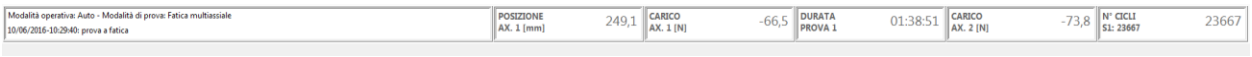
TESTCENTER SOFTWARE USER GUIDE (version 7.3)

Table of contents

The testing machine interface software is structured in four environments: manual handling, alarm management, test mode and post-processing. At startup the manual handling page is displayed, and in order to navigate between the available pages the user just needs to click on the dedicated names on the tab control (except for the post-processing environment, which is launched by a button of the upper bar).

1. Upper and lower sections

In all environments but post-processing, on the upper and lower parts of the user interface (from now on also abbreviated in UI) some buttons and indicators are always visible. This paragraph is dedicated to explain their functioning:

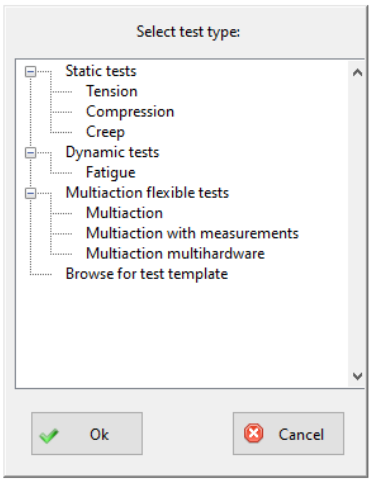


On the left side of the lower bar a field is reserved for messages regarding the operative mode, the ongoing action, the reason of an eventual test end condition, and any other kind of information from the program to the user. On the right instead there is a variety of numeric indicators that can be configured by the user both in terms of how many to be displayed and of which quantity has to be displayed in order to monitor the most significant system parameters in real time; such parameters can be chosen among some general ones (such as load and actuator position) and some others that are specific for the loaded test (for example, the cycle counter); paragraph 5 describes the configuration of the lower bar. Finally, in the grey section on the bottom of the page, the user profile is displayed on the right side (see paragraph 8.1) and a string related to the last alarm occurred can be seen on the left side in case the machine is currently in alarm state.

As for the upper bar, it is composed of seven buttons:

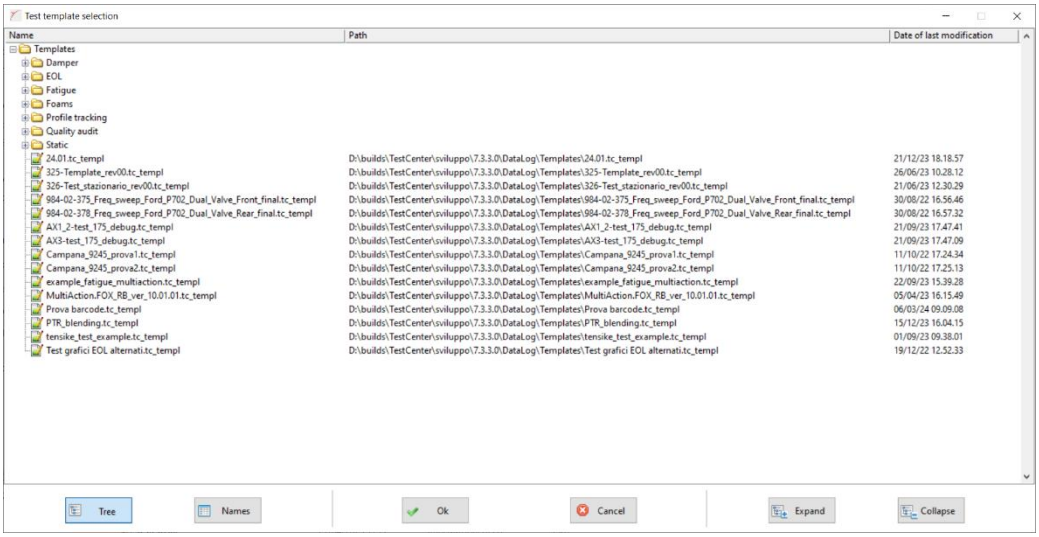
NEW TEST: creates a new test of the selected type after the user chooses among the protocols that are available in the software version (see description of the implemented tests in the following paragraphs).

When the user selects this command, the following interface is displayed to choose among the available test protocols:



Please note that the available choices depend on the machine type and software configuration.

The selection is done by clicking on one of the options and then on the OK button, or by double-clicking on the desired option. Also, the final option (*browse for test template*) opens an additional window through which the user can navigate the computer directories to search for a particular test template file:



The software automatically searches for the available test template files in a pre-configured set of folders, and presents the whole list in a tree menu; the user can load the desired test template by double-clicking on its name.

LOAD TEST: loads a previously saves test configuration file (tests are generally saved by default in the *DataLog – Saves* folder, but it is possible for the user to rename and move the files through the computer); it is possible to load also test files that do not correspond to the test protocol that is currently

loaded (for instance, the software is in *creep* test mode and the user wants to load a *fatigue* test file): in this case, the active test protocol is automatically adapted to the chosen file. While loading a test it is possible to choose between a normal test file and a template: if the first one is selected, the original document (i.e. the test configuration file) is loaded and eventually modified, whereas, if a template is chosen, when the file is loaded the program creates a new test document populated with all the settings included in the test template. In this way, any further change made by the user does not influence the reference template, which is therefore reusable endlessly (see *File – Rename test*, paragraph 5).

RUN: starts the test with the current parameters.

STOP: stops a test that is running.

ADD SPECIMEN: adds a specimen within the current test; on the left side of the window, in the test configuration environment, there is a listbox collecting all the specimens created, naming them with numbers (from 1 incrementing) or with custom names that can be chosen by the user. The parameters of all the specimens created in the same test are gathered in a unique configuration file.

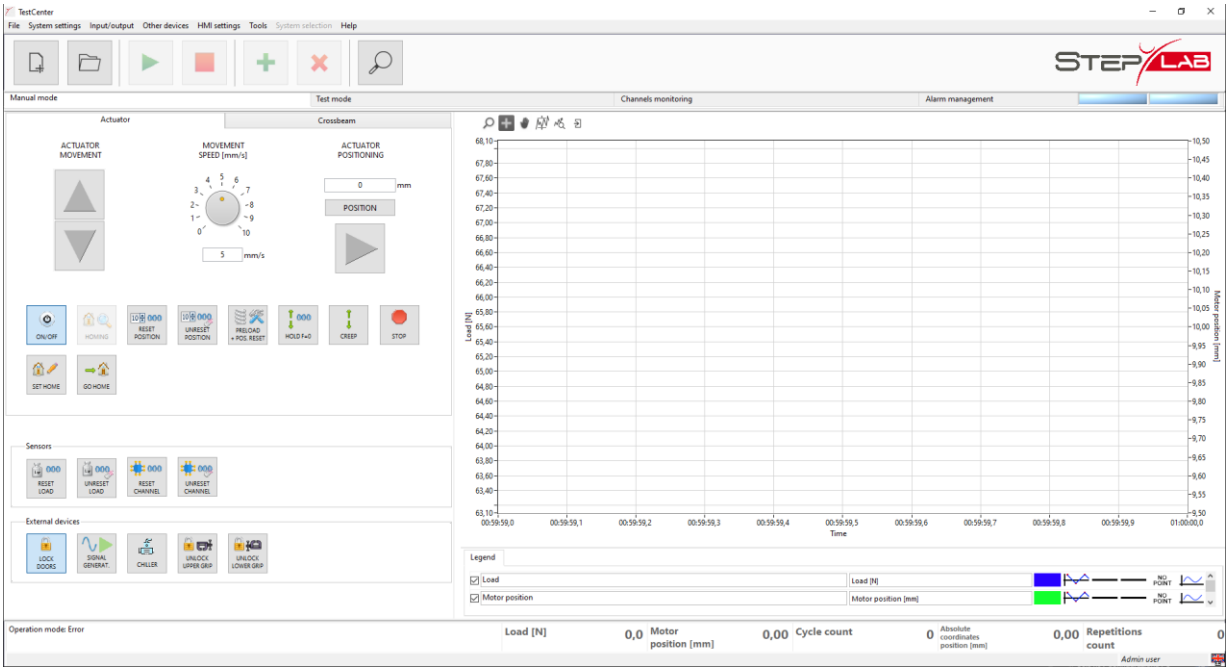
DELETE CURRENT SPECIMEN: if confirmed, permanently deletes the selected specimen from the configuration file and its results – if present – from the log file; these data can't be recovered after executing the delete command. For static tests the results group of the selected specimen is deleted from the log file, whereas for dynamic tests the entire file is cancelled.

TDMS VIEWER: opens the post-processing environment, also called *TDMS viewer* or simply *viewer*, which allows to analyse the results of all the test performed by loading the dedicated files and displaying data on graphs and tables; it is also possible to export data.

When one or more of these buttons are greyed out, the associated commands are disabled in that particular moment of the execution of the program: for example, the *Stop* command is always disabled unless a test is running.

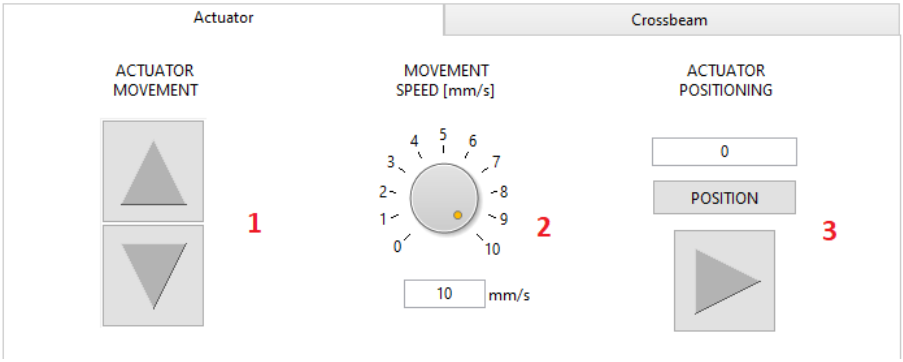
2. Manual handling environment

The interface appears as in the following picture:



Note: the state of the upper bar buttons (see paragraph 1), of the alarm and warning indicators (see paragraph 3), of the lower bar (see paragraph 1) and the number of visible buttons for the axis are not strictly related or bound to the displayed window.

In the region of the page located between the upper and the lower bar described before, there are a real-time chart to constantly monitor load and actuator position on the right, and a group of commands on the left.



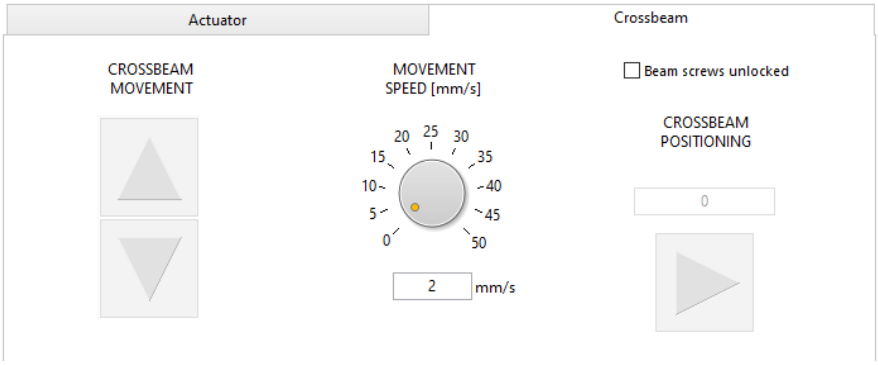
ACTUATOR MOVEMENT: the arrow buttons oriented upwards and downwards (1) move the axis in the desired directions, exactly as the movement buttons on the electric cabinet of the machine do; these arrow buttons can be possibly oriented horizontally or displaying rotational arrows (clockwise and

counterclockwise), depending on the mechanical layout of the actuator; the dial on the centre and the associated numeric control (2) allow to vary the axis manual movement speed within the range shown; finally, on the right of the interface there is a control to execute position-controlled displacements up to the inserted coordinate (3): the target position or displacement value has to be written in millimetres (if the actuator has a linear motion) or degrees (if rotational), and when the button representing an arrow pointing right is pressed the axis starts moving; the control showing *DISPLACEMENT* and *POSITION* options is needed to specify whether the value written on the numeric control has to be interpreted by the software as a displacement with respect to the current position (*DISPLACEMENT* option) or as a coordinate to reach (*POSITION* option); in particular, in the latter case the user reference system has to be considered to set the position instead of the machine absolute reference system, therefore if the user resets the position at a certain coordinate before executing the movement, the value set is managed with respect to the new coordinate system.

Examples:

- The absolute machine reference system is used, and a positioning is executed up to 150 mm, from the starting coordinate of 100 mm, with the *POSITION* control option: the machine reaches 150 mm position in its own absolute coordinate system;
- The absolute machine reference system is used, and a positioning is executed up to -7 mm, from the starting coordinate of 100 mm, with the *DISPLACEMENT* control option: the machine reaches 93 mm position in its own absolute coordinate system, moving of 7 mm from the starting position, in the negative direction, as set by the user;
- The position is reset to 0 when the actuator is located at 100 mm in its absolute coordinate system, then the machine is moved manually until the position indicator displays 10 mm and from this point a positioning is executed up to 45 mm with the *POSITION* control option: the machine reaches 45 mm position in the new user-selected coordinate system, which means its position in the absolute coordinate system is equal to 145 mm;
- The position is reset to 0 when the actuator is located at 100 mm in its absolute coordinate system, then the machine is moved manually until the position indicator displays 10 mm and from this point a positioning is executed up to 45 mm with the *DISPLACEMENT* control option: the machine reaches 55 mm position in the new user-selected coordinate system, which means its position in the absolute coordinate system is equal to 155 mm.

Caution: *the positioning functionality must be used with special care, because when the actuator is moving its load limit is the maximum possible, so that it would easily break any object representing an obstacle to the movement towards the target position, or damage itself if the object is resistant enough not to be broken.*

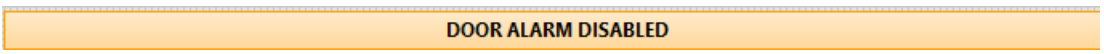


CROSSBEAM MOVEMENT: this tab is only available if the machine is equipped with an electric crossbeam, otherwise it is not displayed; also, not all the crossbeam types support the speed regulation and the positioning functionalities, so these may be permanently disabled: please check the technical specifications of the machine for these details (the software is configured consequently).

This section follows the same principles of the manual actuator movement, but the "beam screws unlocked" checkbox on the top-right corner has to be active in order to move the crossbeam from the TestCenter: depending on the machine technical specification, the checkbox may have to be turned on and off manually (if there are no sensors to check the screws state automatically) or may be updated automatically by the software if screws sensors are installed.

When the "beam screws unlocked" checkbox is checked it is possible to move the crossbeam but it is not possible to start automatic procedures of the actuator, and vice versa.

Caution: *if there are no sensors to detect the crossbeam locking, the checkbox state must be kept up to date with the actual screws condition, and the crossbeam must not be commanded to move if the fastening screws are locked, otherwise the electric motor that activates this mechanism will be damaged; also, when the beam is being moved it is necessary to constantly and carefully verify to avoid collisions between the various parts of the machine in order to prevent damages; as it is obliged for manual handling of the axis, the stroke has to be completely free when moving the beam.*



DOOR ALARM DISABLED: if this warning is displayed just below the actuator/crossbeam movement tab control, it means that the door alarm is currently disabled; in other words, if safety doors are opened no alarm occurs. This condition is needed when setting up a specimen for a test, but no automatic operation can be started when the door alarm is deactivated, except the "hold F=0" procedure.

The buttons located below the command described before control the following functionalities. Please note that the various buttons may be permanently disabled depending on the machine hardware configuration.

ACTUATOR section:

ON/OFF: turns the motor on or off.

HOMING: this button is enabled only if the axis has a relative encoder (that sets the position to zero wherever they are at the moment of powering it on, whereas an absolute encoder preserves its own coordinate system even if the machine is turned off and then restarted), and executes a procedure through which the machine searches for one of the limit switches at the bounds of the available stroke in order to set the maximum or minimum position value and obtain an absolute coordinate system which is valid until the machine is turned off; when this operation is commanded a dialog window requires the user to choose the sensor – upper or lower – that will be searched; then, another dialog window recommends the user to remove any element that may interfere with the movement from the current position to the limit switch, giving the possibility to abort the operation and remove obstacles if there is any. In fact, it needs to be pointed out that the homing procedure, as manual move and positioning, has the maximum load limits of the machine as the only ones, therefore anything placed between the starting and the target position may be broken or possibly cause damages to the machine.

RESET POSITION: opens a window where it is possible to choose a new coordinate that will be assigned to the current position, creating a new reference system with respect to which data will be displayed and saved from that moment until an *unreset* is commanded.

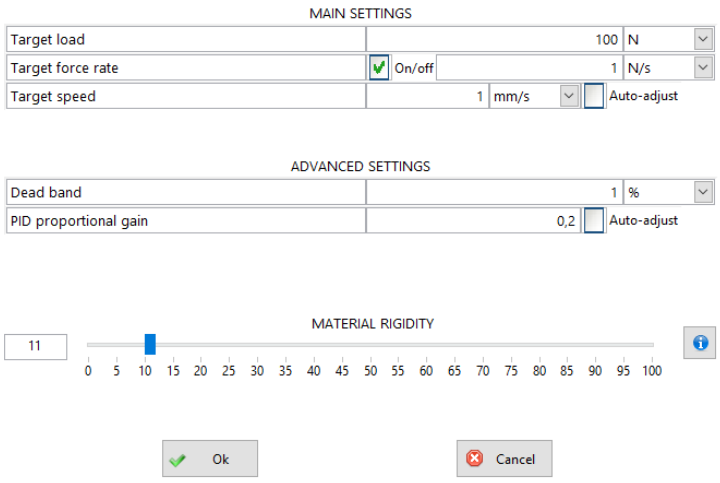
UNRESET POSITION: eliminates the position reset effect, recovering the value displayed in the absolute machine coordinate system, cancelling any offset.

PRELOAD + POS. RESET: opens a dialog window where the user can select a load and a position value: the latter is the position that will be given through an automatic reset operation to the coordinate where the machine reaches the specified load.

HOLD F=0: starts an automatic procedure through which the actuator slowly keeps moving and constantly aims at maintaining the load as close as possible to zero; this operation can be stopped pressing the button while active or giving the STOP command.

> ***Caution:*** *if this feature is accidentally activated without anything being clamped by the grips, the actuator will start to move in a direction that depends on the value of the load measured at that moment; this could cause collisions between the grips or other components of the system, and consequently damage; it is the responsibility of the user to ensure that the "hold F=0" procedure is only activated when a specimen is clamped inside the grips.*

CREEP: this functionality is similar to "HOLD F=0", in the sense that commands the actuator to reach a certain load and hold it indefinitely; however, in this case the user has the possibility to configure some parameters through the following interface that is displayed as soon as the button is pressed:



- Target load: it is the target value of the load to be reached and held; it has to be configured with the appropriate sign;

- Target force rate: if enabled, it is the reference value for the load rate during the movement towards the desired load value;
- Target speed: this is the initial movement speed, that is progressively decreased by the controller while the load gets closer to the target value;
- Dead band: it is the percentage value that sets when the tolerance to take the target load for reached;
- PID proportional gain: proportional gain of the controller during this operation
- MATERIAL RIGIDITY slider: this cursor allows the user to configure values for *target speed* and *PID proportional gain* automatically depending on the rigidity of the material, if the respective *Auto-adjust* checkboxes are activated.

STOP: stops any operation running on the actuator.

SET HOME: sets the current position of the actuator as the reference coordinate to which it will be returned if the next "GO HOME" button is used; the storage of the "home" position is independent of the coordinate system used, as it is based on the absolute, invariant coordinate system of the actuator; furthermore, the position value remains in the memory even if both the TestCenter software and the control cabinet are restarted.

