **section 2.4.2**), or reach of an analog channel value (see **section 2.4.3**).

In any case, user must determine how signal sequences must be sequenced according to the input TTL stamp. To set signal sequences, define a target source to focus on (i.e., time relative to TTL or any analog channel) as well as onset and offset values, edges (i.e., rising or falling) and direction from TTL (i.e., before or after TTL). Direction from input TTL can either be before <-| or after |-> TTL stamp. Edge can either be rising > or falling <.

For time relative sequencing, combining <-| and < with value "**-1**" for sequence onset (see **C** example in **Figure 2-10** top panel) means to start sequencing at the first time value lower than **-1** second relative to input TTL (in practice TTL time is 0, so 1 second before is -1). In parallel, combining |-> and

> with value “**1**” for sequence offset (see **C** example in **Figure 2-10** top panel) means to stop sequencing at the first time value higher than 1 second relative to input TTL.

While direction and edge are related to each other for time relative sequencing source, these options are independent for analog channel sequencing source. Indeed, |-> and value “**5**” in channel 3 for offset sequencing can either be combined with > or < edges, which mean to use first value higher or lower than 5 in channel 3, respectively.



TTL IN: Manual

SEQUENCING: Relative Time

A
Focus Sequencing On: Relative Time
Onset: <-| < -3.5
Offset: <-| < -2

B
Focus Sequencing On: Relative Time
Onset: |-> > 1.5
Offset: |-> > 2.5

C
Focus Sequencing On: Relative Time
Onset: <-| < -1
Offset: |-> > 1

TTL IN: Based on Channel 1

SEQUENCING: Channel 3

☑ INPUT / Based on Analog Channel
Analog Channel to focus on: Channel 1
Threshold Value: 2.5 Scan
● Rising Edge ○ Falling Edge
Focus Sequencing On: Channel 3
Onset: <-| < 5
Offset: |-> < 5

Figure 2-10. *Examples of signal sequencing focused on relative time (top panel) or specific analog channel values (bottom panel), and according to manual (top panel) or analog-based input TTL (bottom panel).*

2.4.2. Manual input TTL

To mark input TTL manually, enable Manual Input TTL button by ticking the ☑ INPUT / Manual TTL checkbox in the “TTL Settings” live panel (see **Figure 2-9**). Then, input TTL would be marked manually every time Manual Input TTL button is clicked during data acquisition – i.e., when Record button is activated – (see ***{ROOT}/resources/videos/live/10.mp4***).

2.4.3. Analog channel-based input TTL

To mark input TTL according to a specific analog channel, tick ☑ INPUT / Based on Analog Channel checkbox in the “TTL Settings” live panel (see **Figure 2-9**). Then, define which channel to focus on

Analog Channel to focus on Channel 2 as well as value to be reached Threshold Value 2.5 and the crossing edge ● Rising Edge or ● Falling Edge. Value can be directly set in the dedicated edit box or by clicking Scan button to read the current value for the designed analog channel.

When Record button is activated and the defined analog channel source reaches the defined value, according to the specified edge, an input TTL is marked. From there, no input TTL will be marked until the defined analog channel source reaches the defined value according to the opposite edge (see ***{ROOT}/resources/videos/live/11.mp4***).

2.4.4. Manual output TTL

To send output TTL sequences manually, enable Manual Output TTL button by ticking ☑ **OUTPUT / Manual TTL** checkbox in the "TTL Settings" live panel (see **Figure 2-9**). Secondly, define which digital output channel to use Digital Port port0/line0 and what output TTL sequence to send Associated DOUT Fcn doublePulse1Hz.m through the selected channel. Output TTL sequences should be user functions from the ***{ROOT}/userFcn/send_ttl/*** folder (see **section 4.5**). If so, the specified user function would be send through the specified digital output channel every time Manual Output TTL button is clicked in the "Live Controls" panel (see ***{ROOT}/resources/videos/live/12.mp4***).

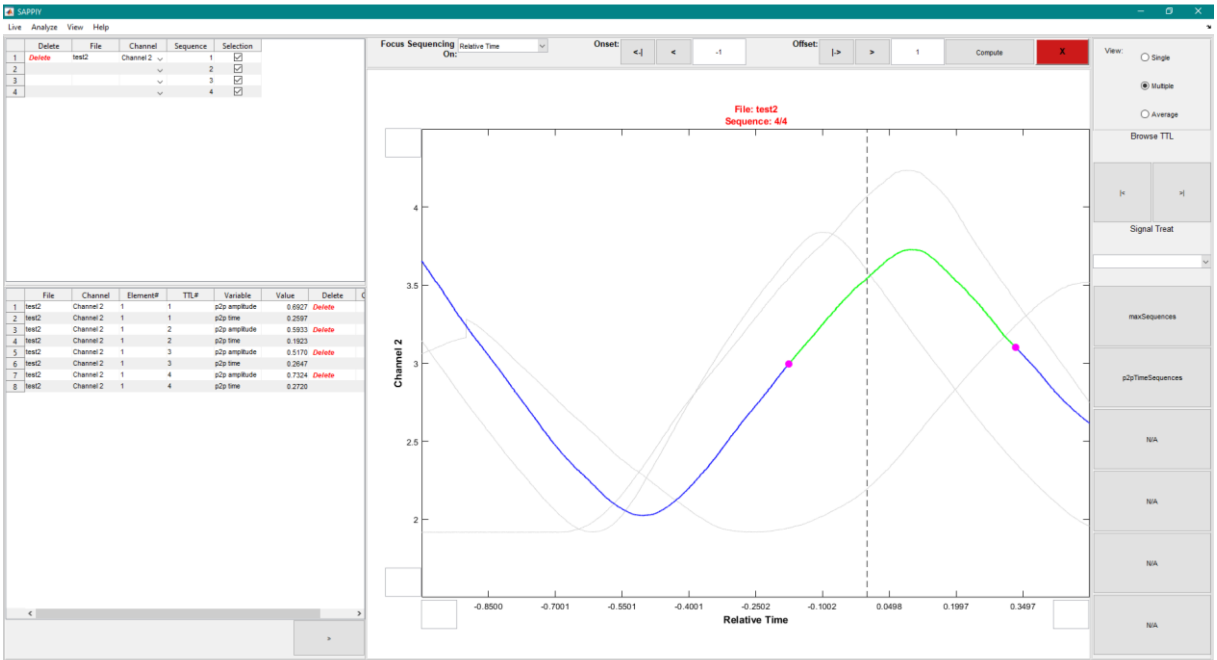
2.4.5. Analog channel-based output TTL

To send output TTL sequences according to a specific analog channel, tick ☑ **OUTPUT / Based on Analog Channel** checkbox in the "TTL Settings" live panel (see **Figure 2-9**). Then, define which channel to focus on Analog Channel to focus on Channel 2 as well as value to be reached Threshold Value 2.5 and the crossing edge ● Rising Edge or ● Falling Edge. Value can be directly set in the dedicated edit box or by clicking Scan button to read the current value for the designed analog channel. As for manual output TTL, define which digital output channel to use Digital Port port0/line0 and what output TTL sequence to send Associated DOUT Fcn doublePulse1Hz.m through the selected channel. Output TTL sequences should be user functions from the ***{ROOT}/userFcn/send_ttl/*** folder (see **section 4.5**).