GO HOME: upon confirmation by the user via a window which opens when the button is pressed, starts a movement of the actuator which returns it to the position previously stored through the "SET HOME" function.

SENSORS section:

RESET LOAD: resets the load assigning 0 to the current value.

UNRESET LOAD: eliminates the load reset effect, recovering the original value without any offset.

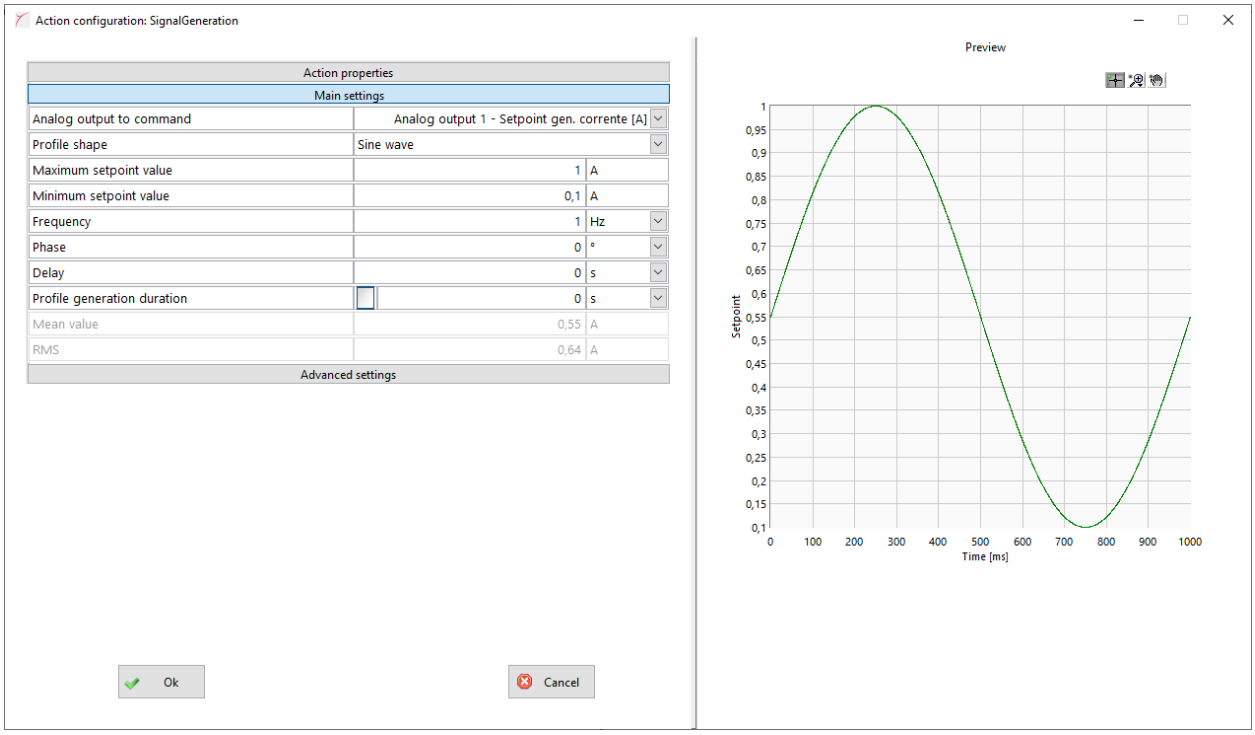
RESET CHANNEL: allows the user to reset one or more analog channels at the same time in the same way as it can be done to the load signal through the *RESET LOAD* functionality.

UNRESET CHANNEL: clears the reset to one or more analog channels at the same time, recovering the original value without any offset.

EXTERNAL DEVICES section:

LOCK-UNLOCK DOORS: this button is enabled only if the testing system is equipped with an electric door lock and allows the user to switch between the locked and unlocked conditions.

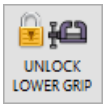
SIGNAL GENERATION: if enabled, activates a manual procedure to command an external analog output signal generator:



This interface is dedicated to command an external device that can generate an electric signal and is the same that configures a *SignalGeneration* action, so the detailed explanation of the configuration parameters can be located at paragraph h4.4.17. In this case, differently, from what happens for the microaction, as soon as the user presses the OK button the system starts commanding the device to generate the current signal; in order to stop a manual signal generation procedure that is running it is necessary to open the interface, select the *stop* option of the *main settings* and press OK.

CHILLER: if the machine is provided with a liquid cooling system, this button turns it on and off manually.

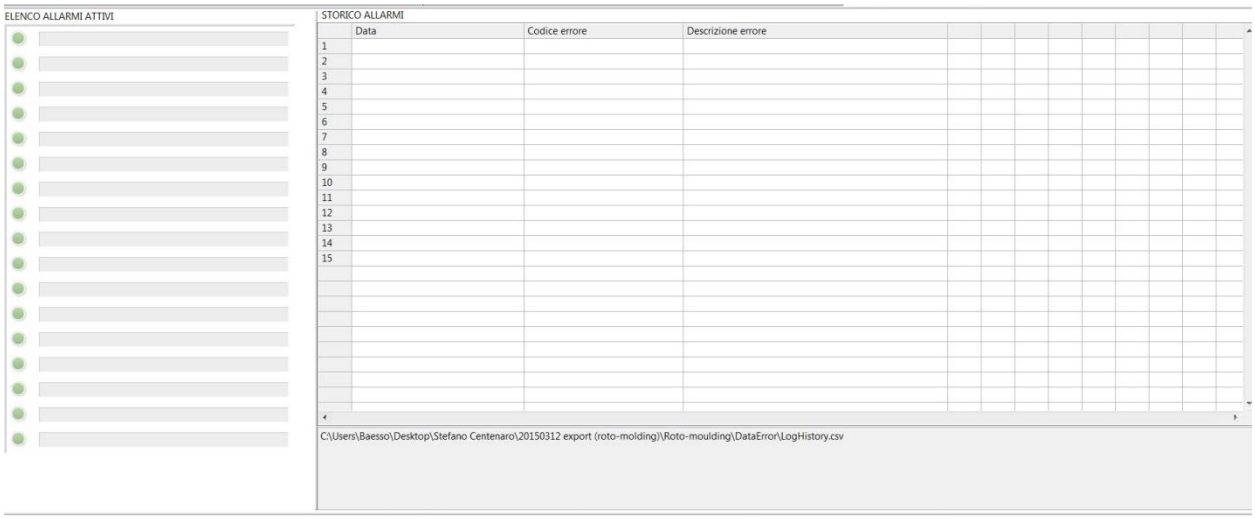
LOCK-UNLOCK UPPER GRIP: if the machine is equipped with an automatic gripper system, this button commands the upper grip opening and closing.



LOCK-UNLOCK LOWER GRIP: if the machine is equipped with an automatic gripper system, this button commands the lower grip opening and closing.

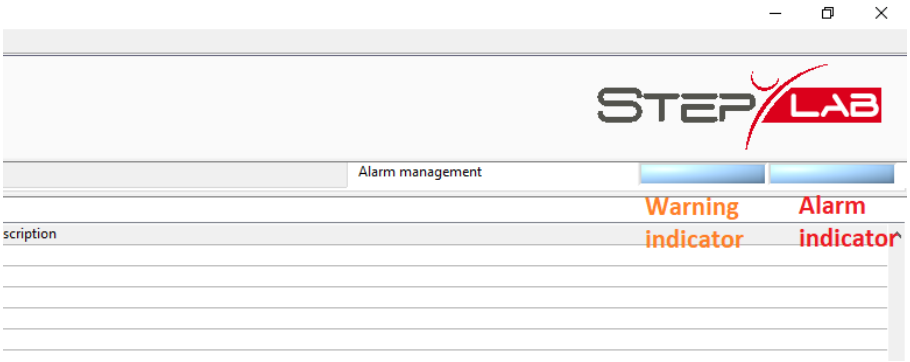
3. Alarm management environment

The interface (visible in the picture below) allows the operator to view the active alarms on the left side of the window, and the alarm history (which is also saved on text files) on the right.



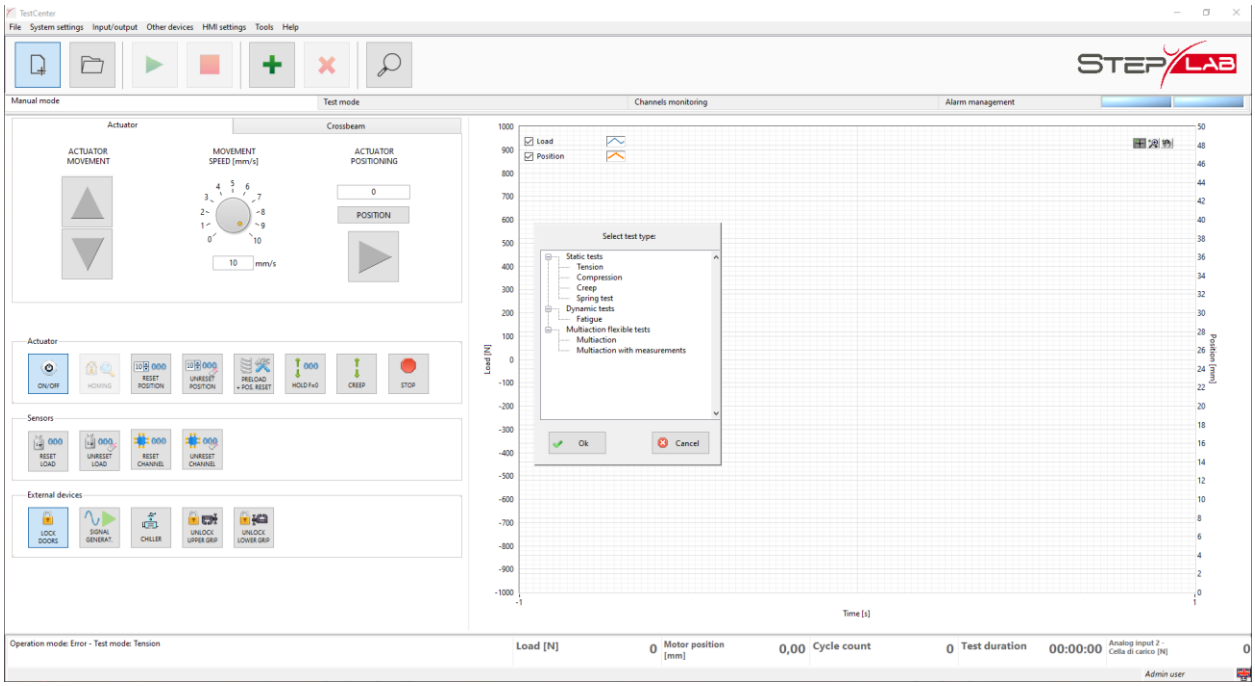
When another environment of the program is loaded, it is possible to constantly monitor whether there are active alarms or not through the two rectangular indicators next to the “Alarm management” tab title on the main tab control: if the machine is in emergency state the indicator on the right becomes red and it is not possible to start any automatic operation; if there is not any active alarm, the indicator is light blue. Instead, the indicator on the left becomes orange when there are active warnings, which are less critical than alarms but anyway conditions requiring attention by the user and preventing the execution of a test.

Both active alarms and alarms stored in the history are provided with a short description of what caused the emergency state; in this page, warnings and alarms are displayed in the same way.



4. Test environment

This is the most complex environment and the centre of the software, because it is where all the tests performed through the machine are configured and monitored. In order to open a new test the user has to click on the dedicated button on the upper bar (the first on the left), and the dialog box which is shown allows to select the desired test type.



For each test there is a tab control to navigate between the test parameters setup interface and some graphs; all the test setup menus are grouped into sections to make the configuration process easier; in order to display a menu, just click on the button displaying its title. Instead, graphs are different between static and dynamic tests.

4.1. Tension/compression test

The parameters to be set are:

GENERAL SETTINGS

Target speed: the translation speed of the axis during the test.

TEST END CONDITIONS

Target load: force to reach in the test (does not take into account the initial load, and has to be written with its sign);

Target position: if the “Position control mode” parameter (advanced settings) is set to *position* this setting represents the limit coordinate to stop the test; instead, if the position control is set to *displacement*, its meaning is the distance to be reached from the starting coordinate of the test;

Timeout: maximum test duration.

BREAK DETECTION

Break detection activation: if this control is set to OFF, no automatic break detection executes; if it is set to ON, the following parameters are visible and active;

Percentage break threshold: this is a percentage, calculated with respect to the current load, that defines a limit difference between two consecutive load values to detect a fracture through a dedicated automatic algorithm; this parameter acts independently from the configuration of the following two parameters, and it is the first break detection criterion;

Load variation for break detection (abs. value) and Reference time interval: these two parameters operate together to set the second possible automatic break detection criterion: a fracture is detected if a load variation greater or equal to the specified threshold is detected within a time interval equal to the value set as *Reference time interval*, at any moment of the test;

Fracture monitoring starting threshold (abs. value): if the checkbox to activate this function is selected, this number represents the absolute value of the load below which the fracture detection algorithm is neutralised even if it is active: this is especially useful in tests that do not initially involve full contact between the specimen and the grips, which in the first moments of the test could generate false fracture detections due to the noise of the load signal; for example, if the set value is 100 N, then the fracture detection algorithm does not intervene under any circumstances for loads between -100 and 100 N.

SPECIMEN GEOMETRY PROPERTIES

These values do not influence the operation of the machine and are only used as archive information and, in case of static tests, to calculate the material properties such as the elasticity modulus.

It is possible to select various geometric shapes for the section of the test specimen: depending on which one is selected, the programme displays the significant parameters of that shape (e.g. width and thickness for a rectangular section, outer diameter for a circular section) and automatically calculates the area of the section; if the “other shape” option is selected, the user has to enter the area value manually.

For tension and compression tests, the software estimates the elasticity modulus, the resistance coefficient and the hardening exponent of the specimen if the checkboxes dedicated to the geometry properties are selected and the value inserted are correct.

SAFETY CONFIGURATION

These are the load and position limit that the machine has to respect for safety purposes; they differ from the test end conditions since the latter are references to consider the test finished, whereas the safety limits – as their name points out – are parameters aimed at preventing damages.

Safety limits setting: through this option the user can choose between using the standard load and position limits of the machine, that are the maximum acceptable values given by the manufacturer and not possible to change, and specific limits that can be customized by the user within the range set by the standard limits.

ADVANCED SETTINGS

Position control mode: gives the possibility to choose between position and displacement regarding the target coordinate of the test configured in the test end conditions; when this parameter is changed a warning popup is displayed to point out that the meaning of another control changed;

Minimum and maximum strain for E calculation: these are the strain limits to use in order to calculate the elasticity modulus of the material; the checkboxes, that are activated and deactivated together, state whether the software has to use the written limits for the calculation or detects the interval through an automatic algorithm; in order to disable the calculation itself, the checkboxes related to the *Specimen geometry properties* have to be deselected;

Load and position sign inversion: this option is visible only for compression tests; if activated, it changes the sign of the load and displacement values that are read from the field (which are usually negative during a compression test) so that the shape of the profile becomes similar to that of the traction test.

DATA SAVING

Sampling frequency: the software automatically saves the results file, with this menu users can choose the acquisition frequency (valid for all channels) for saving the data to file.

EXTENSOMETER MANAGEMENT

These parameters are useful if the tensile or compression test is conducted using a strain gauge, otherwise there is no need to configure them.

Reset extensometer at test start: if this option is activated, the extensometer is automatically reset when starting the test, before starting the actuator movement;

Waiting time to stabilize strain signal: if an automatic reset of the strain gauge is set, it is possible to configure a waiting time after this operation and before starting to move the actuator, in order to stabilise the strain signal against the effect of any filtering: a time within 0.5 seconds is generally sufficient even for the most extreme filtering settings;

Warning popup to remove extensometer: it is possible to have the software display a pop-up message reminding the user to remove the extensometer before it reaches an excessive strain; this option

can only be significant if the machine is not equipped with interlocking doors, otherwise opening the guards to remove the extensometer would cause the test to stop immediately; the message can be activated when a certain level of displacement or strain is reached, or the first of these two conditions to occur;

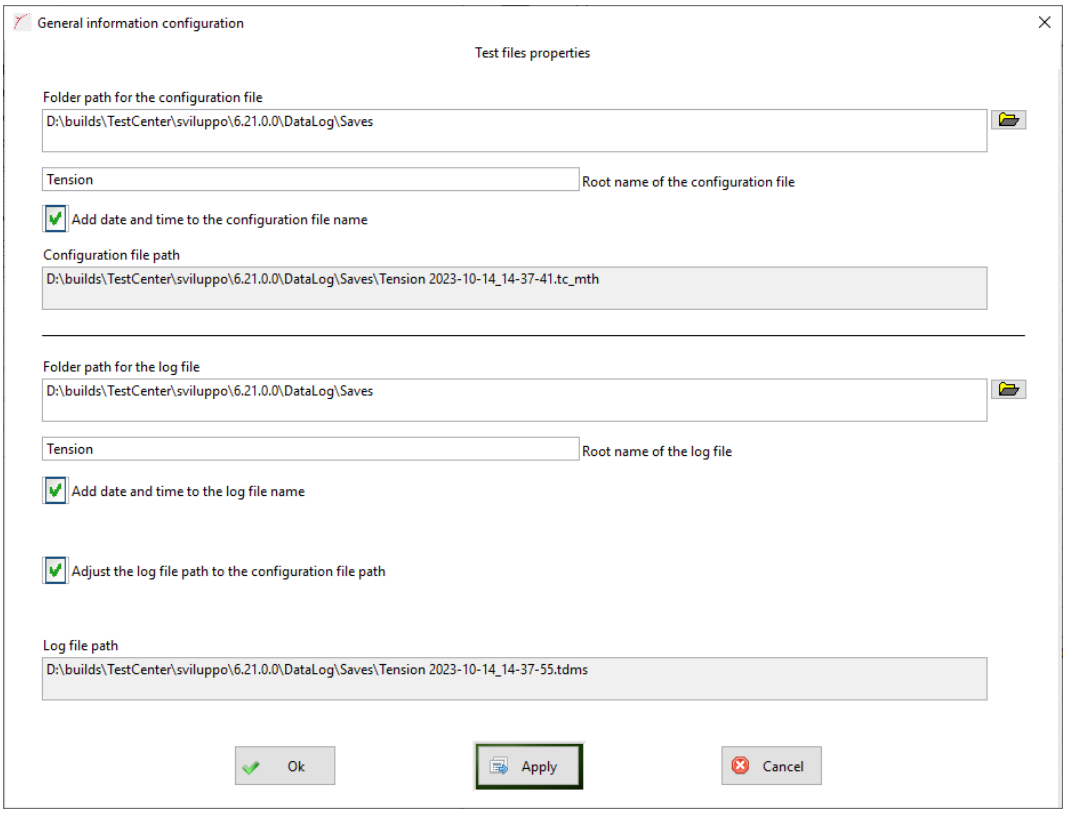
Displacement for extensometer removal: if the warning message to remove the extensometer is active, and if the tick for this control is selected, this is the displacement level (from the actuator position at test start) at which the message should be displayed;

Strain for extensometer removal: if the warning message to remove the extensometer is active, and if the tick for this control is checked, this is the level of strain (measured by the extensometer itself) at which the message should be displayed;

Extensometer base length: this is a datum that is recorded in the results file and allows the strain data to be visualised in post-processing both in dimensionless values and expressed as displacements.

On the right side of the test interface, a graph is visible, which is automatically populated at the end of each test and allows the load-position curve for all specimens collected in the same test file (i.e. those visible in the list on the left) to be displayed.

At the right-hand end of the screen is the button to open the interface from which to set the configuration and save file paths:



The upper section is for the test configuration file (.tc_mth), while the lower section is for the test results file (.tdms); it should be noted that in tensile and compression tests, only one tdms file is generated, comprising all the specimens created within the same configuration file.