When the defined analog channel source reaches the defined value, according to the specified edge, the specified user function would be send through the specified digital output channel. From there, no output TTL will be send until the defined analog channel source reaches the defined value according to the opposite edge (see ***{ROOT}/resources/videos/live/13.mp4***).

3. Analyze mode

The software analyze mode by itself does not require any specific Matlab® Toolbox to be used. However, given that this mode is dedicated to analyze signal sequences, users might require specific toolboxes (e.g., Signal Processing Toolbox) while intending to create processing user functions.

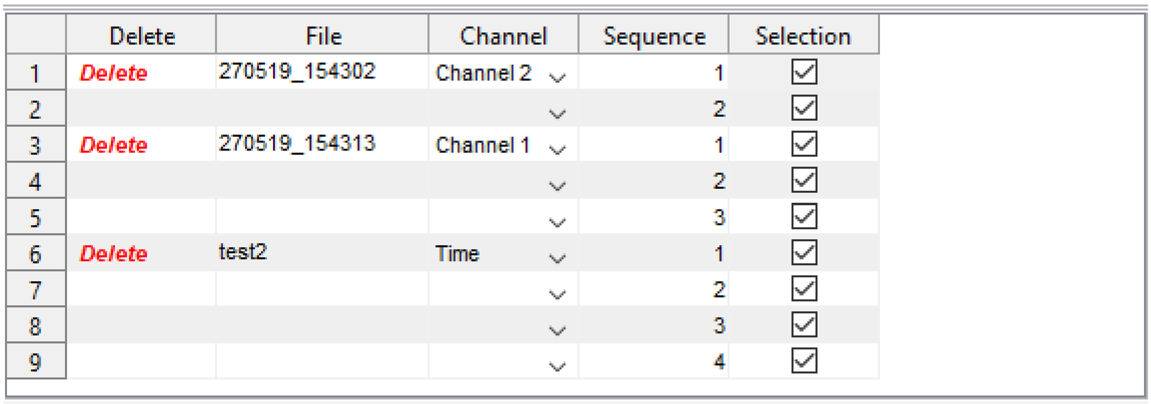


	Delete	File	Channel	Sequence	Selection
1	Delete	test2	Channel 2	1	☑
2				2	☑
3				3	☑
4				4	☑

	File	Channel	Element#	TTL#	Variable	Value	Delete
1	test2	Channel 2	1	1	p2p amplitude	0.6927	Delete
2	test2	Channel 2	1	1	p2p time	0.2597	
3	test2	Channel 2	1	2	p2p amplitude	0.5933	Delete
4	test2	Channel 2	1	2	p2p time	0.1923	
5	test2	Channel 2	1	3	p2p amplitude	0.5170	Delete
6	test2	Channel 2	1	3	p2p time	0.2647	
7	test2	Channel 2	1	4	p2p amplitude	0.7324	Delete
8	test2	Channel 2	1	4	p2p time	0.2720	

Figure 3-1. *Overview of the analyze layout. Analyze layout is separated in different panels: an "Analyze Chart" panel, in the middle, that displays signal sequences; a "Sequence Explorer" panel, in the top left corner, that lists all imported files and sequences; a "Result Explorer" panel, in the bottom left corner, that lists results from signal processing; and an "Analyze Controls" panel, on the right, that contains visualizing and processing controllers. A "Sequencing" panel is available at the top of the analyze layout while importing external files (i.e., this panel is disabled when analyzing sequences from live mode).*

3.1. Data importation and preparation



	Delete	File	Channel	Sequence	Selection
1	Delete	270519_154302	Channel 2	1	☑
2				2	☑
3	Delete	270519_154313	Channel 1	1	☑
4				2	☑
5				3	☑
6	Delete	test2	Time	1	☑
7				2	☑
8				3	☑
9				4	☑

Figure 3-2. *View of the "Sequence Explorer" panel containing sequences imported from both live mode and an external file.*

3.1.1. From live mode (during acquisition session)

When input TTL settings are defined in live mode (see **sections 2.4.2** and **2.4.3**) and input TTL condition (i.e., manual or analog-based mark) is reached, the signal sequences that user defined in live sequencing options (see **section 2.4.1**) is thrown to the analyze layout.

A new analyze file, labeled according to creation date and time (see **Figure 3-2**), is created every time the Record button is toggled in live layout (see **Figure 2-1**). This means that, once Record button is toggled, all input TTL marked are associated as sequences of the same analyze file until Record button is deactivated. Note that toggling Play button does not affect the way sequences are regrouped in analyze layout.

3.1.2. From Live mode (post-acquisition session)

The software also enables processing data in offline mode, after live data have been acquired and saved from live mode.

To import live data files, select "Analyze > Import File(s)" menu and browse to the desired path and file (see ***{ROOT}/resources/videos/analyze/14.mp4***). Live data files can be either of **'.mat'** or **'.csv'** formats (see **section 2.3.6**).

When importing a live file, the software imports all channel 'Data', 'Headers' and 'Sample rates'. However, even if the imported file contains 'TTL_IN' information, the software asks user whether it should use available input TTL indexes (i.e., embedded 'TTL_IN' information) or search for TTL in any source channel from the imported file (see **Figure 3-3**). If user chooses looking for TTL in a source channel, then the source channel to search in and the user function to use must be defined in the following importing steps (see **Figure 3-4**).

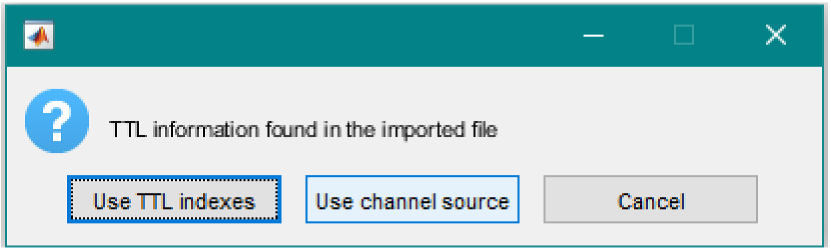
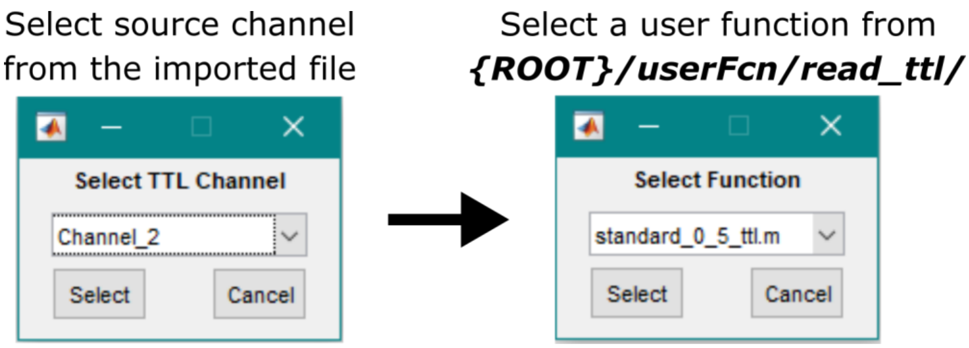

TTL information found in the imported file

Figure 3-3. *Pop-up dialog box displayed when importing a file that contains TTL information (i.e., indexes and rate).*


Select source channel from the imported file

Select a user function from ***{ROOT}/userFcn/read_ttl/***

Select TTL Channel

Channel_2

Select Function

standard_0_5_ttl.m

Figure 3-4. *Steps when choosing to look for TTL in a source channel.*

After having defined TTL indexes in the imported file, the file is available in software analyze layout, but requires sequencing before being processed. The "Analyze Chart" panel displays the entire source channel selected in the "Sequence Explorer" panel (see **section 3.2.1**) until a sequencing is computed (see **Figure 3-5**).

Sequencing is performed according to TTL indexes, and works the same way than in live mode (see **section 2.4.1**). To apply sequencing options defined in the "Sequencing" panel, click Compute button in the "Sequencing" panel. Once sequencing is performed, all sequences are displayed relatively to the signal source Focus Sequencing On: Relative Time set in sequencing options. Contrarily to sequences from live mode during an acquisition session, signal sequencing can be recomputed when working in offline mode. To do so, clear the current sequencing by clicking X button in the "Sequencing" panel and by clicking Compute button after having modified sequencing options (see ***{ROOT}/resources/videos/analyze/15.mp4***).

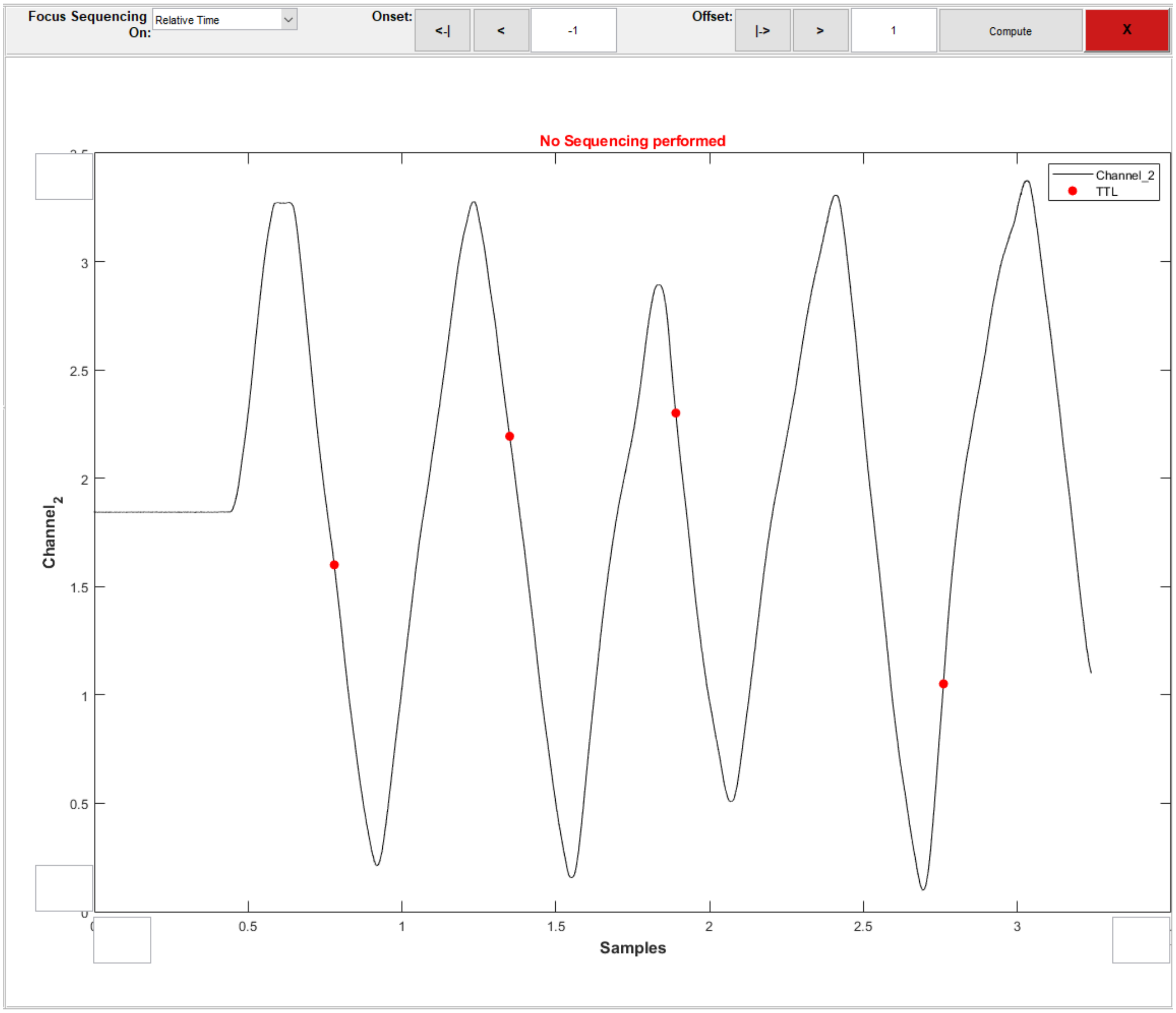


Figure 3-5. View of the "Sequencing" and "Analyze Chart" panels before an imported file has been sequenced.