For each of the two files, it is possible to select the folder path and file name, and also to choose whether to include the creation timestamp in the file name to ensure that it is unique. In addition, for the results file, it is also possible to choose whether to include the specimen name in the file name, and there is also an option to automatically adapt the path and name of the results file to those chosen for the configuration file, so that it is not necessary to make the same change twice if the user wants to keep the two files in the same place and with the same name.

These settings must be configured before starting the first test, otherwise they are no longer editable.

Finally, the file path visible on the bottom of the interface belongs to the test configuration document that is being configured and automatically saved.

4.2. Creep test

The parameters to be set are:

GENERAL SETTINGS

Test control mode: load or position, it is the parameter that has to be controlled during the test;

Position control mode: the user can choose between position and displacement for the setting of the target position, if the position control is selected, and for the visualization and saving of that coordinate during the test;

Target speed: limit speed of the axis in the movement towards the target load or position; the actual speed value decreases with the progression of the test because of the control algorithms that allow, as the creep test protocol requires, to reach the target load or position precisely; if the user tries to start a test with a null speed value the program stops the procedure and opens a warning popup window;

Force rate: this parameter represents the desired variation rate for the load during the movement process that leads to the accomplishment of the target load, and must be specified as an absolute value (i.e. without sign); if the checkbox placed between the description and the value is not selected, the software takes into account only the target speed for the motion;

Target load: defined load for the creep operation (this setting is visible only if the test is load-controlled);

Target position: defined position for the test (this setting is visible only if the test is position-controlled);

Total time: creep duration, starting from the moment when the target load or position is reached (not from the instant when the test is launched);

Material type: this is an enumerative parameter that permits a basic self-regulation of the machine with respect to the kind of material that is being tested (the user has to select the option which is the most similar to the actual material).

TEST END CONDITIONS

Break threshold (force variation): if the machine detects a load variation higher than this value between two consecutive scans, the test is automatically interrupted because a fracture is identified;

Return to neutral position at test end: if this option is activated, after the target time for the creep has elapsed, the machine unloads the specimen until null load is reached, otherwise it holds the final position.

SPECIMEN GEOMETRY PROPERTIES

These values do not influence the operation of the machine and are only used as archive information and, in case of static tests, to calculate the material properties such as the elasticity modulus.

It is possible to select various geometric shapes for the section of the test specimen: depending on which one is selected, the programme displays the significant parameters of that shape (e.g. width and thickness for a rectangular section, outer diameter for a circular section) and automatically calculates the area of the section; if the "other shape" option is selected, the user has to enter the area value manually.

SAFETY CONFIGURATION

These are the load and position limit that the machine has to respect for safety purposes; they differ from the test end conditions since the latter are references to consider the test finished, whereas the safety limits – as their name points out – are parameters aimed at preventing damages.

Safety limits setting: through this option the user can choose between using the standard load and position limits of the machine, that are the maximum acceptable values given by the manufacturer and not possible to change, and specific limits that can be customized by the user within the range set by the standard limits.

ADVANCED SETTINGS

Dead band: tolerance percentage on target load or position, within which the motion phase of the creep is concluded and the holding time count starts;

PID proportional gain: in keeping with the definition of this parameter in a PID controller, it is a constant value used to multiply the error signal in order to correct the setpoint value of load or position; see paragraph 4.2.1 for a more detailed explanation and some usage hints;

Automatic material characterization: activates or deactivates an automatic procedure, performed at the beginning of the test, through which the machine self-regulates with respect to the rigidity of the specimen. The procedure is composed of a movement at controlled speed (which value is automatically obtained combining some test parameters such as material type and geometry properties) until the load reaches 5% of the difference between target value and initial value. This parameter is visible only if the test is load-controlled and, if deactivated, requires the user to precisely set the *Material rigidity* field;

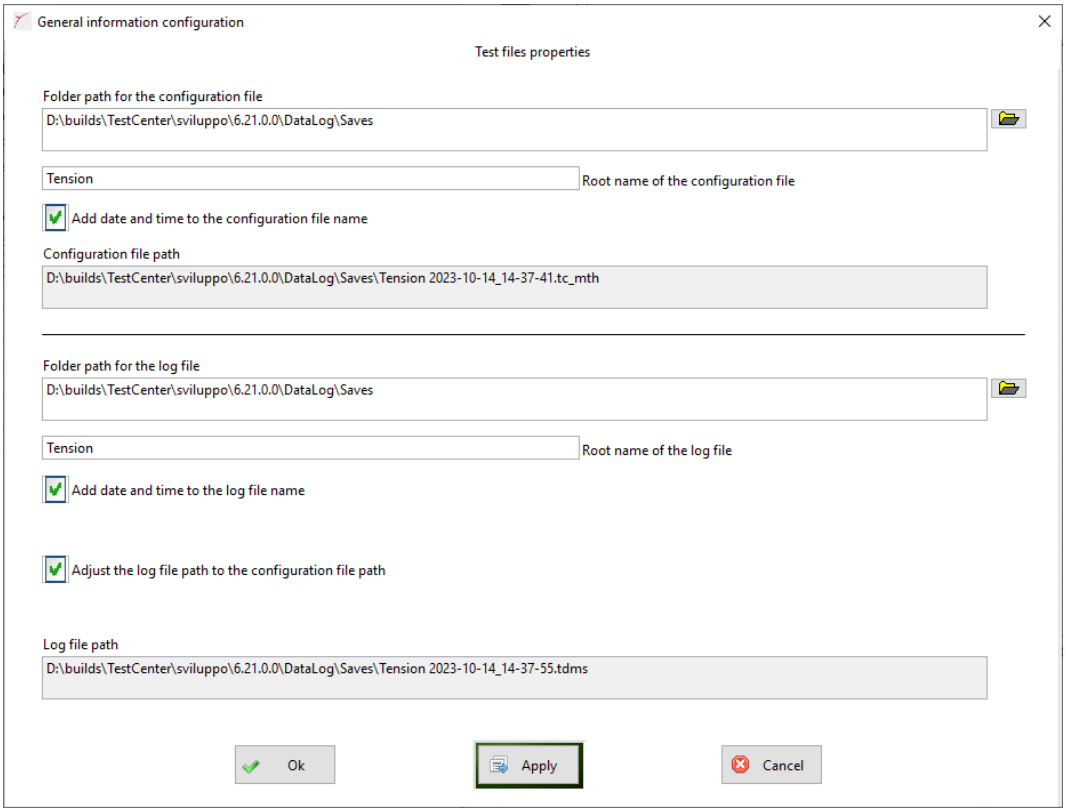
Material rigidity: rigidity value, expressed in N/m or other commensurable unit, that applies for the specimen; this field is visible only if the automatic material characterization is visible and deactivated, but in such a case it is mandatory to insert a valid value; if 0 is left as input, the program automatically re-activates the automatic characterization procedure even if the user deselected it. The rigidity has to be inserted as an absolute value, or in other words without sign.

DATA SAVING

Sampling frequency: it is the frequency of data acquisition, and storage in the TDMS file, for the entire duration of the creep test.

On the right side of the test interface, a graph is visible, which is automatically populated at the end of each test and allows the load-position curve for all specimens collected in the same test file (i.e. those visible in the list on the left) to be displayed.

At the right-hand end of the screen is the button to open the interface from which to set the configuration and save file paths:



The upper section is for the test configuration file (.tc_mth), while the lower section is for the test results file (.tdms); it should be noted that in tensile and compression tests, only one tdms file is generated, comprising all the specimens created within the same configuration file.

For each of the two files, it is possible to select the folder path and file name, and also to choose whether to include the creation timestamp in the file name to ensure that it is unique. In addition, for the results file, it is also possible to choose whether to include the specimen name in the file name, and there is also an option to automatically adapt the path and name of the results file to those chosen for the configuration file, so that it is not necessary to make the same change twice if the user wants to keep the two files in the same place and with the same name.

These settings must be configured before starting the first test, otherwise they are no longer editable.

Finally, the file path visible on the bottom of the interface belongs to the test configuration document that is being configured and automatically saved.

4.2.1. Proportional gain setting

The proportional gain is a central parameter for the correct working of the test especially for load-controlled procedures and in particular when the force rate is needed.

Since inexpert users may have difficulties in getting a correct value for this parameter, it is recommended to enable the automatic characterization of the material unless the specimen mechanical properties are precisely known.

Below some useful guidelines can be found:

- The higher the test dynamics required is (i.e. higher values of force rate), the higher "optimal" proportional gain gets;
- If the load signal oscillates around the target value, then the proportional gain is likely too high;
- On the contrary, if the machine has a slow response or the feedback signal is delayed with respect to the setpoint signal, the proportional gain is too low.

Following some advised value ranges for proportional gain when configuring a creep test:

- For position-controlled tests (relaxation tests), $k_p \leq 1$;
- For load-controlled tests: $k_p \approx 2{,}5$ for low force rate values (e.g. 10 N/s), raising up to approximately 25 in parallel with the raising of the target force rate.

4.3. Fatigue test

The parameters to be set are:

MAIN SETTINGS

Load session index: since several load sessions can be performed within a single fatigue test, each of them with its own configuration regarding all the parameters related to the *General settings*, and all the load sessions are executed automatically and sequentially by the machine, this index permits to load and set the parameters of the desired load session into the interface menu;

Cycle number: target number of cycles to be performed in the current load session;

Test control mode: load or position, it is the parameter that has to be controlled during the load session;

Maximum and minimum load: upper and lower load values for the current session (these settings are visible only if the force control is selected);

Position control mode: choosing between position and displacement, is has to do with the interpretation of the target position values (this setting is visible only if the position control is selected);

Maximum and minimum position: upper and lower values for the current load session (these settings are visible only if the position control is selected): if the position control is set to *position* then these values are considered in the user-selected coordinate system, whereas if the *displacement* option is activated the two numbers become displacements calculated with respect to the actuator position when the test is started;

Frequency: test frequency for the load session, expressed in Hz (1 Hz = 1 cycle per second, regardless of the stroke required);

Temperature: this field is visible only if the machine is coupled with a climatic chamber; when the dedicated checkbox is activated, the load session to which the value is referred is anticipated by a temperature regulation process that leads the chamber to the desired value;

Time to wait before starting the session: time interval before starting the load session; if the temperature control is selected, this time count starts when the target temperature is reached.

At the end of the general settings section of the menu there are two read-only fields dedicated to the recap of the total number of cycles and load session planned for the whole test.

TEST END CONDITIONS

Acceptable % rigidity variation: this is the key parameter for break detection: if it is enabled through the dedicated checkbox, the machine stops if a rigidity variation greater than the threshold configured here is detected within a predefined window of cycles; this parameter can be modified also when a test is running; next to the numeric value there is a button that displays the graph with the current rigidity profile

of the specimen, updated in real time during the test; through this it is immediate to monitor the evolution of the rigidity and adjust the threshold value if needed;

Chamber shutdown at test end: this field is effective only if the machine is coupled with a climatic chamber, and is used to automatically turn off the latter when the fatigue test finishes;

Unloading procedure at test end: if this setting is activated, the machine automatically unloads the specimen at the end of the test, unless the stop is due to a command given by the user (instead, completion due to cycle count, break detection and alarm outbreak are followed by the unloading procedure); the unloading operation is similar to the "hold F=0" functionality described in paragraph 2;

Target load for unloading procedure: this field is visible only if the unloading procedure at test end is enabled, and sets the target load for that operation, in keeping with the following two position limits (if enabled);

Maximum position for unloading procedure: this field is visible only if the unloading procedure at test end is enabled, and represents the maximum actuator position that must be respected by the machine during that operation;

Minimum position for unloading procedure: this field is visible only if the unloading procedure at test end is enabled, and represents the minimum actuator position that must be respected by the machine during that operation.

SPECIMEN GEOMETRY PROPERTIES

These values do not influence the operation of the machine and are only used as archive information and, in case of static tests, to calculate the material properties such as the elasticity modulus.

The program automatically calculates the area of the specimen, if the checkbox next to it is enabled; as soon as the width and thickness values are modified (the two values are multiplied, calculating the area for a rectangular-section specimen, but it is also possible to manually edit the area numeric value).

THERMAL TEST PROPERTIES

The test environmental temperature is just an archive piece of information and does not modify the machine behaviour in any way.

SAFETY CONFIGURATION

These are the load and position limit that the machine has to respect for safety purposes; they differ from the test end conditions since the latter are references to consider the test finished, whereas the safety limits – as their name points out – are parameters aimed at preventing damages.

Safety limits setting: through this option the user can choose between using the standard load and position limits of the machine, that are the maximum acceptable values given by the manufacturer and not

possible to change, and specific limits that can be customized by the user within the range set by the standard limits.

DATA SAVING

Automatic saving percentage regulation: if activated, this setting automatically adapts the *cycle percentage to save* and the *max and mins saving percentage* with respect to the target cycle count of the whole test (i.e. sum of the cycle counts for each load session configured), suggesting a value that generates a good combination between data size and quantity of information gathered during the test;

Cycle percentage to save: sets how many cycles have to be saved with respect to the total planned for the test; in fact, it is not possible to save every single cycle of the test because the log file would grow too big to be handled (and such an operation would also be completely useless for long lasting tests); if the *automatic saving percentage regulation* is activated, this parameter is automatically adjusted every time a setting is modified on the interface, according to the target cycle count of the test;

Saving mode: if the linear mode is chosen, the saving interval remains constant for the entire test (e.g. a percentage set to 5% means that one cycles each 20 is saved – cycles 1, 21, 41…); if the logarithmic saving is selected, the percentage of cycles to be saved is distributed so that the interval between two consecutive cycles saved is shorter at the beginning of the test, and gradually increases as the test evolves; vice versa applies for exponential mode: savings are less frequent at the beginning of the test and more dense towards the end; finally, the on-off mode gets the specified saving percentage by logging some consecutive cycles for each interval, in keeping with the *on-off saving benchmark* parameter (see description), instead of a single one as the linear mode does;

Number of cycles to keep in memory: the value written in this control becomes the number of complete cycles that the software constantly keeps in memory during the test using a circular buffer, which means data are constantly overwritten in a dedicated memory space in order to have always the very last few cycles ready to be automatically logged on disk (in the same file that includes the cycles and peak-valley saves) in case the machine stops for any reason;

Nr. of latest cycles displayed in XY graph: sets the memory length, in terms of cycles, for the hysteresis graph of the test, given that is has a limited memory anyway, so this parameter has effect only if the number of points it sets is less than the maximum graph memory; once the graph is filled up to the number of cycles plot matches the maximum memory or the value set through this parameter, the older points are automatically substituted with the new chunks of data continuously coming from the field; this parameter can also be modified while a test is running;

Logarithmic/exponential saving parameter: referring to the saving mode previously described, this value sets the ratio between the maximum and the minimum saving interval of the test (respectively the last and the first in case of logarithmic saving, and the first and the last for exponential); in other words, this parameter establishes how much evident the logarithmic or exponential characterization of the saving

process must be (it has no effect for linear and on-off savings); for example, if a value of 20 is selected for an exponential saving mode with 5% of the cycles to be saved, the intervals distribution will be calculated in order to get 5% of the cycles logged with the first interval being 20 times bigger than the last; on the right side of this field a button is provided to open a preview window where two graphs are displayed, showing the evolution of the number of savings and of the saving interval all through the test duration, comparing all the different saving methods;

Max and mins saving percentage: the program detects a maximum and a minimum value for load and position (and also other parameters) automatically for each cycle of the test, therefore through this feature it is possible to choose how many of these peaks and valleys will be saved with respect to the total cycle number of the test (including all the load sessions); peaks and valleys are logged in a group for each load session, separated from the cycles saved; if the *automatic saving percentage regulation* is activated, this parameter is automatically adjusted every time a setting is modified on the interface, according to the target cycle count of the test;

On-off saving benchmark: this parameter operates only if the on-off saving mode is selected and, together with the percentage of cycles to save, defines how savings have to be organized; for example:

- If 5% of the cycles is selected for logging and the benchmark is set to 100, then the program saves 5 cycles each 100 benchmarked against intervals of 100 cycles, therefore the cycles to be saved are 1..5, 101..105, 201..205 and so on;
- If 5% of the total cycles is selected for logging and the benchmark is set to 200, then the program saves 5 cycles each 100 benchmarked against intervals of 200 cycles (i.e. 10 out of 200), therefore the cycles to be saved are 1..10, 201..210, 401..410 and so on.

ADVANCED SETTINGS

Load range % to use for calibration: this is a percentage value needed only if the load control mode is selected, and regulates the preliminary calibration procedure of the test: in fact, before starting the fatigue cycles the actuator operates three movements: the first one towards the maximum load value, the second towards the minimum load value and the third to the average load value; this parameters sets the portion of the range that has to be covered; the range covered is equally distributed starting from the average load value;

Peak-valley regulation gain: this is the proportional gain for the peak-to-peak algorithm that regulates the load-controlled fatigue tests: the higher is this parameter, the bigger are the corrections commanded by the controller to the target setpoint of the next cycle basing on the difference between the target peak load and the value measured in the previous cycle;

Peak-valley regulation saturation: if the dedicated checkbox is activated, this is the maximum position correction that the controller can command in one cycle; a value of 0 means that the controller uses its own default settings for this;

Cycle count during the initial transient: if this option is activated, the controller starts to increment the cycle counter at the very first waveform executed; otherwise, the count starts only once the commanded maximum and minimum values are met (if the test is controlled in load, it takes some preliminary cycles to reach the target values after the calibration).

Examples for the usage of the Load range % to use for calibration *parameter:*

- Maximum load = 500 N, minimum load = 100 N, % to use for calibration is 75: the average load is 300 N, the range is 400 N, therefore the calibration procedure covers a range of 0,75 * 400 = 300 N, so that the maximum load during the calibration is 300 + 150 = 450 N and the minimum is 300 – 150 = 150 N;
- Maximum load = 1000 N, minimum load = 100 N, % to use for the calibration is 40: the average load is 550 N, the range is 900 N, therefore the calibration procedure covers a range of 0,4 * 900 = 360 N, so that the maximum load during the calibration is 550 + 180 = 730 N and the minimum is 550 – 180 = 370 N

In order to use this parameter at best, the following annotations need to be taken into consideration:

- The lower the percentage value is set, the longer becomes the transition phase in which the fatigue cycles are still not in tolerance and therefore are not counted;
- For low-cycle fatigue, especially in case of ductile materials, it is recommended to use medium-low percentage values (e.g. around 50%) in order to prevent the calibration from damaging the specimen: in fact, the calibration procedure is carried out through a creep action so that, is the maximum value is targeted, there might be quite a long time (10-15 seconds or more, depending on the characteristics of the material) when the specimen is stressed by a load that is very close to its yield limit.

RESULTS EXPORT

CSV file export at test end: if activated, this setting commands the software to export a csv-format file with some significant test information and results automatically when the test is finished; all the following options are visible only if this is set to ON;

Folder path where to save the file: the folder where the csv results file has to be saved;

File name (without extension): this is the name to be given to the output file, without comprehending the extension, which is given by default;

Add date and time to file name automatically: if this setting is activated, the software adds the timestamp to the file name when the output file is generated; this option is useful to prevent other files with the same name and at the same path from being overwritten;

Column separator character (write TAB or \\t for tab character): this is the character to be used as column separator in the csv output file;

File generation condition: it is possible to configure whether to save the file always, always except for the case of a test stop command given by the user manually, or only in case of a fracture detection.

On the right side of the interface there is another tab menu: the first tab displays a preview of the profile of a single cycle that is being configured for the current load session.

The second tab includes a recap table for all the load sessions planned for the test, showing all the characteristic parameters (number of cycles, target load or position, frequency, temperature if included, etc.); the table is automatically updated as soon as the user changes one of the parameters involved; also, the table works as a load session selector, since if the user clicks on a row then the software automatically selects the load session corresponding to that row; the selected session row is in bold, whereas during a test the current session row is underlined

The third tab is updated while a test is running and displays some data about how the machine is performing: the current frequency in Hz and in cycles per hour, the current peak speed, and possibly events occurred during the test, such as break detections.

In the final tab of this section of the interface there graphs are visualized: these are previews of the hysteresis cycles (load-position) saved from the beginning of the current load session; at least 3 cycles need to be saved and the test must be running for the preview to work, and the user can use the scrollbar on the right to navigate through all the saved cycles. Once the test is finished, it is not possible anymore to consult this preview section, and the user needs to run the post-processing environment.

The test environment also includes three graphs: the first one displays the sampled channels with respect of time during the test, the second provides the load-position hysteresis loop, the third shows the maximum and minimum values for load and position sampled for each cycle.

For each graph there is an arrow button on the bottom-right corner of the interface, and by clicking on it a window containing the following controls appears:

 : saves a picture of the graph;

 : exports the graph data as a CSV file, which can be loaded in a common notepad or imported in a spreadsheet;


 : invert the axes scales;

 : autoscales the graph.

In order to make the window disappear it is enough to exit it with the cursor or press the arrow button again.

Finally, the file path displayed on the bottom applies to the configuration file in use: it is possible to move and rename this document and the results file connected to it, the details are explained in paragraph 5.

4.3.1. Changing settings during a test

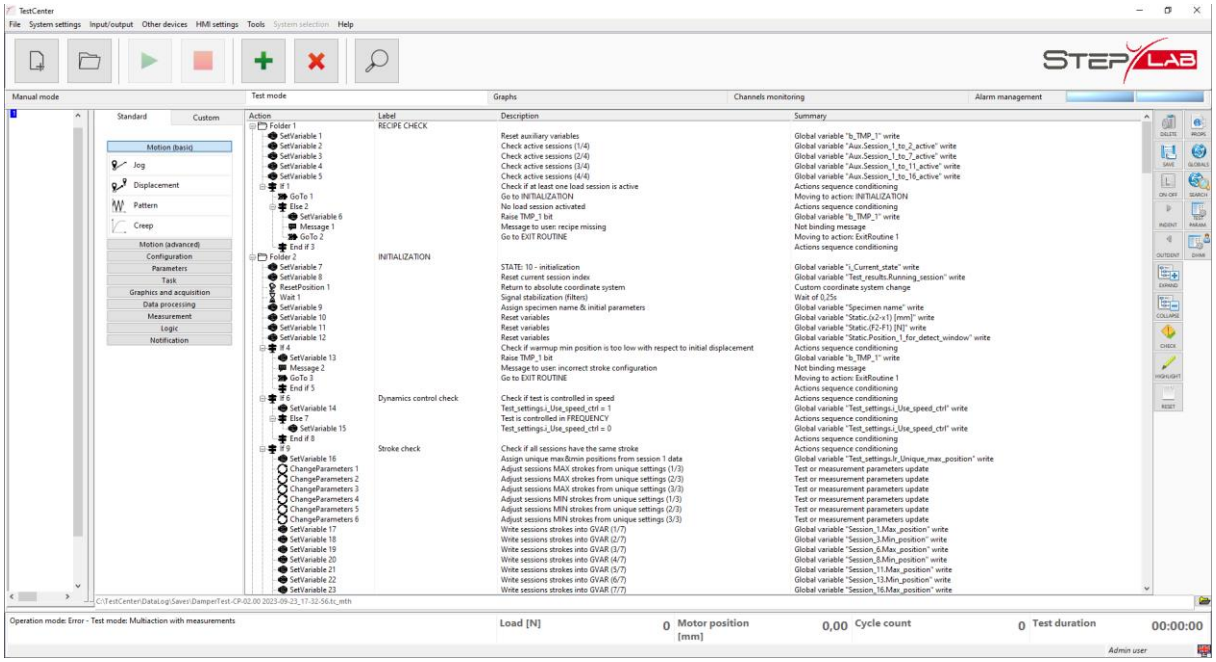
While a test is running, most of the configuration menu is greyed out and disabled, except for a few parameters, among which the most significant are the test frequency, the break threshold and the safety limits (if custom ones are in use): these values can be modified even if the machine is operating, in that case being automatically transferred to the controller that adapts to the new setting, without any need to stop the test.

Instead, the parameters that are greyed out when the test is started and remain in that condition even if it is interrupted are not editable in any case for the current specimen: a new specimen must be created to have them changed.

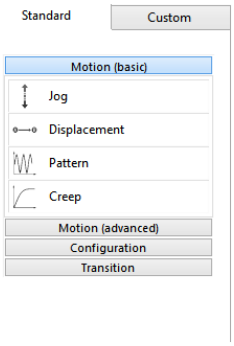
Caution: *if the frequency of a load session needs to be modified while the machine is running the user must care about the load session index value, since an eventual change to the frequency displayed applies only to the session to which that value is referred. Therefore, if the user changes the frequency for a session that does not match the one in progress, the actuator does not change its speed.*

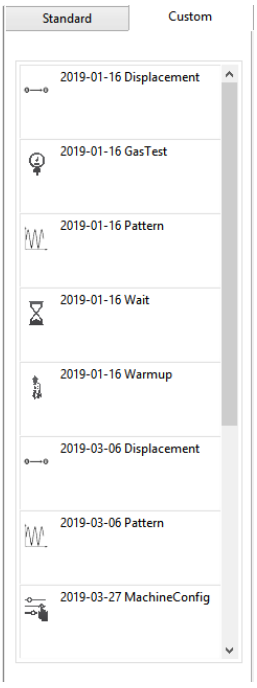
4.4. Multiaction test

The *multiaction test* mode is an advanced testing protocol that allows users to build their own test procedure through the composition of simple actions, also called *microactions*, as if they are bricks of a wall. The following image represents the interface of a multiaction test:



On the left side of the window there is the same listbox of specimens that is available also on the other test environments to gather several tests into the same configuration file. Next, moving rightwards, there is a tab control divided into *Standard* and *Custom*:



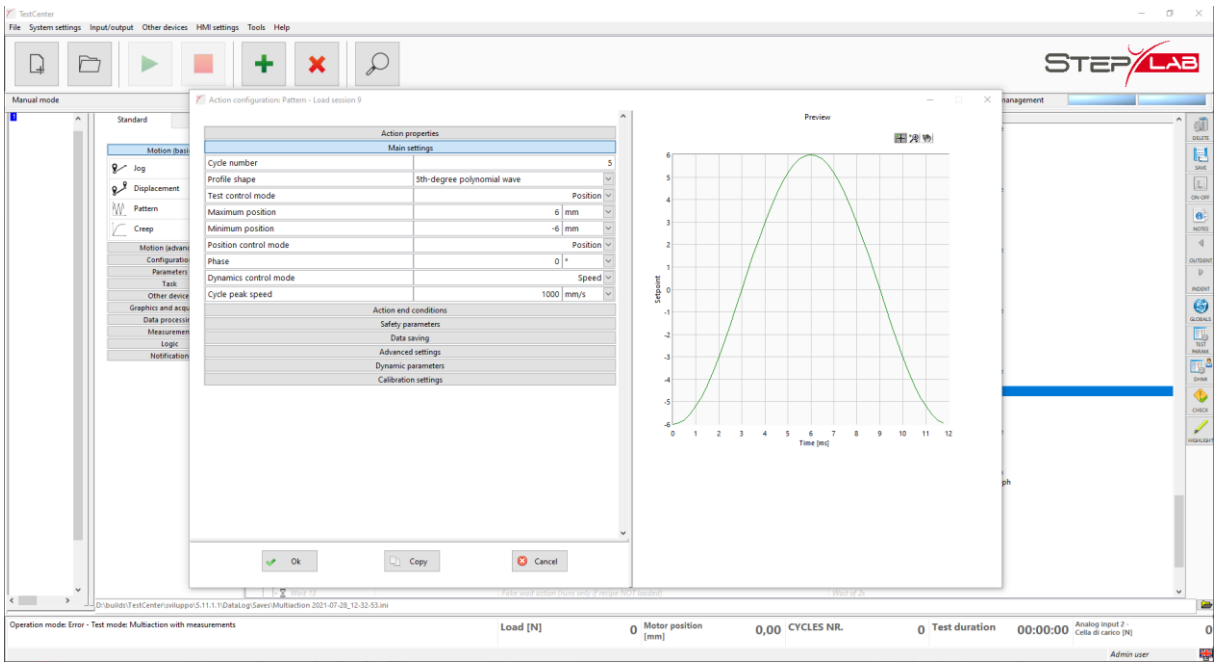


The standard section includes all the default actions available to be placed in the sequence by dragging and dropping them into the tree menu that is contained in the rest of the interface; the custom section also includes actions that can be copied into the test procedure, but these actions are saved by the user as templates in the *xml\Actions* subdirectory of the application path, so that it is possible to build a preconfigured action with the desired parameters and then store it to recall is as many times as it is needed; the way to place the template actions as part of the sequence is also dragging and dropping.

Aside the microactions tab there is the tree menu that represents the test sequence, where four columns can be distinguished:

- The first column displays the action name and an icon that represents it;
- On the second column there is the label of the action, which can be written by the user: the label determines part of the name of the data group that will include the savings of that action in the results file;
- The third column is dedicated to a description for the action, which can be also chosen by the user;
- On the fourth column, a summary string for the most relevant parameters of the action can be found.

By double-clicking on an action of the sequence an interface for the configuration of the chosen action is opened:

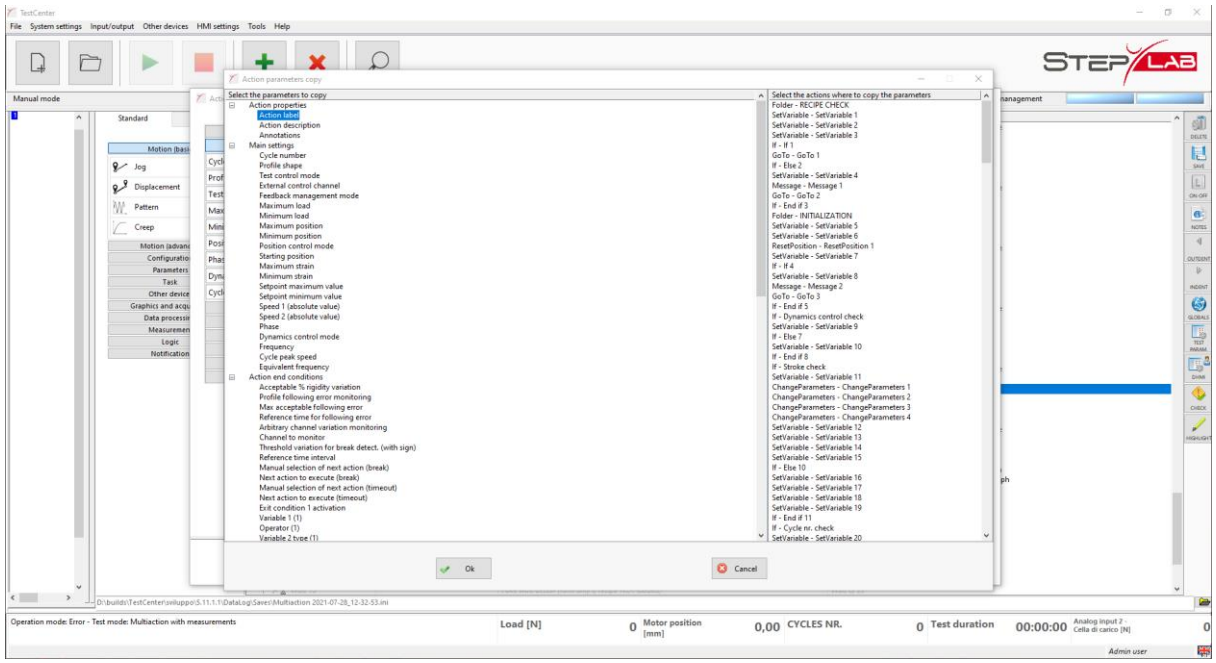


The configuration is divided into sections with the same logic of the standard test interfaces configuration menus; some sections are common, some other are specific of the various actions and will be described in detail in the following paragraphs.

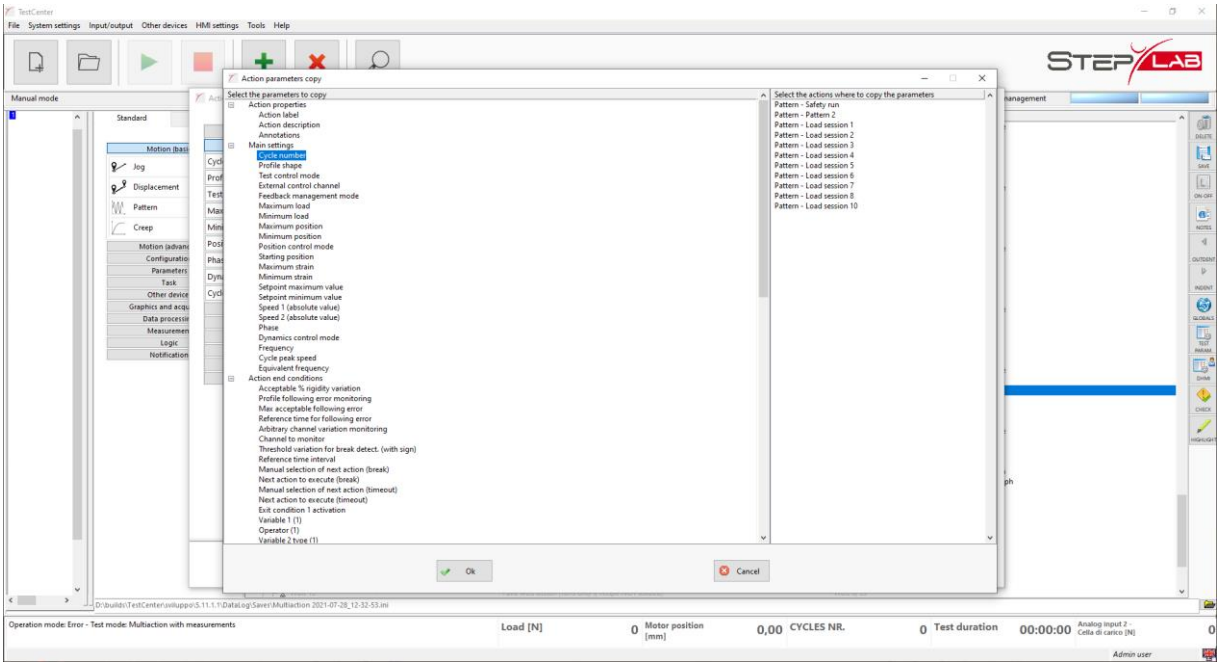
Finally, on the bottom of each configuration interface three buttons can be found:

- OK: saves all the changes and closes the configuration interface;
- CANCEL: if there are changes done to the configuration, the user is asked to confirm that they won't be saved; if so, the window gets closed and all the action parameters are restored as they were at the moment of the window opening;
- COPY: opens an interface dedicated to copy the parameters from the selected action to others; on the left of the window there is a tree with the list of the parameters available for copy, on the right a list of the actions available for the copy is located; multiple parameters can be selected at the same time through the CTRL or SHIFT keys, and if a parent is selected on the tree then all the children get selected at the same time.

  If only common parameters are selected, all the actions of the sequence are available for the copy:



  If there are parameters that are specific of the current action among the selections, only the action of the sequence that are of the same type are available:



The list on the right accepts also multiple selection through the CTRL and SHIFT buttons; then, if the user presses the OK button on the bottom of the page, the selected parameters are copied on the chosen microactions.

4.4.1. Common configuration parameters

The following parameters are common for the various microactions and therefore are described only here instead of repeating them for all the actions:

ACTION PROPERTIES

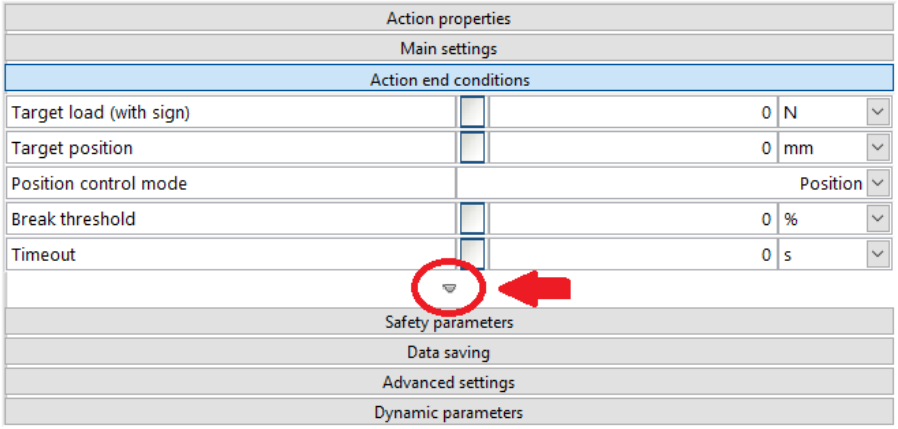
Action label: a label for the action that determines part of the name of the data group that will include the savings of that action in the results file, and is also reported on the second column of the test sequence tree of the multiaction interface;

Action description: the user can type here a text to describe the action, that will be visible on the third column of the multiaction interface sequence tree;

Annotations: some annotations regarding the action, that can be also modified during the test.

ACTION END CONDITIONS

Some of the actions are equipped with advanced end conditions, that are customizable: it is possible to access the advanced action end conditions if the action interface displays the arrow highlighted in the following picture:



Manual selection of next action (break) and Manual selection of next action (timeout): each of these two controls is visible only if the respective end condition (automatic break detection and timeout) is activated through its checkbox; if the selection is ON, another parameter – named *Next action to execute* – is displayed, with a drop-down menu that allows to choose which action to execute next in case of fracture or timeout; if the selection is OFF, the sequence automatically moves to the following action in case the end condition is met;

First custom exit condition, second custom exit condition: these are the two customizable action end condition that can be configured in the same way:

Exit condition 1 (or 2) activation: if this is ON, the custom exit condition is activated and the following parameters are displayed; otherwise, it does not execute;

Variable 1: the drop-down menu allows to choose among all the variables available to be placed as first member of the end condition;

Operator: this is the mathematical operator that has to be used for the comparison that determines the end condition;

Variable 2 type: the second member of the comparison can be a constant value or another variable;

Constant value for variable 2: this element is visible only if *Variable 2 type* is set to *Constant*, and it is the constant value to be used as second member of the comparison that states the end condition;

Variable 2: this element is visible only if *Variable 2 type* is set to *Variable*, and it is the variable (to be chosen through the drop-down menu as *Variable 1*) to be used as second member of the comparison that states the end condition;

Manual selection of next action and Next action to execute: these two controls operate in the same way as the ones for break detection and timeout: if *Manual selection of next action* is ON and the respective custom end condition is met, then the sequence moves to the action specified in *Next action to execute*.

Examples:

1. Variable 1 is *Load [N]*, operator is ≥, Variable 2 type is *Constant*, Constant value for variable 2 is 500: the action custom end condition is met as soon as the measured load value becomes greater or equal to 500 N
2. Variable 1 is *Analog input 1*, operator is ≤, Variable 2 type is *Variable*, Variable 2 is *Analog input 3*: the action custom end condition is met as soon as the measured value of the first analog input 1 is less or equal than the measured value of the third analog input.

SAFETY CONFIGURATION

These are the load and position limit that the machine has to respect for safety purposes; they differ from the test end conditions since the latter are references to consider the test finished, whereas the safety limits – as their name points out – are parameters aimed at preventing damages.

Safety limits setting: through this option the user can choose between using the standard load and position limits of the machine, that are the maximum acceptable values given by the manufacturer and not possible to change, and specific limits that can be customized by the user within the range set by the standard limits.