3.1.3. From other source files

The software also enables processing data from a file of any other sources, as long as data organization respects the software importing requirements.

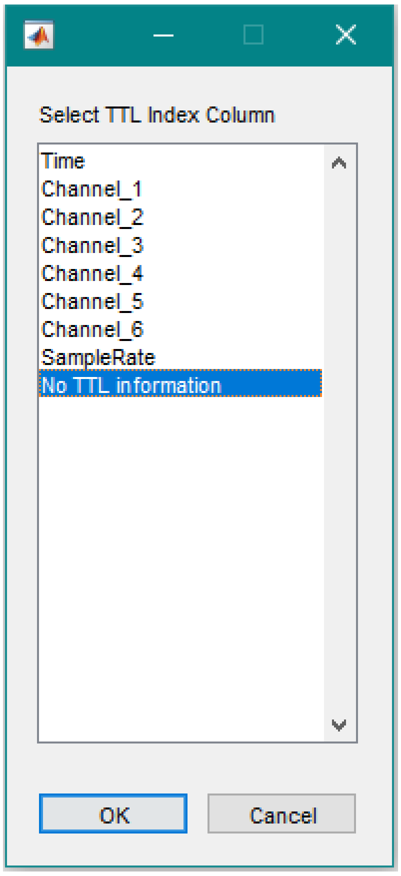
To import data from external files, select "Analyze > Import File(s)" menu and browse to the desired path and file. External files can either be of **'.mat'**, **'.csv'**, **'.xls'** or **'.xlsx'** formats. To make importing easier and quicker, external files should respect the same structure and header labelling than those created by the software live mode (see **section 2.3.6** and examples in ***{ROOT}/resources/example_files/*** folder), but this is not a requirement.

An external file must contain 'Data' and 'SampleRate' columns to be successfully imported. As much 'Data' columns as channel sources, and one single 'SampleRate' column with as much rows as 'Data' sources.

Conversely, 'Header' and 'TTL_IN' information are optional. Indeed, the software automatically assigns standard header labels (i.e., "Column 1", "Column 2", etc.) if absent, and asks user to define which column to associate to 'Data', 'SampleRate' and 'TTL_IN' (optional) in the consecutive step[5]. If the file being imported contains TTL information, which is not labelled the same than files originally created in the live mode[5], user

[5] If 'SampleRate', 'TTL_IN_indexes' and 'TTL_IN_rate' header labelling is the same for imported files than files originally created in the live mode, 'Data', 'SampleRate', 'TTL_IN_indexes' and 'TTL_IN_rate' column assignment is automatically performed.

must define TTL index and rate columns in the dedicated pop-up dialog box. In the opposite case, user must select “No TTL information” in the dedicated pop-up dialog box (see **Figure 3-6**), and the software would consecutively ask user to search for TTL in a source channel, using a user function, as for offline data from live mode (see **section 3.1.2** and **Figure 3-4**).



Select TTL Index Column

Time
Channel_1
Channel_2
Channel_3
Channel_4
Channel_5
Channel_6
SampleRate
No TTL information

Figure 3-6. *Pop-up dialog box that enables selecting TTL indexes column in the imported file.*

Once TTL indexes defined, an imported file requires to be sequenced. As for offline data from live mode (see **section 3.1.2**), use the “Sequencing” panel to compute sequencing (see **Figure 3-5**).

3.2.Define signal source an treats

3.2.1. Choose abscissa and ordinate sources

Ordinate. To set the signal source to display as ordinate in the “Analyze Chart” panel, use the scroll-down menu of the ‘Channel’ column in the “Sequence Explorer” panel (see **Figure 3-2**).

Abscissa [live mode during acquisition]. When importing sequences directly from live mode, analog channels are displayed according to the abscissa signal source defined in the live sequencing options (i.e., relative time or any analog channel; see **section 2.4.1**). There is no way to display sequences according to another signal source.

Abscissa [post-acquisition or external file]. The signal source displayed as abscissa in the “Analyze Chart” panel is the source signal defined in the sequencing options in the “Sequencing” panel (see **Figure 3-5**). To edit the abscissa signal source, edit the source signal and restart sequence computing (see **section 3.1.2**).

3.2.2. Select / Unselect sequences

Once sequencing has been performed in the analyze layout, all sequences can be displayed in the “Analyze Chart” panel by clicking lines in the “Sequence Explorer” panel or by clicking |< and >| buttons in the “Analyze Controls” panel (see **Figure 3-1**). By using |< and >| buttons, user can switch between all sequences of the current file; however, user must click on any sequence line of a specific file to navigate to another file.

User might want to select or unselect some sequences to display/process. To do so, tick/untick checkboxes corresponding to sequences to keep/ignore (see **Figure 3-2**). Only sequences ticked are displayed and processed (see ***{ROOT}/resources/videos/analyze/16.mp4***).

3.2.3. Set sequence visualization options

Sequence View. Sequences can be displayed one-by-one by ticking ● Single checkbox in the "Analyze Controls" panel (see **Figure 3-1**). To superimpose all selected sequences of the current file, tick ● Multiple checkbox in the "Analyze Controls" panel. Note that, in ● Multiple view mode, the current sequence is underlined with a specific color compared to other sequences of the current file. To calculate the average sequence of all selected sequences, tick ● Average checkbox in the "Analyze Controls" panel. In ● Average view mode, all sequences are superimposed with the averaged sequence, which is underlined with a specific color.

Editing view mode affect both the way sequences are displayed in chart and the way sequences are processed. Indeed, only the current sequence is processed in ● Single view mode, only the averaged sequence is processed in ● Average view mode, and all selected sequences are processed in ● Multiple view mode (see **section 3.3.2**).

Axes limits. Both abscissa and ordinate limits can be customized using dedicated edit boxes at each side of X-axis and Y-axis in the "Analyze Chart" panel. Note that editing X-axis and Y-axis limits only affect the way sequences are displayed, but not the way they are processed (to do so, edit sequencing options in "Sequencing" panel).

If user leaves one of the X-axis or Y-axis edit boxes empty, the software automatically adjusts corresponding axis limits. In the opposite case, the defined axis limits are applied.