DATA SAVING *(this section may differ from the following standard layout for some action, in which case it is described in detail in the dedicated section)*

Data saving: if OFF, data logging on the results file is deactivated for the action; if ON, it is activated;

Sampling frequency: this ring menu is effective only if saving is activated for the action, and sets the sampling frequency of the data for logging during that action; the frequency can be chosen only among the given values;

Saving group name: if the user writes some text here, this becomes part of the data group dedicated to the current action on the results file, together with the action label;

Include action label in group name: if this option is active, the label of the action becomes part of the group name, otherwise it is ignored;

Include execution nr in group name: if this option is active, the number of executions of the action becomes part of the group name, otherwise it is ignored: the number of executions may be useful to separate the data acquired during the various iterations of a single action, if the test sequence is more complex than a classic top-down one and includes loops and repetitions of some sections.

DYNAMIC PARAMETERS

Acceleration: the motor acceleration for the movements of the current action;

Jerk: the motor jerk for the movements of the current action.

Please note that, depending on the system layout and technical specifications, the number of actions available and their parametrisation may vary.

4.4.2. Jog action

A *jog* action is an movement at constant speed in the chosen direction, that is interrupted only when one of the established trigger conditions occurs.

The specific parameters that can be configured for a *Jog* action are:

MAIN SETTINGS

Target speed: this is the value of the constant speed that the actuator will keep all through the motion of this action;

Motion direction: depending on the configuration chosen for the machine, the positive direction can be associated either to tension or to compression.

ACTION END CONDITIONS

Target load: force to reach in the test (does not take into account the initial load, and has to be written with its sign);

Target position or target displacement (with sign): if the “Position control mode” parameter is set to *position* the description displayed is *Target position* and the parameter represents the limit coordinate to stop the test; instead, if the position control is set to *displacement*, the description becomes *Target displacement* and its meaning is the distance to be reached from the starting coordinate of the test;

Position control mode: gives the possibility to choose between position and displacement regarding the target coordinate of the test configured in the action end conditions;

Break threshold: this is a percentage, calculated with respect to the current load, that defines a limit difference between two consecutive load values to detect a fracture through a dedicated automatic algorithm;

Timeout: maximum action duration.

4.4.3. Displacement action

A *Displacement* action is a movement targeted to a specific coordinate that is reached with complete precision unless a trigger condition (if any is set) or a safety limit occurs before.

The specific parameters that can be configured for a *Displacement* action are:

MAIN SETTINGS

Target position: this is the coordinate where the displacement action has to end unless a trigger or safety condition intervenes before;

Target speed: this is the value of the actuator speed for the displacement; this value has to be always positive, the motion direction does not matter.

ACTION END CONDITIONS

Target load (with sign): if the checkbox dedicated to this control is activated, then a trigger condition is set for the load value during the displacement action, defined by the numeric value and the operation: as soon as the trigger occurs, the action stops;

Break threshold: this is a percentage, calculated with respect to the current load, that defines a limit difference between two consecutive load values to detect a fracture through a dedicated automatic algorithm;

Timeout: maximum action duration.

4.4.4. Pattern action

A *Pattern* action is a sequence of fatigue cycles that can be performed with several types of waves (please note that it is not possible to execute different types of waves within the same action).

The specific parameters that can be configured for a *Pattern* action are:

MAIN SETTINGS

Cycle number: target number of cycles to be performed in the current load session;

Profile shape: it is possible to select the wave type that will be applied to the entire action;

Test control mode: it is the parameter that has to be controlled during the load session; starting from software version 6.7, there is also the possibility to select a "hybrid" control mode, for which two different feedbacks can be used for the two setpoints value (at the end of this paragraph there is the description of the parameters dedicated to the hybrid control mode);

External control channel: this setting is visible only if external control is used, and sets the analog input channel that has to be taken as feedback; starting from software version 6.7, the external control mode can be used setting also mathematical channels as feedback;

Feedback management mode: this setting is visible only if load or external control is used, and allows to choose between "pure" load feedback and peak-to.peak algorithm, that is a position-feedback profile with a continuous correction of the maximum and minimum value to match the commanded loads as precisely as possible;

Maximum and minimum load: upper and lower load values for the current session (these settings are visible only if the load control is selected);

Position control mode: choosing between position and displacement, is has to do with the interpretation of the target position values (this setting is visible only if the position control is selected);

Maximum and minimum position: upper and lower values for the current load session (these settings are visible only if the position control is selected): if the position control is set to *position* then these values are considered in the user-selected coordinate system, whereas if the *displacement* option is activated the two numbers become displacements calculated with respect to the actuator position when the test is started;

Dwell time: this parameter is visible only if the "haversine (pulse + dwell)" waveform is selected, and states the time between two consecutive pulses;

Speed 1 (absolute value) and Speed 2 (absolute value): if the chosen *Test control mode* is speed, these are the two reference speed values that the cycle has to execute;

Setpoint maximum value and Setpoint minimum value: if the chosen *Test control mode* is external, these are the maximum and minimum value for the chosen analog channel during the fatigue cycles; the unit of measurement comes from the channel calibration;

Starting position: this parameter is visible only if the *combined* mode is selected in the *dynamics control* and sets the reference coordinate for the maximum and minimum position, that are disabled because the combined control mode calculates the stroke from the speed and frequency values, that are the two independent parameters to be configured;

Phase: it is possible to set a phase different than 0 for the cycle by editing this value;

Dynamics control mode: this ring option allows the user to choose among three different modes:

- Frequency: the frequency is the only variable used to set the period of the cycle;
- Speed: the period of the cycle is calculated from the peak speed;
- Combined: frequency and speed are both set by the user; in this case, the stroke becomes a dependent variable and it is only possible to offset it through the *starting position* field.

Frequency: test frequency for the load session, expressed in Hz (1 Hz = 1 cycle per second, regardless of the stroke required);

Cycle peak speed: this is the peak speed value that has to be reached during the cycle;

Equivalent frequency: this is a read-only field where a frequency value is calculated from the current speed and stroke parameters, so that the user can have a benchmark in terms of frequency even if a speed-controlled test is being configured;

On the right side of the interface a graph is dedicated to display a preview of the profile of a single cycle.

Finally, this table summarises which parameters are shown (and therefore needed to configure) depending on the selected *dynamics control mode*:

Dynamics mode	Frequency control	Speed control	Combined control
Frequency	visible	hidden	visible
Cycle peak speed	hidden	visible	visible
Equivalent frequency	hidden	visible, read-only	hidden
Starting position	hidden	hidden	visible
Max&min position	to be configured	to be configured	visible, read-only

Only for hybrid control mode:

Hybrid mode setpoint 1 and 2 control mode: this is the type of feedback to be used for the respective setpoint;

Hybrid mode setpoint 1 and 2 external ctrl channel: in case the respective “hybrid mode setpoint control mode” is “external”, this is the channel to be used as feedback;

Setpoint 1 and 2 commanded om hybrid mode: the two values of the setpoint to be applied with reference to their respective feedback channel.

ACTION END CONDITIONS

Acceptable % rigidity variation: this is the key parameter for break detection: if it is enabled through the dedicated checkbox, the machine stops if a rigidity variation greater than the threshold configured here is detected within a predefined window of cycles; this parameter can be modified also when a test is running;

Profile following error monitoring: if this option is activated, *Max acceptable following error* and *Reference time for following error* are displayed and used by the controller to set another possible break detection criterion;

Max acceptable following error and Reference time for following error: the automatic break detection condition consists in an overall following error that becomes greater than the specified *Max acceptable* value within the *Reference time interval*;

Arbitrary channel variation monitoring: if this option is activated, the following three parameters are displayed and taken into consideration by the controller to set another possible break detection criterion;

Channel to monitor: the user has to choose a channel, among those that are available, that will be subject to continuous monitoring through the *Pattern* action in order to detect if the condition stated by *Threshold variation for break detect.* and *Reference time interval* is intercepted;

Threshold variation for break detect. (with sign) and Reference time interval: the automatic break detection condition consists in a variation of the selected channel, within the *Reference time interval*, that is greater or equal to the specified *Threshold*.

Max/min tolerances activation: if this option is enabled, the following four parameters are visible and the controller monitors that the maximum and minimum values of the feedback channel do not deviate from the setpoint beyond the set tolerances;

Type of tolerances to use: the user can choose to set percentage tolerances or enter absolute values to define tolerance bands on maximum and minimum values;

Tolerance to respect for value 1 (max): this is the tolerance value (absolute or percentage) to be observed on the first setpoint value, typically the maximum;

Tolerance to respect for value 2 (min): this is the tolerance value (absolute or percentage) to be observed on the second setpoint value, typically the minimum;

Manual selection of next action (tolerances): the user can select which action of the multi-action sequence to have the controller execute in the event that the *Pattern* action is interrupted by the monitoring of these tolerances.

DATA SAVING

In this action the data saving settings are more complex than in the standard version because the *Pattern* action is typically articulated into lots of cycles and therefore data need to be decimated.

Data saving: if activated, the action saves its data; otherwise, the data acquired during this action are not stored in the TDMS file;

Saving group name: if the user writes some text here, this becomes part of the data groups dedicated to the current action on the results file, together with the action label;

Include action label in group name: if this option is active, the label of the action becomes part of the group name, otherwise it is ignored;

Include execution nr in group name: if this option is active, the number of executions of the action becomes part of the group names, otherwise it is ignored: the number of executions may be useful to separate the data acquired during the various iterations of a single action, if the test sequence is more complex than a classic top-down one and includes loops and repetitions of some sections

Sampling frequency: this drop-down menu is effective only if saving is activated for the action, and sets the sampling frequency of the data for logging during that action; the frequency can be chosen only among the given values;

Data concatenation into same saving group: if this option is activated, all the data of the entire cycles saved during the action are collected in the same group; the maximum and minimum data group remains separated in any case;

Saving mode: the cycles to be saved can be distributed through the test duration in multiple ways:

1. Linear: the saving interval is uniform: for instance, if 10% is selected then one cycle each 10 is going to be saved; if 1%, then one cycles each 100 is going to be saved;
2. Logarithmic: the percentage of cycles to be saved is distributed so that the interval between two consecutive cycles saved is shorter at the beginning of the test, and gradually increases as the test evolves
3. Exponential: the same concept as the logarithmic saving, but with an opposite trend: less cycles saved at the beginning of the test, more at the end;
4. On-off: the on-off mode gets the specified saving percentage by logging some consecutive cycles for each interval, in keeping with the *on-off saving benchmark* parameter (see description), instead of a single one as the linear mode does;
5. Multi-linear: it is possible to configure multiple linear saving rules, each of them to be applied within a specified range of cycles;
6. Manual: the user specifies manually the numbers of the cycles to be saved.

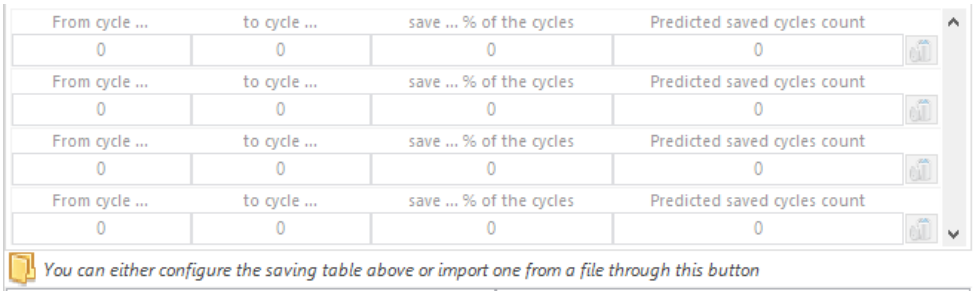
Cycle number or percentage to save: this parameter is visible only for linear, logarithmic, exponential and on-off saving, and depending on the chosen unit of measurement it is interpreted as a percentage of the target number of cycles or as a “pure” number of cycles to be saved out of the total; the *auto-adjust* checkbox provides the possibility to let the software keep this value automatically updated according to the target cycle count configured for the action (if the checkbox is active, any modification to the numeric value is rejected);

Logarithmic/exponential saving parameter: this parameter is visible only if the selected saving mode is logarithmic or exponential; it sets the ratio between the maximum and the minimum saving interval

of the test (respectively the last and the first in case of logarithmic saving, and the first and the last for exponential); in other words, this parameter establishes how much evident the logarithmic or exponential characterization of the saving process must be (it has no effect for linear and on-off savings); for example, if a value of 20 is selected for an exponential saving mode with 5% of the cycles to be saved, the intervals distribution will be calculated in order to get 5% of the cycles logged with the first interval being 20 times bigger than the last; on the right side of this field a button is provided to open a preview window where two graphs are displayed, showing the evolution of the number of savings and of the saving interval all through the test duration, comparing all the different saving methods;

On-off saving benchmark: this parameter is visible only the selected saving mode is on-off and, together with the percentage of cycles to save, defines how savings have to be organized; for example:

- If 5% of the cycles is selected for logging and the benchmark is set to 100, then the program saves 5 cycles each 100 benchmarked against intervals of 100 cycles, therefore the cycles to be saved are 1..5, 101..105, 201..205 and so on;
- If 5% of the total cycles is selected for logging and the benchmark is set to 200, then the program saves 5 cycles each 100 benchmarked against intervals of 200 cycles (i.e. 10 out of 200), therefore the cycles to be saved are 1..10, 201..210, 401..410 and so on.



Multi-linear saving table: it is visible only if the selected saving mode is *Multi-linear*; each row includes the starting and finishing cycles to configure an interval, and a target percentage of cycles to be saved during that interval; finally, the *Predicted saved cycles count* provides a preview of how many cycles are going to be saved within each interval using the specified settings; it is also possible to import a multi-linear saving table from an external file through the folder button displayed on the bottom of the table: the file has to be a txt or csv document with three columns (first column: initial cycle of the interval; second column: final cycle of the interval; third column: saving percentage to be used on the specified interval), and the column separator has to be a “tab” character;

Cycles to be saved: this parameter is visible only if the selected saving mode is manual, and is represented by a text field where to type the exact numbers separated by a comma or by two points to set an entire interval: for example, writing “1,10,50..60,100” means commanding the program to save cycles 1, 10, from 50 to 60 and 100; if the *automatic saving percentage regulation* is activated, this parameter is

automatically adjusted every time a setting is modified on the interface, according to the target cycle count forecast for the action;

Max and mins saving percentage or nr.: the program detects a maximum and a minimum value for load and position (and eventually for other channels, depending on the peak-valley acquisition configuration) automatically for each cycle of the test, therefore through this feature it is possible to choose how many of these peaks and valleys will be saved with respect to the total cycle number of the test; this parameter works in the same way as *Cycle number or percentage to save*, since – differently from the cycles logging – the maximums and minimums logging always follows a linear law (the saving interval is uniform for maximums and minimums).

ADVANCED SETTINGS

Dynamic inertia load compensation: this parameter activates or deactivates an automatic algorithm to compensate the inertis of the known actuator mass while executing the profile, to improve the motion precision and performances;

Setpoint application without initial transition: this parameter does not intervene on position-controlled fatigue cycles; if this control is set to ON, the wave amplitude is instantly set to the configured value without an initial transition that is otherwise done to improve the precision on the setpoint achievement;

Peak-valley algorithm activation: this option commands the controller to activate or deactivate the peak-valley algorithm to control the correspondence between the configured target maximum and minimum and the feedback results, applying corrections to minimize the error;

Gradual transition when frequency is changed: if this option is set to ON, when the action frequency is changed during its execution the transition phase is progressive, whereas if the configured value is OFF the frequency change is immediate;

Gradual approach to the initial frequency: if this value is set to ON, at the beginning of the action the controller gradually increases the cycles frequency to progressively get to the target period; if it is OFF, the target frequency is immediately applied starting from the very first cycle;

Peak-valley regulation gain: this is the proportional gain for the peak-to-peak algorithm that regulates the load-controlled fatigue tests: the higher is this parameter, the bigger are the corrections commanded by the controller to the target setpoint of the next cycle basing on the difference between the target peak load and the value measured in the previous cycle;

Peak-valley regulation saturation: if the dedicated checkbox is activated, this is the maximum position correction that the controller can command in one cycle; a value of 0 means that the controller uses its own default settings for this;

Cycle count during the initial transient: if this option is activated, the controller starts to increment the cycle counter at the very first waveform executed; otherwise, the count starts only once the commanded

maximum and minimum values are met (if the test is controlled in load, it takes some preliminary cycles to reach the target values after the calibration);

Setpoint tolerances configuration mode: a tolerance is applied to each of the setpoint values to determine when the transition phase is finished and the steady-state mode begins, so that the cycle count starts; with this option it is possible to choose whether the tolerance has to be the same for the two setpoint values, and calculated basing on the range ("percentage of configured setpoint range"), or specific for each of the two setpoint values ("point values for max and min");

Tolerance on range: if the tolerance is calculated basing on the range, this is the percentage of the difference between the maximum and minimum value of the setpoint which defines the tolerance to be applied to both of the values: for instance, if the maximum setpoint value is 1000 and the minimum is 100, with a percentage of 0,5%, the tolerance will be 0,5% of (1000 – 100) → ±4,5 around each of the setpoint vaues;

Tolerance on value 1 (max): if the tolerance is set to be specific for each of the setpoints, this is the one for the maximum value (if X is the value written, the tolerance range will be ±X around the setpoint value);

Tolerance on value 2 (min): if the tolerance is set to be specific for each of the setpoints, this is the one for the minimum value (if X is the value written, the tolerance range will be ±X around the setpoint value);

Electric noise compensation: this option allows to activate a functionality which compensates the electric noise on the feedback channel;

Electric noise amplitude: if the electric noise compensation is active, this is the amplitude of the noise to compensate;

Cycle count management: this option determines the behaviour of the action cycle counter:

- Resume at start & reset at successive iterations: when the action is executed for the first time since the test start, the cycle counter is preserved; if the development of the sequence leads to performing the same action again (for instance, through a for loop or a *GoTo* action), the cycle counter is reset in the successive beginnings of the action;
- Always reset when entering action: the cycle counter is reset every time the action begins;
- Never reset when entering action: the cycle counter is never reset when the action begins.

CALIBRATION SETTINGS

Calibration activation: if this option is set to ON, the actuator executes a preliminary calibration before starting the fatigue cycles of the *pattern* action; it is recommended to always use the calibration for actions with load and external control, whereas it is not necessary for position-controlled cycles; the calibration can be ran in standard mode (suitable for specimens which rigidity does not vary a lot) or in

non-linear mode (which is recommended if the specimen shows significant rigidity variations within the stroke of the test);

Setpoint range % to use for calibration: if the calibration is active, this parameters sets the range to be used for the procedure, with respect to the setpoint range: for instance, if the fatigue cycles have to be carried out between 100 and 1000 N (load control) and 50% is selected as the calibration range, the preliminary movement is executed between 225 and 775 N (average load = 550 N; load range = 900 N; calibration range = 50% of load range = 450 N; the calibration range is distributed equally starting from the average value, so the maximum calibration load is *average + half calibration range* = 550 + 450/2 = 775 N, and the minimum calibration load is *average – half calibration range* = 550 – 450/2 = 225 N);

Speed during calibration: this is the speed of the actuator during the calibration movements; if the value is set to 0, then the controller uses an internal default value.

SWEEP MODE

Sweep mode activation: if this option is set to ON, the Pattern action can be executed as a frequency sweep, defining parameters so that a certain number of cycles is executed for each frequency level from the beginning to the end; all the following parameters are visible only if this option is activated;

Starting value: initial frequency of the action; this value overwrites the *Frequency* configuration of the main settings;

Final value: final frequency of the action, the last frequency value where the sweep ends;

Increment: the difference between the frequency of two consecutive steps of the sweep;

Cycle number for each level: this parameter sets how many cycles have to be executed at each frequency level;

Go on at sweep end until target cycles runout: if this setting is set to ON, when finishing the number of cycles stated in *cycle number for each level* at the highest frequency, the actuator keeps running the action at that frequency aiming at the cycle count configured in the main settings; otherwise, the action is stopped at the end of the sweep regardless of the cycle count written in the main settings.

4.4.5. Creep action

A *Creep* action is a motion that aims at a specific load value gradually reducing the actuator speed to reach the target value as precisely as possible; once this is done, the load is maintained for the time set by the user.

The specific parameters that can be configured for a *Creep* action are:

MAIN SETTINGS

Test control mode: load or position, it is the parameter that has to be controlled during the test;

Position control mode: the user can choose between position and displacement for the setting of the target position, if the position control is selected, and for the visualization and saving of that coordinate during the test;

Target speed: limit speed of the axis in the movement towards the target load or position; the actual speed value decreases with the progression of the test because of the control algorithms that allow, as the creep test protocol requires, to reach the target load or position precisely; if the user tries to start a test with a null speed value the program stops the procedure and opens a warning popup window;

Target force rate: this parameter represents the desired variation rate for the load during the movement process that leads to the accomplishment of the target load, and must be specified as an absolute value (i.e. without sign); if the checkbox placed between the description and the value is not selected, the software takes into account only the target speed for the motion;

Target load: defined load for the creep operation (this setting is visible only if the test is load-controlled);

Target position: defined position for the test (this setting is visible only if the test is position-controlled);

Total time: creep duration, starting from the moment when the target load or position is reached (not from the instant when the test is launched);

Material type: this is an enumerative parameter that permits a basic self-regulation of the machine with respect to the kind of material that is being tested (the user has to select the option which is the most similar to the actual material).

ACTION END CONDITIONS

Maximum displacement: maximum acceptable displacement of the actuator during the test procedure with respect to the initial position when the test is ran;

Break threshold (force variation): if the machine detects a load variation higher than this value between two consecutive scans, the test is automatically interrupted because a fracture is identified.

ADVANCED SETTINGS

Dead band: tolerance percentage on target load or position, within which the motion phase of the creep is concluded and the holding time count starts;

PID proportional gain: in keeping with the definition of this parameter in a PID controller, it is a constant value used to multiply the error signal in order to correct the setpoint value of load or position; see paragraph 4.2.1 for a more detailed explanation and some usage hints;

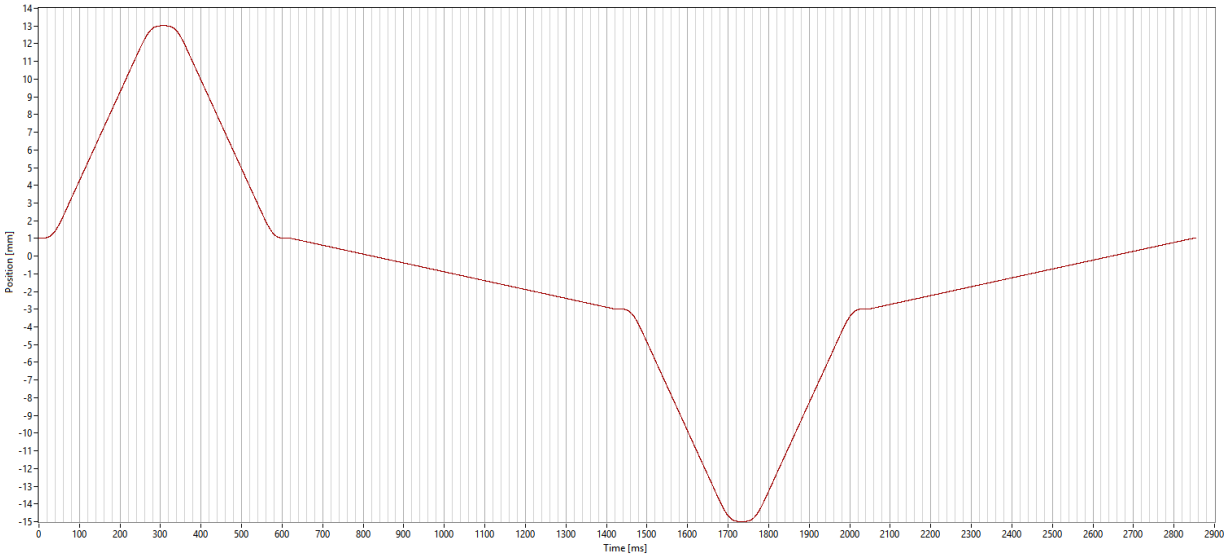
Ti – PID integral time: the integral time of the PID controller to be used during the action execution;

Automatic material characterization: activates or deactivates an automatic procedure, performed at the beginning of the test, through which the machine self-regulates with respect to the rigidity of the specimen. The procedure is composed of a movement at controlled speed (which value is automatically obtained combining some test parameters such as material type and geometry properties) until the load reaches 5% of the difference between target value and initial value. This parameter is visible only if the test is load-controlled and, if deactivated, requires the user to precisely set the *Material rigidity* field;

Material rigidity: rigidity value, expressed in N/m or other commensurable unit, that applies for the specimen; this field is visible only if the automatic material characterization is visible and deactivated, but in such a case it is mandatory to insert a valid value; if 0 is left as input, the program automatically re-activates the automatic characterization procedure even if the user deselected it. The rigidity has to be inserted as an absolute value, or in other words without sign.

4.4.6. GasTest action

A *GasTest* action is a particular motion sequence described by this position plot:



The two regions where speed is higher are called *extra strokes*; speeds and strokes for extra strokes and for the rest of the action can be set separately.

The specific parameters that can be configured for a *GasTest* action are:

MAIN SETTINGS

Procedure control mode: it is possible to choose whether to control speed or frequency for the measurement phase (not for the extra-strokes): in case frequency control is selected, the chosen frequency value specifies the theoretical overall duration of the two movements that compose the measurement cycle;

Speed in measurement cycle: the *measurement cycle* is the part of the gas test that is between the two extra strokes and after the second extra stroke with respect to the graph above; it is called measurement cycle because the calculated results only take into account this section of the cycle; this control is dedicated to set the speed of the actuator during these two movements;

Frequency: if *procedure control mode* is set to *frequency*, this is the reciprocal of the theoretical overall duration (not considering the acceleration and deceleration phases) of the two displacements that generate the measurement cycle;

Positive position/displacement for the gas test: this is the position or displacement value (depending on the setting of *position control mode*), with respect to the actuator coordinate when the gas test begins, that has to be considered as the upper target for the measurement cycle;

Negative position/displacement for the gas test (with sign): this is the position or displacement value (depending on the setting of *position control mode*), with respect to the actuator coordinate when the gas test begins, that has to be considered as the lower target for the measurement cycle; if negative, its sign has to be specified;

Extra-stroke over the maximum position: this is the displacement value, with respect to the *Positive position/displacement for the gas test*, that has to be considered as the upper limit for the extra stroke phase;

Extra-stroke under the maximum position (with sign): this is the displacement value, with respect to the *Negative position/displacement for the gas test*, that has to be considered as the upper limit for the extra stroke phase; if negative, its sign has to be specified;

Speed in extra-strokes: this is the actuator speed while moving within upper and lower extra strokes;

Position control mode: it is possible to choose whether the coordinates for the measurement cycle (positive and negative *position/displacement for gas test*) have to be specified in terms of position (absolute movements with respect to the current coordinate system) or displacement (relative movements from the starting position);

Motion sequence: the user can choose whether to execute first the movement in positive or negative direction in the measurement cycle (warning: "positive direction" means that the position value increases, "negative direction" means that the position value decreases; the matching between the two direction and tension/compression depends on the machine configuration).

ACTION END CONDITIONS

Break threshold: this is a percentage, calculated with respect to the current load, that defines a limit difference between two consecutive load values to detect a fracture through a dedicated automatic algorithm;

Timeout: maximum action duration.

GAS FORCE ACQUISITION PROCEDURE

Number of points for gas force calculation: it is possible to set one or more measurement points for the gas force computation; if multiple points are selected, the final gas force output is the average of the values measured in the various points set;

Automatic distribution of measurement points: if this setting is turned on, the coordinates of the measurement points are automatically distributed through the travel of the measurement movement; if it is off, the user has to type the coordinates manually;

Measurement point N: N stands for the index of the measurement point; these controls are displayed only if the *automatic distribution of measurement points* is deactivated, and are the coordinates of each measurement point;

Gas force measurement pauses duration: if this value is set to 0, the movements of the measurement cycle are never interrupted; if a non-null value is configured, the actuator stops for this time interval at each measurement point coordinate while moving in the measurement cycle;

Average force value calculation in stops: only visible if *gas force measurement pauses duration* isn't equal to 0; if it is set to ON, the software calculates the average of a subset of the force values acquired

during the movement interruption, whereas if the option is OFF then the output of each measurement point is the last force value collected before restarting the movement;

Interval for average calculation on load value: this control is only visible if *average force calculation in stops* is active, and its value has to be less or equal to the *gas force measurement pauses duration*; this is the duration of the subset of data to average, taken from the latest data acquired during the pause: for instance, if the pause is set to 1 second and this interval is set to half a second, the data taken into consideration for the calculation are those acquired in the second half of the pause.

On the right of the action configuration interface, a preview graph displays the motion sequence of the gas test procedure as it would be with the current set of parameters.

4.4.7. Warmup action

A *Warmup* action is a sequence of cycles which configuration is simpler than in the *pattern* action; the warmup is typically dedicated to increase the temperature of the specimen in case of needed, before starting the rest of the test.

The specific parameters that can be configured for a *Warmup* action are:

MAIN SETTINGS

Target speed: this is the speed value that the controller uses for the position-controlled, triangular-waved profile;

Positive position/displacement: this is the position or displacement value in positive direction: if position, it has to be written with respect to the actuator coordinate system in use; if displacement, with respect to the action initial position;

Negative position/displacement: this is the position or displacement value in negative direction: if position, it has to be written with respect to the actuator coordinate system in use; if displacement, with respect to the action initial position; if negative, its sign has to be specified;

Position control mode: it is possible to choose whether the coordinates (positive and negative *position/displacement*) have to be specified in terms of position (absolute movements with respect to the current coordinate system) or displacement (relative movements from the starting position).

ACTION END CONDITIONS

Break threshold: this is a percentage, calculated with respect to the current load, that defines a limit difference between two consecutive load values to detect a fracture through a dedicated automatic algorithm;

Timeout: maximum action duration;

Maximum cycle count: if activated, stop the warmup procedure after the specified number of cycles if no other exit condition occurred before.

ADVANCED SETTINGS

Cycle count management: this option determines the behaviour of the action cycle counter:

- Resume at start & reset at successive iterations: when the action is executed for the first time since the test start, the cycle counter is preserved; if the development of the sequence leads to performing the same action again (for instance, through a for loop or a *GoTo* action), the cycle counter is reset in the successive beginnings of the action;
- Always reset when entering action: the cycle counter is reset every time the action begins;
- Never reset when entering action: the cycle counter is never reset when the action begins.

4.4.8. ProfileTracking action

A *ProfileTracking* action is the reproduction of a random profile which can be configured point-by-point or imported from a file.

The specific parameters that can be configured for a *ProfileTracknig* action are:

MAIN SETTINGS

Cycle number: the target number of times the configured profile has to be repeated;

Test control mode: it is the parameter that has to be controlled during the load session;

External control channel: choosing among the available analogue inputs, it is the channel to use as feedback in case the *Test control mode* is set to *External*;

Profile generation mode: the user can choose between different configurations for the profile: it can be manually set point by point or imported from an external text file (see the following paragraphs); starting with version 7.1 of the program, only if this feature is activated, the "imported from file with local max-min only" profile is also available, which involves importing a file that does not contain the entire profile to be followed but only the peaks and valleys that characterise it, leaving the controller to manage the junctions; according to what the user selects here, the software chooses which interface to display on the second tab of the action configuration window.

ACTION END CONDITIONS

Acceptable % rigidity variation: this is the key parameter for break detection: if it is enabled through the dedicated checkbox, the machine stops if a rigidity variation greater than the threshold configured here is detected between two consecutive cycles; this parameter can be modified also when a test is running; next to the numeric value there is a button that displays the graph with the current rigidity profile of the specimen, updated in real time during the test; through this it is immediate to monitor the evolution of the rigidity and adjust the threshold value if needed;

Profile following error monitoring: if this option is activated, *Max acceptable following error* and *Reference time for following error* are displayed and used by the controller to set another possible break detection criterion;

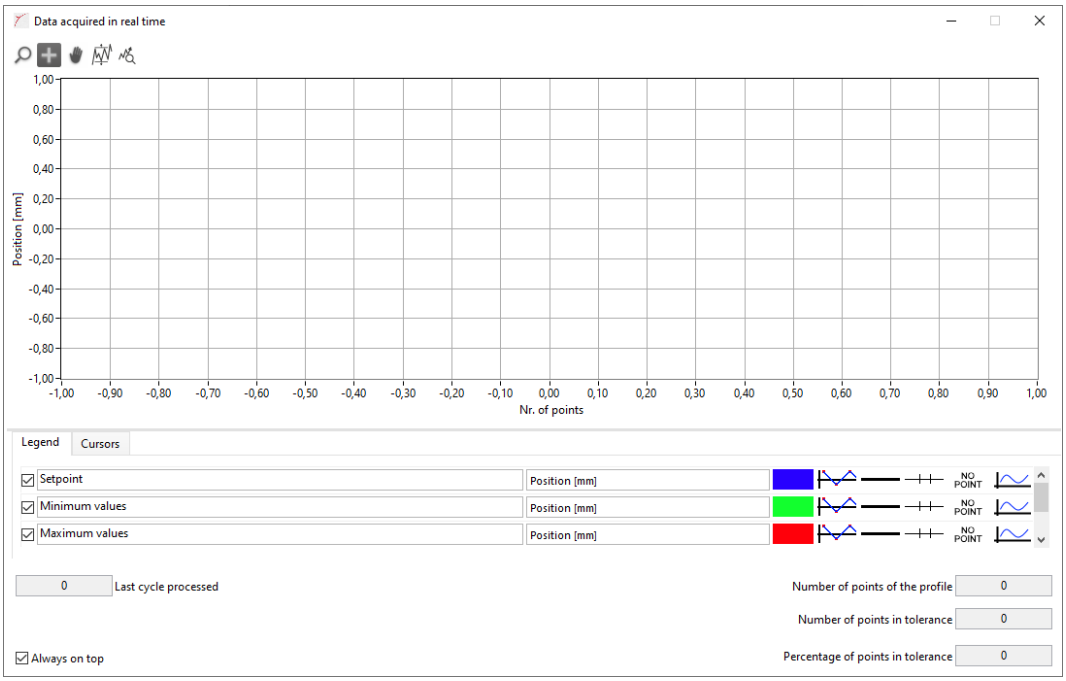
Max acceptable following error and Reference time for following error: the automatic break detection condition consists in an overall following error that becomes greater than the specified *Max acceptable* value within the *Reference time interval*;

Arbitrary channel variation monitoring: if this option is activated, the following three parameters are displayed and taken into consideration by the controller to set another possible break detection criterion;

Channel to monitor: the user has to choose a channel, among those that are available, that will be subject to continuous monitoring through the *Pattern* action in order to detect if the condition stated by *Threshold variation for break detect.* and *Reference time interval* is intercepted;

Threshold variation for break detect. (with sign) and Reference time interval: the automatic break detection condition consists in a variation of the selected channel, within the *Reference time interval*, that is greater or equal to the specified *Threshold*;

Nr of points in tolerance below which action ends: this option is only visible if the profile is imported from an external file with the maxima and minima only; after the steady state is reached, for each complete cycle that is executed (i.e. for each complete reproduction of the profile that includes all the points defined in the file), the software calculates for how many of the points the feedback value was within the range defined by the tolerances specified in the file, and stops the *ProfileTracking* action if the percentage of "good" points is less than the value specified in this control; If the reference file does not contain valid tolerance values, this parameter plays no role; on the right is a button  that opens a window containing a graph with a comparison between the measured points and their respective tolerance ranges, as well as an indication of how many points were within this range (this information is updated every cycle):



DATA SAVING

In this action the data saving settings are more complex than in the standard version because the pattern action is typically articulated into lots of cycles and therefore data need to be decimated.

Data saving: if activated, the action saves its data; otherwise, the data acquired during this action are not stored in the TDMS file;

Saving group name: if the user writes some text here, this becomes part of the data groups dedicated to the current action on the results file, together with the action label;

Include action label in group name: if this option is active, the label of the action becomes part of the group name, otherwise it is ignored;

Include execution nr in group name: if this option is active, the number of executions of the action becomes part of the group names, otherwise it is ignored: the number of executions may be useful to separate the data acquired during the various iterations of a single action, if the test sequence is more complex than a classic top-down one and includes loops and repetitions of some sections

Sampling frequency: this drop-down menu is effective only if saving is activated for the action, and sets the sampling frequency of the data for logging during that action; the frequency can be chosen only among the given values;

Data concatenation into same saving group: if this option is activated, all the data of the entire cycles saved during the action are collected in the same group; the maximum and minimum data group remains separated in any case;

Saving mode: the cycles to be saved can be distributed through the test duration in multiple ways:

1. Linear: the saving interval is uniform: for instance, if 10% is selected then one cycle each 10 is going to be saved; if 1%, then one cycles each 100 is going to be saved;
2. Logarithmic: the percentage of cycles to be saved is distributed so that the interval between two consecutive cycles saved is shorter at the beginning of the test, and gradually increases as the test evolves
3. Exponential: the same concept as the logarithmic saving, but with an opposite trend: less cycles saved at the beginning of the test, more at the end;
4. On-off: the on-off mode gets the specified saving percentage by logging some consecutive cycles for each interval, in keeping with the *on-off saving benchmark* parameter (see description), instead of a single one as the linear mode does;
5. Multi-linear: it is possible to configure multiple linear saving rules, each of them to be applied within a specified range of cycles;
6. Manual: the user specifies manually the numbers of the cycles to be saved.

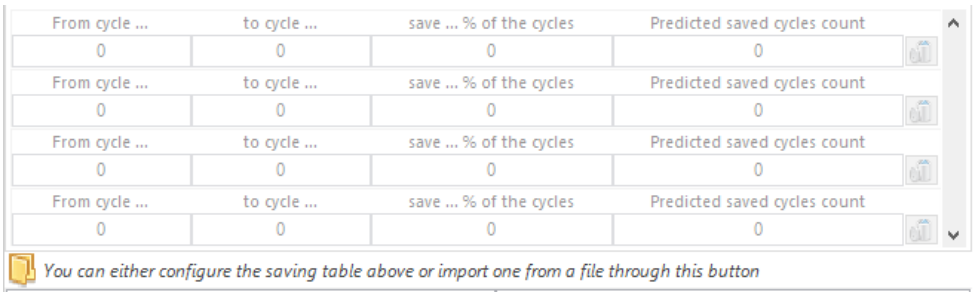
Cycle number or percentage to save: this parameter is visible only for linear, logarithmic, exponential and on-off saving, and depending on the chosen unit of measurement it is interpreted as a percentage of the target number of cycles or as a “pure” number of cycles to be saved out of the total; the *auto-adjust* checkbox provides the possibility to let the software keep this value automatically updated according to the target cycle count configured for the action (if the checkbox is active, any modification to the numeric value is rejected);

Logarithmic/exponential saving parameter: this parameter is visible only if the selected saving mode is logarithmic or exponential; it sets the ratio between the maximum and the minimum saving interval of the test (respectively the last and the first in case of logarithmic saving, and the first and the last for exponential); in other words, this parameter establishes how much evident the logarithmic or exponential characterization of the saving process must be (it has no effect for linear and on-off savings); for example, if a value of 20 is selected for an exponential saving mode with 5% of the cycles to be saved, the intervals

distribution will be calculated in order to get 5% of the cycles logged with the first interval being 20 times bigger than the last; on the right side of this field a button is provided to open a preview window where two graphs are displayed, showing the evolution of the number of savings and of the saving interval all through the test duration, comparing all the different saving methods;

On-off saving benchmark: this parameter is visible only the selected saving mode is on-off and, together with the percentage of cycles to save, defines how savings have to be organized; for example:

- If 5% of the cycles is selected for logging and the benchmark is set to 100, then the program saves 5 cycles each 100 benchmarked against intervals of 100 cycles, therefore the cycles to be saved are 1..5, 101..105, 201..205 and so on;
- If 5% of the total cycles is selected for logging and the benchmark is set to 200, then the program saves 5 cycles each 100 benchmarked against intervals of 200 cycles (i.e. 10 out of 200), therefore the cycles to be saved are 1..10, 201..210, 401..410 and so on.