Axis colors. As for live layout, "Analyze Chart" panel background color can be adjusted by right clicking anywhere in the panel and selecting "Background Color" in the menu that pops up. In the same way, user can adjust line colors, by selecting "Line Colors > Main Line", for current or average sequence (depending of the view option select; see **section 3.2.2**), or "Line Colors > Secondary Line(s)", for the superimposed sequences, in the same menu.

3.2.4. Apply signal treat on sequences

User has the possibility to apply a treatment to signal sequences by selecting a user function from ***{ROOT}/userFcn/signalTreats/*** folder in the butterLowFilter_5.m ⌄ scroll-down menu in the "Analyze Controls" panel (see **Figure 3-1** & ***{ROOT}/resources/videos/analyze/17.mp4***). This way, sequences are pre-processed for both display and process purposes. However, user can rather choose to apply signal treats in ***{ROOT}/userFcn/processFcns/*** user functions, which would not affect how sequences are displayed in the "Analyze Chart" panel.

User should note that signal treats are only applied sequence-by-sequence after sequencing, and not on the entire source signal before sequencing (see section 4.6).

3.3. Process data

The software analyze mode aims at applying custom user functions to a data sequence or a series of data sequences (i.e., multiple view mode) so that extracting specific results from the analyzed sequences (see ***{ROOT}/resources/videos/analyze/18.mp4*** & ***{ROOT}/resources/videos/analyze/19.mp4***). Therefore, user must create user functions and store them in ***{ROOT}/userFcn/processFcns/*** folder to be able processing imported sequences (see section 4.3).

3.3.1. Associate / Disassociate user functions

To use a processing function from ***{ROOT}/userFcn/processFcns/*** folder, the target user function should firstly be associated to a N/A button of the "Analyze Controls" panel. To associate a user function to a N/A button, right click on the N/A button, and select "Associate Function" in the menu that pops up. Then, select a user function among all functions stored in ***{ROOT}/userFcn/processFcns/*** folder. After having selected a function, N/A button is labeled by the selected function name (e.g., maxSequences if ***{ROOT}/userFcn/processFcns/maxSequences.m*** is the selected user function).

The software enables associating six processing user functions at a time. However, user can disassociate user functions or replace user functions associated to a button of the "Analyze Controls" panel, by right clicking on the target button and selecting "Clear Function" or "Associate Function" menus, respectively.

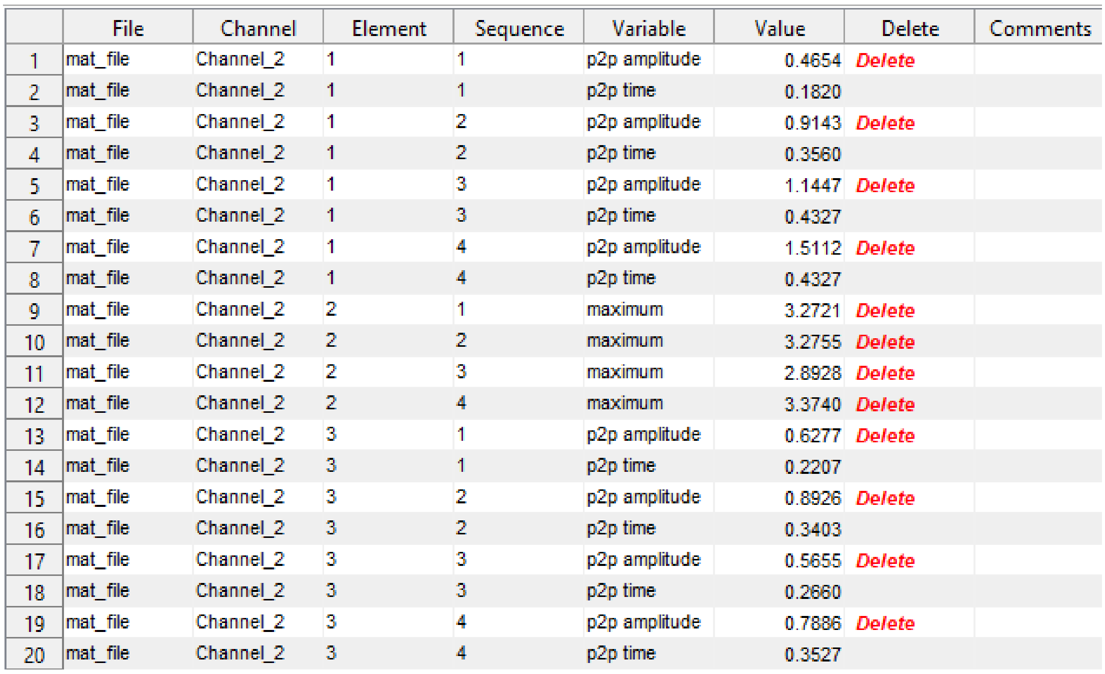
3.3.2. Run user functions

Once a processing user function has been associated to a processing button of the "Analyze Controls" panel, user is ready to process data sequences. To launch a processing user function, click on the associated button (e.g., maxSequences to run ***{ROOT}/userFcn/processFcns/maxSequences.m***). User should note that this action sends one single data sequence as an argument to the associated user function if sequence view option is ● Single or ● Average, and sends all selected sequences of the current file as arguments to the associated user function if sequence view option is ● Multiple (see **section 4.3**).

3.4. Review processing

Every time a processing user function is called – i.e., every time a processing button is clicked – results of the function are displayed in the "Result Explorer" panel (see **Figure 3-7**). If sequence view option is ● Multiple when running processing user function, all selected sequences of the current file are processed and associated results are displayed in the "Results Explorer" panel. Further, if processing user function is written to save multiple variables for each sequence, then all variables are listed in the "Results Explorer" panel (see *'p2p amplitude'* and *'p2p time'* in **Figure 3-8**).

Since user might want to apply different processing user functions (or the same processing user function several times) to the same data sequence, results are associated to a new element count in the "Results Explorer" panel every time a new function call is made for a specific sequence (see 'Element' column in **Figure 3-7**).



	File	Channel	Element	Sequence	Variable	Value	Delete	Comments
1	mat_file	Channel_2	1	1	p2p amplitude	0.4654	Delete	
2	mat_file	Channel_2	1	1	p2p time	0.1820		
3	mat_file	Channel_2	1	2	p2p amplitude	0.9143	Delete	
4	mat_file	Channel_2	1	2	p2p time	0.3560		
5	mat_file	Channel_2	1	3	p2p amplitude	1.1447	Delete	
6	mat_file	Channel_2	1	3	p2p time	0.4327		
7	mat_file	Channel_2	1	4	p2p amplitude	1.5112	Delete	
8	mat_file	Channel_2	1	4	p2p time	0.4327		
9	mat_file	Channel_2	2	1	maximum	3.2721	Delete	
10	mat_file	Channel_2	2	2	maximum	3.2755	Delete	
11	mat_file	Channel_2	2	3	maximum	2.8928	Delete	
12	mat_file	Channel_2	2	4	maximum	3.3740	Delete	
13	mat_file	Channel_2	3	1	p2p amplitude	0.6277	Delete	
14	mat_file	Channel_2	3	1	p2p time	0.2207		
15	mat_file	Channel_2	3	2	p2p amplitude	0.8926	Delete	
16	mat_file	Channel_2	3	2	p2p time	0.3403		
17	mat_file	Channel_2	3	3	p2p amplitude	0.5655	Delete	
18	mat_file	Channel_2	3	3	p2p time	0.2660		
19	mat_file	Channel_2	3	4	p2p amplitude	0.7886	Delete	
20	mat_file	Channel_2	3	4	p2p time	0.3527		

Figure 3-7. *View of "Results Explorer" panel after having processed data sequences using a processing user function.*

Processing user functions enables associating chart elements (optional) to results (see section 4.3). If any chart element is associated to the current sequence, it is displayed in the "Analyze Chart" panel (see **Figure 3-8**).

To delete result lines in the "Results Explorer" panel, user should click the **Delete** text on lines. User should note that this action deletes all variables of the corresponding file, sequence and element, together with associated chart element(s) in the "Analyze Chart" panel.

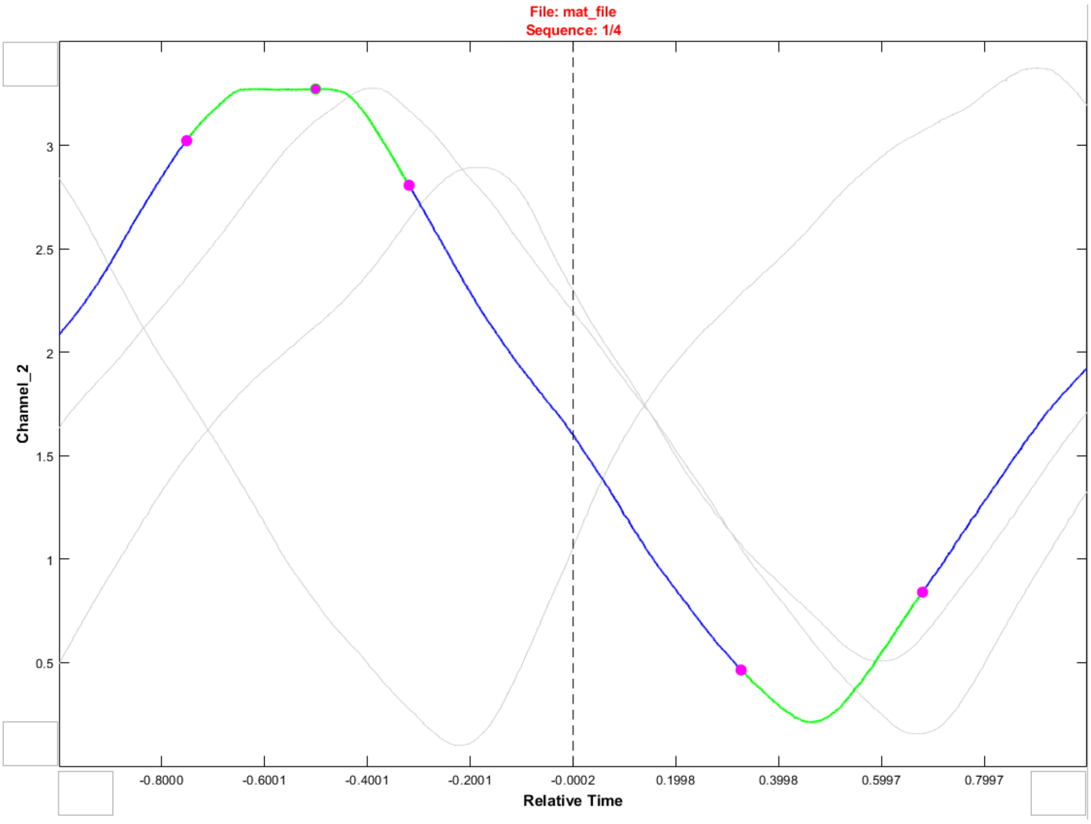


Figure 3-8. *View of "Analyze Chart" panel associated with results displayed in the "Results Explorer" panel of* ***Figure 3-7****.*

3.5. Export results

Data of the “Results Explorer” panel can be exported at any time to a **‘.mat’** or **‘.csv’** file. To do so, select “Analyze > Export Results” menu and browse to the preferred path to save the results. Whatever the output format of the saved file (i.e., **‘.mat’** or **‘.csv’**), results are saved exactly as they appear in the “Results Explorer” panel.