Multi-linear saving table: it is visible only if the selected saving mode is *Multi-linear*; each row includes the starting and finishing cycles to configure an interval, and a target percentage of cycles to be saved during that interval; finally, the *Predicted saved cycles count* provides a preview of how many cycles are going to be saved within each interval using the specified settings; it is also possible to import a multi-linear saving table from an external file through the folder button displayed on the bottom of the table: the file has to be a txt or csv document with three columns (first column: initial cycle of the interval; second column: final cycle of the interval; third column: saving percentage to be used on the specified interval), and the column separator has to be a “tab” character;

Cycles to be saved: this parameter is visible only if the selected saving mode is manual, and is represented by a text field where to type the exact numbers separated by a comma or by two points to set an entire interval: for example, writing “1,10,50..60,100” means commanding the program to save cycles 1, 10, from 50 to 60 and 100; if the *automatic saving percentage regulation* is activated, this parameter is automatically adjusted every time a setting is modified on the interface, according to the target cycle count forecast for the action;

Max and mins saving percentage or nr.: the program detects a maximum and a minimum value for load and position (and eventually for other channels, depending on the peak-valley acquisition configuration) automatically for each cycle of the test, therefore through this feature it is possible to choose

how many of these peaks and valleys will be saved with respect to the total cycle number of the test; this parameter works in the same way as *Cycle number or percentage to save*, since – differently from the cycles logging – the maximums and minimums logging always follows a linear law (the saving interval is uniform for maximums and minimums).

ADVANCED SETTINGS

Cycle count management: this option determines the behaviour of the action cycle counter:

- Resume at start & reset at successive iterations: when the action is executed for the first time since the test start, the cycle counter is preserved; if the development of the sequence leads to performing the same action again (for instance, through a for loop or a *GoTo* action), the cycle counter is reset in the successive beginnings of the action;
- Always reset when entering action: the cycle counter is reset every time the action begins;
- Never reset when entering action: the cycle counter is never reset when the action begins.

Setpoint gradient to use for connections: this parameter is only visible if you choose to import the profile from a file with only the maximums and minimums; the wording "setpoint gradient" refers to a derivative before the feedback channel (speed in the case of a profile in position, force gradient for a profile in load...), and the numerical value to be set is a percentage where 100% represents the maximum speed with which the actuator can connect two consecutive points of the commanded profile, compatible with the mechanical characteristics of the actuator;

Peaks correction algorithm activation: this parameter is only visible if you choose to import the profile from file with only the maximums and minimums; if it is activated, the controller calculates and applies for each point of the profile a correction according to the deviation detected in the previous cycle between the commanded value and the value obtained, in order to bring the latter as close as possible to the former;

kp for peaks correction calculation: this parameter is only visible if you choose to import the profile from a file with only maxima and minima and activate the peak correction algorithm (see previous parameter); it represents the proportional gain of the algorithm that calculates the corrections to be made to the profile points to make the measured value as close as possible to the setpoint;

Peak-valley regulation saturation: this parameter is only visible if you choose to import the profile from a file with only maxima and minima; it automatically assumes the same unit of measurement as the feedback channel and, if activated by means of the appropriate tick, represents the maximum correction that the controller can apply to each point of the profile;

Min nr of points in tolerance for steady state: this parameter is only visible if you choose to import the profile from a file with only the maximums and minimums; when an action of this type begins, for the controller to decree entry into steady state, it is necessary to have reproduced a complete profile in which the measured value has been within the tolerance range established in the file for a percentage of points

greater than or equal to this value; if the reference file does not contain valid tolerance values, this parameter has no effect and the state is considered steady state from the first cycle.

Hybrid control mode with adaptive algorithm: this option is only visible if the profile is in load control and imported from file, and allows the use of a hybrid load-position control and an adaptive algorithm that are particularly suitable for cases where the specimen is not always in contact with the gripping system.

AUTO-TUNING SETTINGS

Auto-tuning activation: if this option is set to ON, the actuator executes a preliminary calibration before starting the fatigue cycles of the *pattern* action; it is recommended to always use the calibration for actions with load and external control, whereas it is not necessary for position-controlled cycles; the calibration can be ran in standard mode (suitable for specimens which rigidity does not vary a lot) or in non-linear mode (which is recommended if the specimen shows significant rigidity variations within the stroke of the test);

Setpoint range % to use for auto-tuning: if the calibration is active, this parameters sets the range to be used for the procedure, with respect to the setpoint range: for instance, if the fatigue cycles have to be carried out between 100 and 1000 N (load control) and 50% is selected as the calibration range, the preliminary movement is executed between 225 and 775 N (average load = 550 N; load range = 900 N; calibration range = 50% of load range = 450 N; the calibration range is distributed equally starting from the average value, so the maximum calibration load is *average + half calibration range* = 550 + 450/2 = 775 N, and the minimum calibration load is *average – half calibration range* = 550 – 450/2 = 225 N);

Speed during auto-tuning: this is the speed of the actuator during the calibration movements; if the value is set to 0, then the controller uses an internal default value.

In the event that the *ProfileTracking* action is used to reproduce a profile imported from a file with only maximums and minimums, the above parameters are part of the section called "Auto-tuning preliminary to the profile reproduction" and there is another one called "Profile auto-tuning" that includes the following settings:

Auto-tuning coefficients to use at first test start: “profile auto-tuning' refers to the automatic adjustment by the controller of the profile to be controlled, so that the measured feedback values conform as closely as possible to the profile points contained in the reference file; this adjustment is carried out by means of correction coefficients to be applied to each profile point, which the controller calculates and updates cycle by cycle; when a *ProfileTracking* action has never been performed before, it is possible to choose whether these coefficients are to be used:

- Default values: in this case, the auto-tuning procedure starts over;
- The last values acquired in steady state during the previous test: This option only has significance if the test specimen being configured was obtained via the "add specimen"

function from a test that had already been performed, and allows you to restart from the auto-tuning coefficients that were stored at the steady-state entry of the previous test; this option may be useful in the event that the specimen of the test that has just been completed has altered its mechanical characteristics due to the stresses imposed by the test, and for the new specimen it is useful to recover the auto-tuning parameters obtained from the previous specimen before it degraded;

- The last values acquired during the previous test: in this case, the coefficients transmitted to the controller for auto-tuning of the new test are the last ones that were stored during the previous test; this option may be useful in the event that the new test created represents a continuation of the one that was performed before.

Waiting between achievement of steady-state and storage of auto-tuning coefficients: at the start of a new test, when the controller establishes that the steady state is reached, the programme starts a timer that waits for the time interval specified here before storing the self-tuning coefficients relating to the steady state (those that can be retrieved in a new test if the second option is selected in the previous parameter); once these coefficients are stored, they cannot be overwritten either manually or by stopping and resuming the same test.

ADAPTIVE ALGORITHM

This section is only visible if the parameter *Hybrid control mode with adaptive algorithm* is set to ON, and contains the variables that parameterise the adaptive algorithm itself:

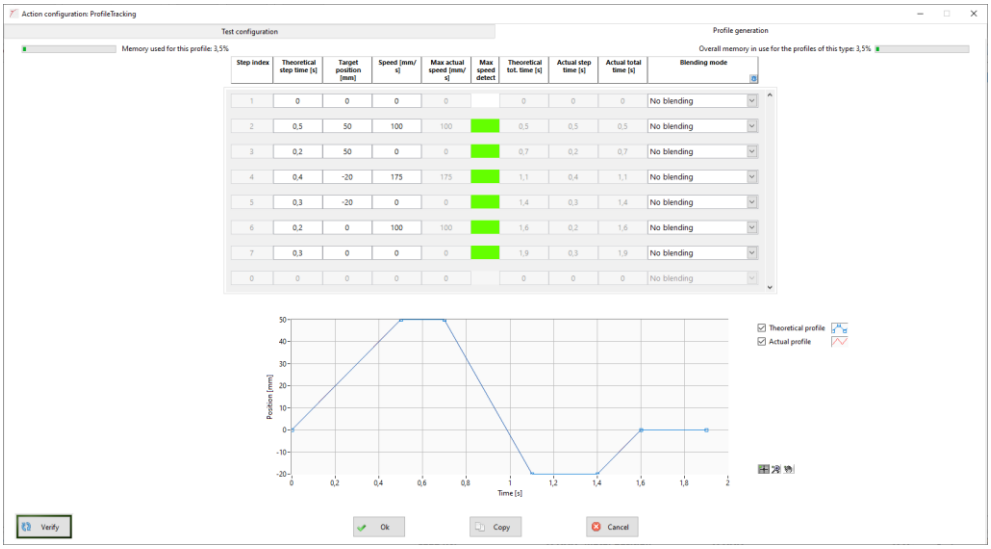
Nr of points for profile approximation (0=default): the controller approximates the commanded profile with a certain number of points, which is set via this parameter;

Nr of points for moving avg window (0=default): a moving average filter is applied to the profile to be reproduced, the window width of which can be configured;

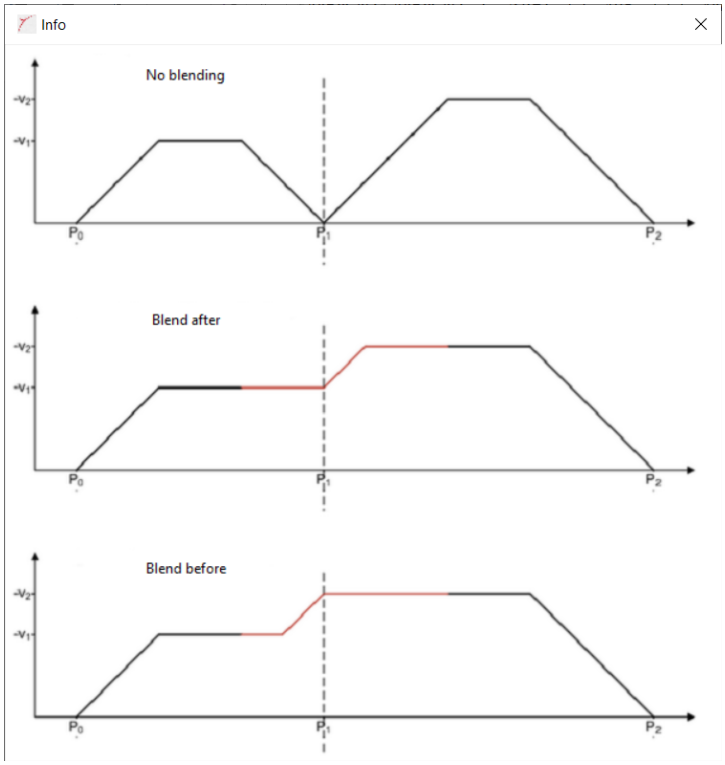
Adaptive algorithm gain: this is the proportional gain of the error correction of the adaptive algorithm: the higher the value, the faster the correction; the lower the value, the smoother the correction.

The *ProfileTracking* action provides the possibility to reproduce a random profile that can be configured point-by-point or imported from an external file. In the first case, the user manually configures the profile to follow by typing the setpoint and the duration (or, alternatively for position feedback, the speed) of each motion step (the *Theoretical step time* for the first row has to be 0, since that is the starting point).

The following picture shows the interface to configure a point-by-point position profile:



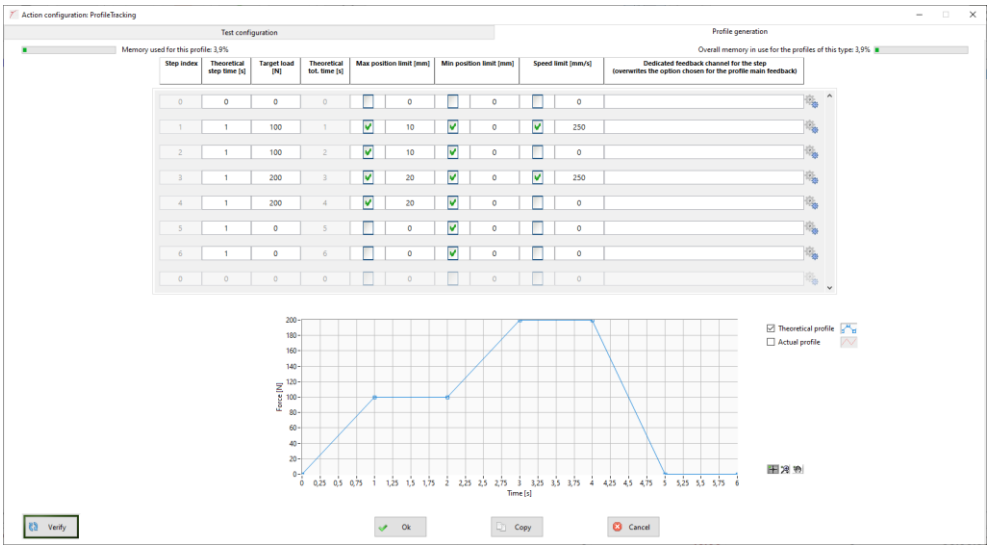
Starting with version 7.0 of the software, only for position-controlled profiles and only if this functionality is activated in the programme, there is a function called “blending”: for each point of the profile that is not the first or the last, it is possible to choose between “no blending”, “blend before” and “blend after”, with the three options having the following meanings, exemplified by this speed-time diagram:



- No blending: the transition from the previous to the next step takes place with a very short stop of the actuator at the set point;
- Blend before: the transition from the previous step to the next is made by linking the speeds without ever stopping, and by ensuring that at the desired position the speed already has the value of the movement step to start;

- Blend after: the transition from the previous step to the next is made by linking the speeds without ever stopping and by ensuring that the transition from the speed of the previous step to the speed of the next step begins when the actuator is at the configured point.

If the profile feedback is not in position but in load or in an analog channel, the interface shows some additional options:



It is possible to notice some additional options:

Max position limit: this is the maximum position limit that must not be exceeded during the step; it is not a safety setting that interrupts the action, but a saturation value that determines the termination of the movement step even if the setpoint has not been reached; it is possible to activate or deactivate this option separately for each step of the profile by means of the appropriate tick;

Min position limit: same as above, for the minimum position;

Speed limit: this is the maximum speed value that the actuator cannot exceed during the step, even though theoretically the time set to go from the initial setpoint to the final setpoint would require it; again, there is a tick mark for activation or deactivation for each step;

Dedicated feedback channel for the step: by pressing the button with the gear icon, it is possible to configure steps within the profile in which the feedback channel is not the one configured in the action's primary settings, but another one chosen by the user from among those available; in this case, the setpoint value of the step is to be interpreted according to the dedicated feedback channel; it is therefore possible, for example, to configure a step in position within a profile controlled in load.

When the profile is modified, the user has to validate it by clicking on the *Verify* button on the bottom-left corner of the interface; if the frame around the button blinks in yellow, this means that there are pending changes waiting for a verification.

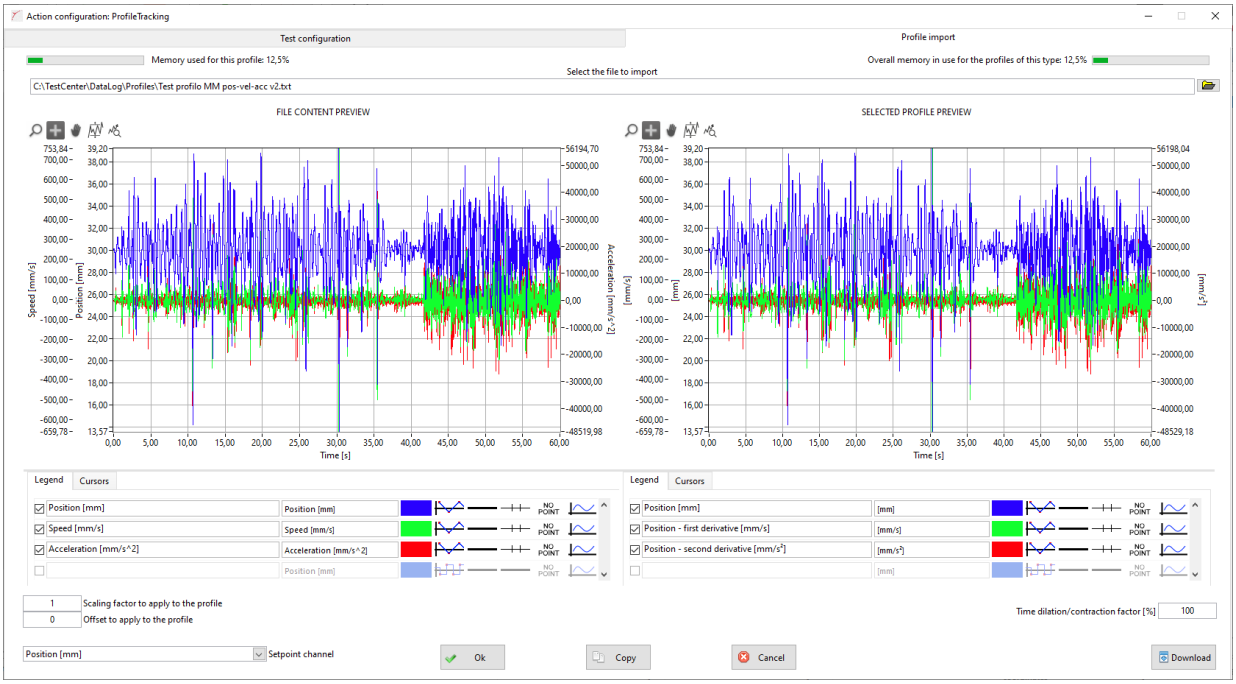
On the other hand, if the *Imported from file* option is selected for the profile, the user can select an external file including the profiles that will be imported; however, the file has to stick to some rules:

- The column separation character is a “tab”;
- The first row is dedicated to the names of each data series, and each name has to contain the unit of measurement between square brackets;
- The decimal separator can be . or , regardless of the computer regional settings;
- The first column has to be dedicated to time values;
- All the time values have to be separated by the same interval;
- The sampling frequency of the profile included in the file must not be less than 1/10 of the hardware clock frequency and must not be greater than the hardware clock frequency.

Time [s]	Position [mm]	Speed [mm/s]	Acceleration [mm/s^2]
0,000000	0,000000	3,457348	1,567385
0,000250	0,000864	3,458027	5,013567
0,000500	0,001729	3,460058	10,904501
0,000750	0,002595	3,463459	16,555132
0,001000	0,003461	3,468167	20,174880
0,001250	0,004329	3,473133	18,828381
0,001500	0,005197	3,477461	16,038316
0,001750	0,006067	3,481184	13,628971
0,002000	0,006938	3,484339	11,981218
0,002250	0,007809	3,487319	12,065128
0,002500	0,008682	3,490405	12,549581
0,002750	0,009555	3,493583	12,914615
0,003000	0,010428	3,496807	12,619482
0,003250	0,011303	3,499798	11,184852
0,003500	0,012178	3,502385	9,576338
0,003750	0,013054	3,504596	8,086077
0,004000	0,013931	3,506394	6,188283

Figure 4.1 - Example of acceptable file format

Once the file is imported, preview graphs are displayed as in the picture below:



The graph on the left side of the interface displays all the data sets found in the file; instead, the graph on the right provides the channel chosen as setpoint through the drop-down menu on the bottom-left corner of the interface (*Setpoint channel*) together with an estimation of its first and second derivative.