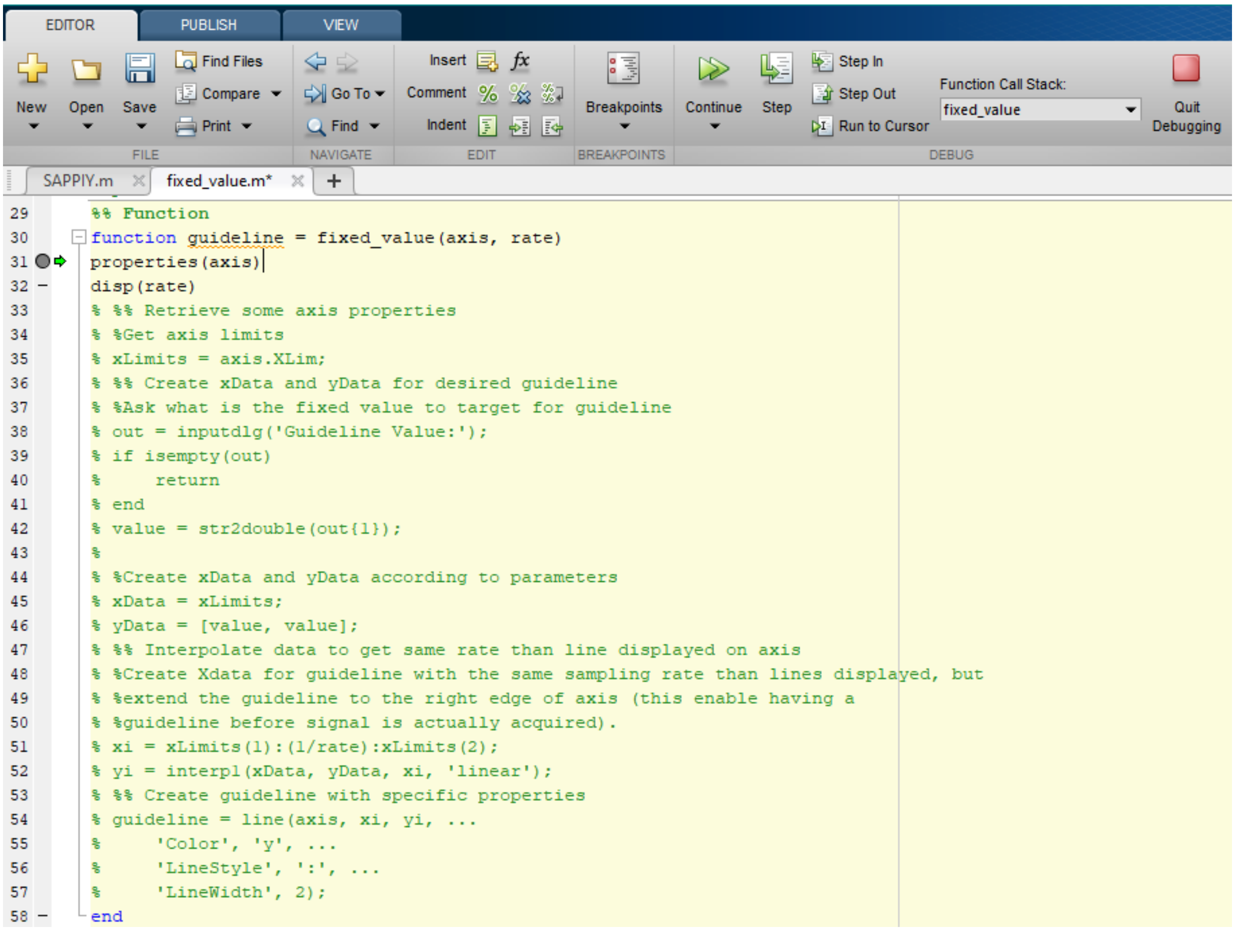
User can also export “Analyze Chart” panel at any time by right clicking on this panel and selecting “Export Chart” in the menu that pops up. The “Analyze Chart” panel is saved as **‘.pdf’** file.

4. User functions formatting

This chapter describes input argument types as well as output argument requirements for user functions from all ***{ROOT}/userFcn/*** subfolders. When user intends to write its own user function, it is recommended to read attentively this chapter as well as some user function examples of the same type (i.e., from the same subfolder). Another advice is to copy and paste header instructions of a preexistent script of the same type in each new user function to conserve creation guidelines through all created files.

The best way to understand how to create its own user function is to explore input arguments first. To do so, user should comment all content of the function that it intends to create[6], mark a "pause breakpoint" immediately after the function declaration, and execute the function (see 'Execution' section in consecutive sections) to enter in debugging mode. From there, to explore input arguments, different Matlab® commands can be used, and results would be displayed in the Matlab® command window. It is recommended to use *'properties(arg)'* function for input arguments of Matlab® objects type to get all accessible properties associated with the input Matlab® object, and use *'disp(arg)'* function for vector or array input arguments to explore its values (see **Figure 4-1**).

Once user has a good overview of how input arguments are structured, it can start creating its own function from the debugging mode (recommended) or in offline mode.



```
29  %% Function
30  function guideline = fixed_value(axis, rate)
31  properties(axis)
32  disp(rate)
33  % %% Retrieve some axis properties
34  % %Get axis limits
35  % xLimits = axis.XLim;
36  % %% Create xData and yData for desired guideline
37  % %Ask what is the fixed value to target for guideline
38  % out = inputdlg('Guideline Value:');
39  % if isempty(out)
40  %     return
41  % end
42  % value = str2double(out{1});
43  %
44  % %Create xData and yData according to parameters
45  % xData = xLimits;
46  % yData = [value, value];
47  % %% Interpolate data to get same rate than line displayed on axis
48  % %Create Xdata for guideline with the same sampling rate than lines displayed, but
49  % %extend the guideline to the right edge of axis (this enable having a
50  % %guideline before signal is actually acquired).
51  % xi = xLimits(1):(1/rate):xLimits(2);
52  % yi = interp1(xData, yData, xi, 'linear');
53  % %% Create guideline with specific properties
54  % guideline = line(axis, xi, yi, ...
55  %     'Color', 'y', ...
56  %     'LineStyle', ':', ...
57  %     'LineWidth', 2);
58  end
```

Figure 4-1. *A guideline user function in debugging mode. Lines 31 and 32 of the script being debugged show how to explore a Matlab® objet (i.e., 'properties(arg)' function) and vector or array variable types (i.e., 'disp(arg)' function), respectively.*

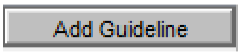
[6] Since the function being created must not contain any error to enter in debugging mode, it is recommended to comment all script content, but this is not a requirement.

4.1. Guidelines

Function:

User functions from ***{ROOT}/userFcn/guidelines/*** subfolder are used to create a guideline that is displayed on live chart while acquiring data.

Execution:

This user function type is executed by clicking  button in the “Analog Channels” panel of live layout (see **section 2.3.3**).

Input Arguments:

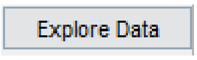
Argument	**Type**	**Description**
‘axis’	Axis Matlab® object	Axis of the “Live Chart” panel. *This axis contains several information that can be useful for guideline creation. Examples already present in the* ***{ROOT}/userFcn/guidelines/*** *subfolder mainly deal with ‘XLim’ axis property.*
‘rate’	1-by-1 double value	Rate of the live session. *The rate information in examples already present in the* ***{ROOT}/userFcn/guidelines/*** *subfolder is mainly used to interpolate guideline to the same sampling rate than analog channel lines (this is a requirement; see output arguments).*

Output Argument:

Argument	**Type**	**Description**
‘guideline’	Line Matlab® object	Line to display on axis of the “Live Chart” panel. *The guideline to display must have the same sampling rate than analog channel lines of axis of the “Live Chart” panel. This is required since the internal function that updates axis display automatically shifts all lines of the axis to the same sample count. However, this is not a requirement to display a guideline from side to side of the axis (i.e., ‘XData’ property can be of different dimension between analog channel lines and guideline objects).*

4.2. PreviewTreats

Function:

User functions from ***{ROOT}/userFcn/previewTreats/*** subfolder are used to process a source signal in the live data preview window that pops up after clicking  button in live layout (see **section 2.3.5**). These functions only deal with the signal part that is visible on the source axis at function execution, not the entire signal acquired (unless the entire signal is visible).

Execution:

This user function type is executed when selecting a user function in the scroll-down menu at the top left corner of the live data preview window (see **section 2.3.5**).

Input Arguments:

Argument	Type	Description
'source'	Axis Matlab® object	Sub-axis of the live data preview window for which a user function has been called. *This axis contains several information that can be useful for display purposes. Examples already present in the **{ROOT}/userFcn/previewTreats/** subfolder mainly deal with source to focus all elements drawn to the proper source.*
'x'	Double 1D vector	'XData' property of the line part visible in source axis.
'y'	Double 1D vector	'YData' property of the line part visible in source axis. 'x' and 'y' are useful to analyze parameters on signal in abscissa and ordinate dimension, respectively.
'rate'	1-by-1 double value	Rate at which line of source axis is sampled.

Output Argument:

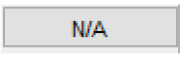
No output argument permits

4.3. ProcessFcns

Function:

User functions from ***{ROOT}/userFcn/processFcns/*** subfolder are used to process data sequences from software analyze mode.

Execution:

This user function type is executed by clicking a  processing buttons in the “Analyze Controls” panel once it has been associated with a processing user function (see **section 3.3.1**).

Input Arguments:

Argument	Type	Description
‘axis’	Axis Matlab® object	Axis of the “Analyze Chart” panel. *This axis argument contains chart elements (e.g., data sequences lines) but also several information that can be useful for data sequence processing or for associating chart elements to processed results. Further, ‘p2pTimeSequences.m’ example in the **{ROOT}/userFcn/processFcns/** subfolder also deals with axis interactive control callbacks to manually draw elements on axis while running processing user function.*
‘data’	structure	‘.xData’: contains XData of all sequences to process. ‘.sequences’: contains YData of all sequences to process. ‘.rate’: contains rate of sequences to process. *The data structure contains most information required to process data.*
‘results’	cell array	Prefilled results cell array to fill and add to the “Results Explorer” panel as output argument. *The two last columns of the prefilled results cell array are empty, and must be filled with variable label and variable value columns. Prefilled results cell array is built to store only 1 variable for each sequence. If user wants to store multiple variables, it should duplicate all results cell array rows as much time as variable to add.*
‘callbacks’	function handles	Internal function callbacks used to update results in the “Results Explorer” panel and chart elements drawn on axis in the “Analyze Chart” panel. *These function handles are automatically called after the processing function is executed, so user does not need to use them for “direct” process. However, when user associates an edit function to chart elements, as it is performed for ‘p2pTimeSequences.m’ example in the **{ROOT}/userFcn/processFcns/** subfolder, then these callback can be useful to update both “Results Explorer” and “Analyze Chart” panels.*

Output Argument:

Argument	**Type**	**Description**
‘results’	cell array	Filled results cell array. *The results cell array must contain 6 columns, the 4 firsts prefilled as input argument, and the 2 lasts filled with variable labels and values in processing user function.*
‘newElements’	cell array	Elements associated to lines of results, which should be drawn on live axis when corresponding file, channel and sequence is displayed. *newElements variable must have the same number of row than results output argument. Each column represents a new element to associate with results variable of associated row. Each element must be a structure with at least a field labeled 'type', which define object type (e.g., line, bar, area, surf). The other element fields must be valid properties of associated object type (e.g., ‘XData’, ‘Color’, ‘MarkerSize’ for line objects).*

4.4. Read_ttl

Function:

User functions from ***{ROOT}/userFcn/read_ttl/*** subfolder are used to detect TTL indexes in a source signal defined by user.

Execution:

This user function type is executed when an imported file does not already contain TTL information (see **section 3.1.2**).

Input Arguments:

Argument	**Type**	**Description**
‘signal’	1D double vector	Source signal in which look for TTL indexes.

Output Argument:

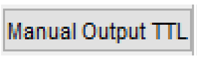
Argument	**Type**	**Description**
‘ttl’	1D double vector	TTL indexes detected in the source signal.

4.5. Send_ttl

Function:

User functions from ***{ROOT}/userFcn/send_ttl/*** subfolder are used to send output TTL from Digital Output object of the data acquisition hardware in use in the software live mode.

Execution:

This user function type is executed either when user click  button in the "Live Controllers" panel (see **section 2.4.4**) or when analog-based output TTL condition is reached (see **section 2.4.5**).

Input Arguments:

Argument	**Type**	**Description**
'DOsession'	Session Matlab® object	Data acquisition session object controlling Digital Output object. *User should explore Data Acquisition Matlab® documentation for more details about session object use, or call 'methods(DOsession') & 'properties(DOsession)' commands in debugging mode to explore all possibilities offer to control Digital Object.*
'channel'	1-by-1 double value	Channel index of the Data acquisition session object to deal with. *This channel corresponds to the port selected in the "TTL Settings" live panel.*

Output Argument:

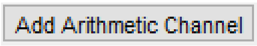
No output argument permits

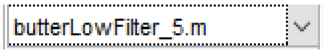
4.6. SignalTreats

Function:

User functions from ***{ROOT}/userFcn/signalTreats/*** subfolder are used to apply some treats to different signal sources: either in live mode, for arithmetic channels, or in analyze mode, for data sequence.

Execution:

This user function type is executed (i) when acquiring live data if a user function of this type has been declared in the equation while adding arithmetic channel with 

button in "Analog Channels" panel of live mode (see **section 2.2.2**) and (ii) when selecting a signal treat function in the dedicated scroll-down menu 

in the "Analyze Controls" panel (see **section 3.2.4**).

Input Arguments:

Argument	**Type**	**Description**
'inSignal'	1D double vector	Source signal to treat.
'rate'	1-by-1 double value	Rate at which source signal is sampled.

Output Argument:

Argument	**Type**	**Description**
'outSignal'	1D double vector	Signal treated.