After importing the profile, it has to be downloaded to the field through the dedicated button on the bottom-right corner of the interface; if the download operation is pending, the frame around the button blinks in yellow. Alternatively, the software automatically executes the download if the Ok button is pressed.

In whichever way the profile is configured, on the top-left section of the profile generation interface there is a fill bar indicator stating how much of the available memory is being allocated for that single profile: in fact, the memory dedicated to store all the profiles in the controller is limited to 620000 points, so it is not possible to use infinite ProfileTracking actions; the memory used for each profile depends on its duration and its sampling frequency: in fact, 620000 points are needed for 155 seconds if the sampling rate is 4 kHz, but they last for 620 seconds if the sampling rate descends to 1 kHz. The profiles configured point-by-point have always the same sampling frequency as the controller loop (which is the maximum possible), whereas if the profile is imported from a file its sampling rate may also be smaller (the interpolation is automatically performed by the controller).

Starting with version 7.3 of the TestCenter, in the lower left-hand corner it is possible to see two numerical values with which there is the possibility of applying a scaling factor and an offset to the original profile: a law $y=mx+q$ is then defined in which y is the generic setpoint value in the profile that will be reproduced, x is the corresponding setpoint value in the original profile contained in the file, m is the scaling factor and q the offset. At the same level of positioning, but on the right-hand side of the interface, there is a time expansion or contraction factor, also called an override, which allows the profile to expand or contract in time according to the law

$$t' = \frac{t \cdot 100}{override\ [\%]}$$

where t' is indifferently the duration of the new profile or the generic time interval between two of its consecutive points, and t is the correspondent of the original profile. Therefore, override values greater than 100% increase the profile's dynamics, and lower values reduce it.

The latter parameter does not affect the constraint on the minimum sampling rate allowed for profiles imported from file, which always remains the same regardless of the configured override value. In any case, it is not possible to set values lower than 10%.

Finally, if the *ProfileTracking* action is used to reproduce a profile imported from file but with only the maximums and minimums, the interface for selecting and previewing the profile to be reproduced is very similar to that of the normal profile imported from file, with the path to the file to be imported, only a preview graph (there is no calculation of the first and second derivatives, because the profile is not entirely specified in the file) and the selection of the channel representing the setpoint in the case the file contains more than one set of data:

When a *ProfileTracking* action is used to reproduce a profile imported from a file containing only maxima and minima, the file must have the following characteristics:

- The column separator must be a tab character;
- The first line must be dedicated to the names of each data set (channel), and each name must contain the unit of measurement of the channel in square brackets;
- The decimal separator may be . or , regardless of the computer's regional settings;
- The time column shall not be provided: the speed of the actuator in connecting the various points is set by the "gradient of the setpoint to be used for connections" parameter, which is visible in the advanced settings;
- It is possible to import a file with only one column dedicated to the setpoint, or with tolerances as well: in this case, the first column must be dedicated to the setpoint, the second to the lower tolerance and the third to the upper tolerance;
- If you wish to group more than one set of data dedicated to a possible setpoint in the same file (the software allows you to choose which one to use each time), the presence of the tolerances is necessary: therefore, the scheme of the columns to follow is
    - Column 1: first setpoint channel;

- Column 2: lower tolerance associated with the first setpoint channel;
- Column 3: upper tolerance associated with the first setpoint channel;
- Column 4: second setpoint channel;
- Column 5: lower tolerance associated with the second setpoint channel;
- Column 6: upper tolerance associated with the second setpoint channel;
- ...

4.4.9. ExternalCmd action

An *ExternalCmd* action is a motion action in which the profile to follow is provided by an external control unit.

The specific parameters that can be configured for an *ExternalCmd* action are:

MAIN SETTINGS

Control mode: the user sets the feedback to use for the control;

Feedback channel: this control is visible only if the *control mode* is set to *external*, and allows to choose as feedback one of the various analog inputs available;

Reference setpoint channel: this is the analog input channel through which the setpoint is provided;

Sampling frequency of the reference: a drop-down menu allows to choose among various options for the frequency of the reference signal that is going to be provided;

Setpoint starting value: this is the value of the setpoint channel that the actuator has to reach prior to start following the profile commanded by the external device.

ACTION END CONDITIONS

Break threshold: this is a percentage, calculated with respect to the current load, that defines a limit difference between two consecutive load values to detect a fracture through a dedicated automatic algorithm;

Timeout: maximum action duration.

ADVANCED SETTINGS

Pointer code origin for READY state, Channel to use for READY state code, Pointer code origin for START command, Channel to use for START command code: please refer to the manufacturing company technical support for these parameters;

Initial and calibration speed: this is the speed of the initial motion;

Calibration activation: if this option is set to ON, the actuator executes a preliminary calibration movement at the beginning of the action;

Maximum calibration value: maximum setpoint value for the calibration movement;

Minimum calibration value: minimum setpoint value for the calibration movement.

4.4.10. ResetPosition action

This action allows to modify the custom position coordinate system of the actuator while a test is running.

> ***Caution:*** *this action does NOT change the position values specified for the following actions: so, for instance, if there is a Displacement action next and it is configured to move to 50 mm, this value is going to remain but the controller will interpret is with respect to the NEW coordinate system.*

The specific parameters that can be configured for a *ResetPosition* action are:

MAIN SETTINGS

Operation to execute: depending on what the user chooses, the action can reset the position at a certain value or even cancel any previous position reset and restore the absolute actuator coordinate system;

New value for the position: if the *operation to execute* is set to *position reset*, this numeric control allows to set the new position value that has to be given to the coordinate where the action is executed;

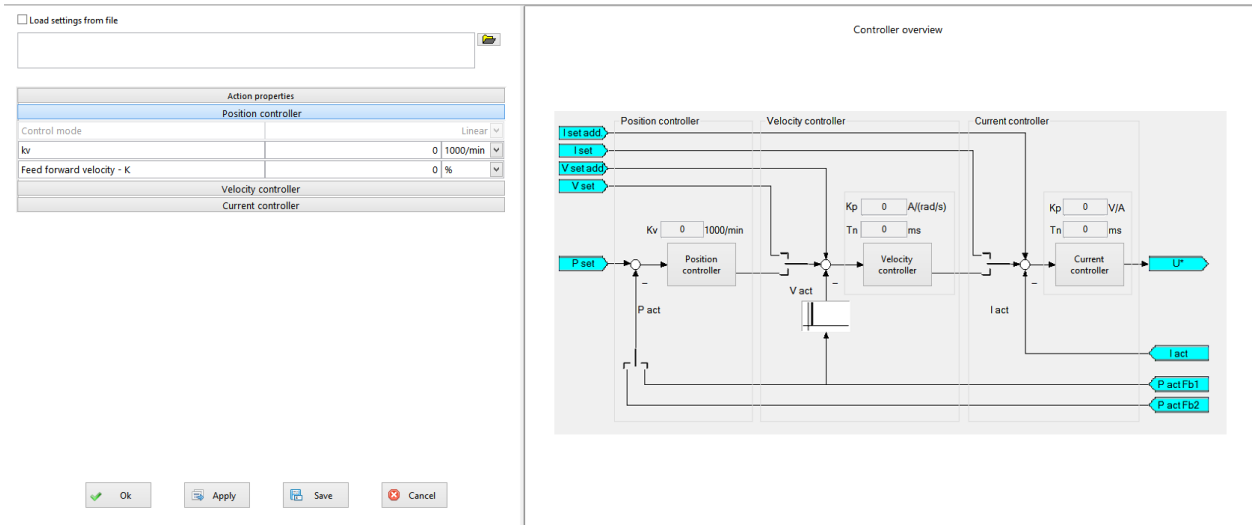
Save offset value in the results file: it is possible to command the software to log the applied position offset (difference between actuator absolute coordinate system and user-specified coordinate system) into the TDMS results file.

ADVANCED SETTINGS

Waiting time before position reset: is a waiting time that is observed at the beginning of the action execution, before the controller performs the position reset; it serves to prevent the effect of any filters configured for the position channel from affecting the value set during reset;

Waiting time after position reset: is a waiting time that is observed after the position reset has been carried out, before moving on to the next action; it serves to prevent the effect of any filters configured for the position channel from affecting the values acquired during subsequent actions in the sequence.

4.4.11. MachineConfig action (version 1)



A *MachineConfig* action is an action through which the user can configure the PID controller of the machine and its layout depends on the hardware installed on the system.

> ***Caution:*** *the PID configuration must be restricted to highly qualified users since editing these parameters improperly may be cause of damages in the machine components.*

The specific parameters that can be configured for a *MachineConfig* action are:

POSITION CONTROLLER

Control mode: this is the type of control algorithm and can't be modified by the user;

kp: proportional gain of the position controller;

Feed forward velocity – K: velocity feed forward, varying from 0 to 100;

Feed forward acceleration – K: (available only for cubic control mode) acceleration feed forward, varying from 0 to 100;

Feed forward acceleration – T: (available only for cubic control mode) acceleration feed forward filter time constant.

VELOCITY CONTROLLER

Acceleration: motor acceleration limit;

Deceleration: motor deceleration limit;

Maximum speed: maximum speed of the motor;

Emergency deceleration: motor deceleration in case of error;

Emergency jerk: motor jerk in case of error (if 0 is written here then the controller uses an infinite jerk value);

Halt deceleration: ;

Halt jerk: ;

T1: velocity command filter time constant;

kp: proportional gain of the velocity loop controller;

Tn: integral component time for the velocity loop controller.

CURRENT CONTROLLER

Imax +: maximum positive peak current percentage value;

Imax –: maximum negative peak current percentage value (it has to be written with sign);

Imax: maximum absolute peak current percentage value (it operates both on positive and on negative values);

kp: proportional gain of the current loop controller;

Tn: integral component time for the current loop controller.

FORCE CONTROLLER

Proportional gain: proportional gain of the force controller;

Integral time constant: integral time constant of the force controller;

Derivative time constant: derivative time constant of the force controller;

Spring effect compensation:

Equivalent rigidity management: the PID control algorithms can take into account the rigidity of the specimen under test: if this option is set to “single value”, the rigidity is expressed by a unique number; otherwise, if the “table” option is selected, the rigidity table displayed on the right of the interface is considered;

Equivalent rigidity: if the equivalent rigidity management is set to “single value”, this is the rigidity of the specimen to be tested expressed in the [load / position] standard unit;

Nr. of samples for derivative/feedforward/input/output filter moving avg.: moving average windows of the various filters available for the PID control.

EXTERNAL CONTROLLER

Proportional gain: proportional gain of the external controller;

Integral time constant: integral time constant of the external controller;

Derivative time constant: derivative time constant of the external controller;

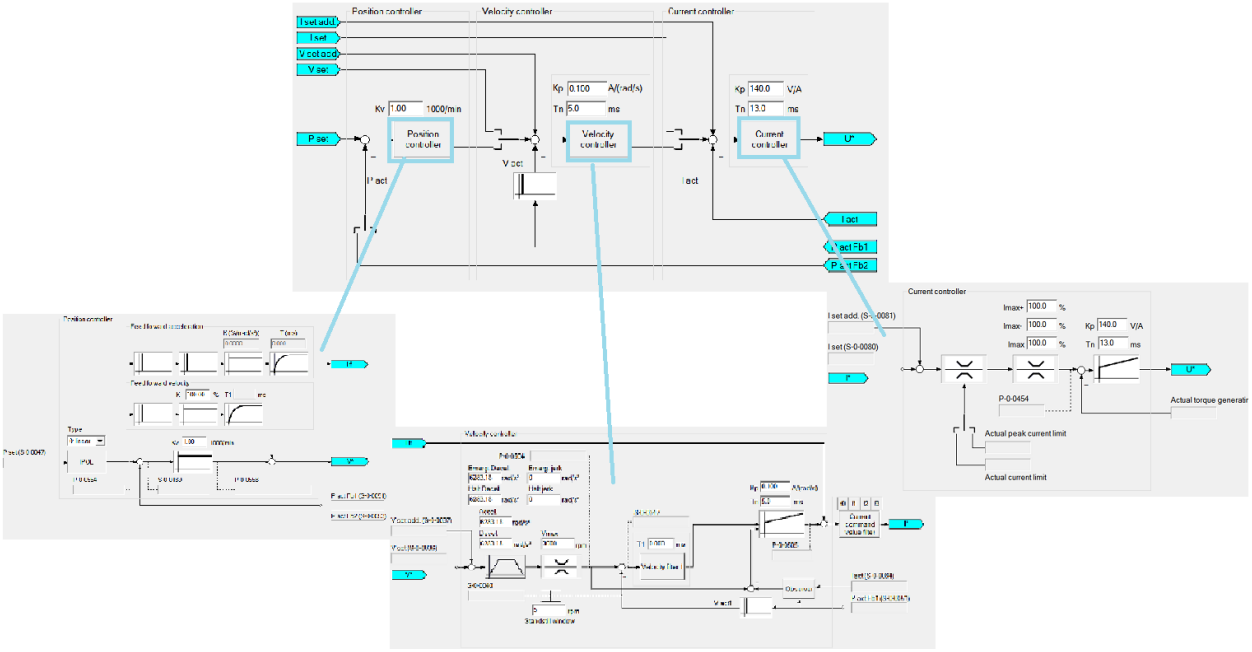
Spring effect compensation:

Equivalent rigidity management: the PID control algorithms can take into account the rigidity of the specimen under test: if this option is set to “single value”, the rigidity is expressed by a unique number; otherwise, if the “table” option is selected, the rigidity table displayed on the right of the interface is considered;

Equivalent rigidity: if the equivalent rigidity management is set to “single value”, this is the rigidity of the specimen to be tested expressed in the [feedback channel unit / position] standard unit;

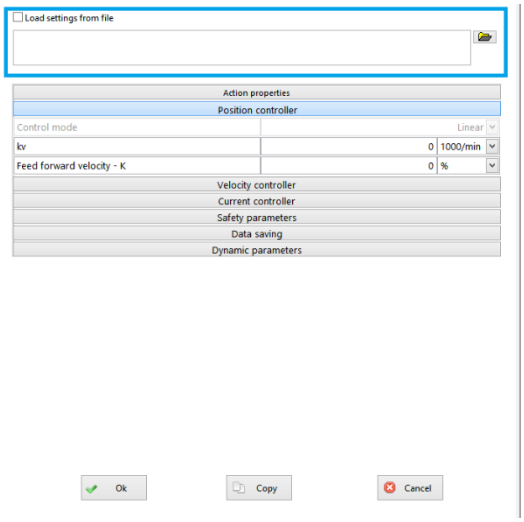
Nr. of samples for derivative/feedforward/input/output filter moving avg.: moving average windows of the various filters available for the PID control.

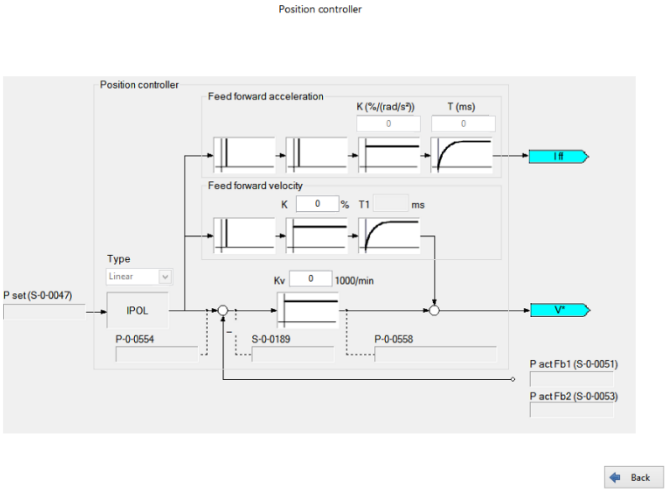
On the right side of the window there is a section dedicated to display a graphical scheme of the controller, with the possibility to open the sections dedicated to the three controller sections and read or write each of the parameter as it can be done on the configuration menu:



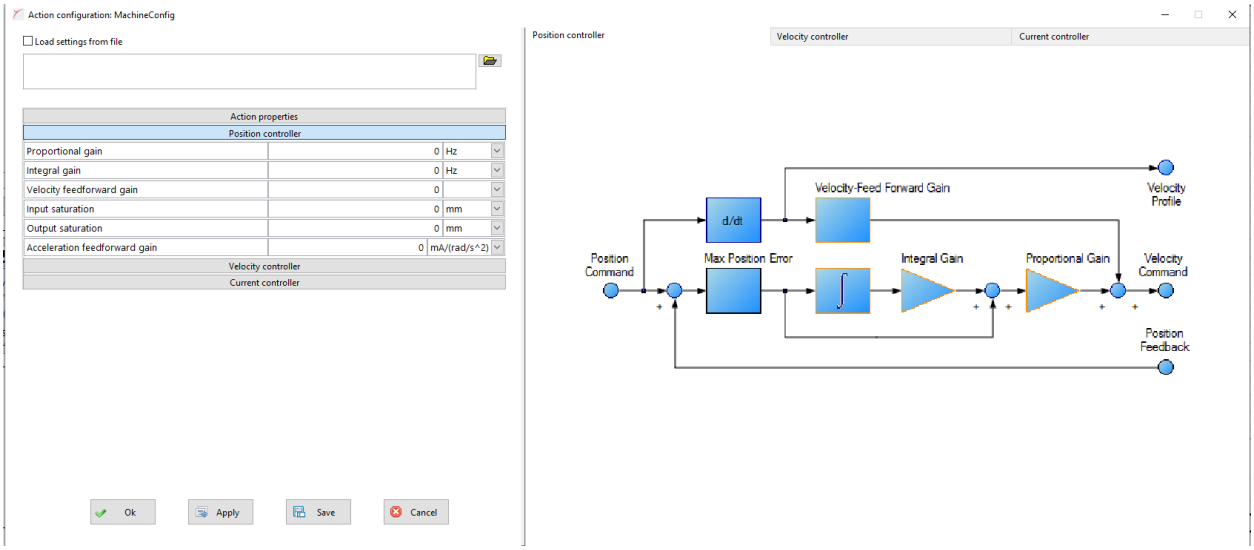
Finally, on the top-left corner of the interface displays a checkbox and a file path: if the checkbox is activated, the configuration parameters are not taken from the user interface, but read from an existing file that becomes the reference for the configuration: this means that if the template configuration file is changed, each action pointing to that file will have its parameters updated; the file is not read when the action is loaded or when its path is specified, but when the test that includes the action is started; in this way the software can be sure that the parameters are updated when needed.

If a *MachineConfig* action is saved as a template action through the dedicated button on the right of the test interface, it becomes a template file that can be loaded for the *load settings from file* options.


Load settings from file
Action properties
Position controller
Control mode
Linear
kv
0
1000/min
Feed forward velocity - K
0
%
Velocity controller
Current controller
Safety parameters
Data saving
Dynamic parameters
Ok
Copy
Cancel


Position controller
Position controller
Feed forward acceleration
K (%/(rad/s²))
T (ms)
0
0
Feed forward velocity
K
0
%
T1
ms
Type
Linear
Kv
0
1000/min
P set (S-0-0047)
IPOL
P-0-0554
S-0-0189
P-0-0558
V*
P act Fb1 (S-0-0051)
P act Fb2 (S-0-0053)
Back

4.4.12. MachineConfig action (version 2)



A *MachineConfig* action is an action through which the user can configure the PID controller of the machine and its layout depends on the hardware installed on the system.

> ***Caution:*** *the PID configuration must be restricted to highly qualified users since editing these parameters improperly may be cause of damages in the machine components.*

The specific parameters that can be configured for a *MachineConfig* action are:

POSITION CONTROLLER

Proportional gain: proportional gain of the position controller;

Integral gain: integral gain of the position controller, varying from 0 to 248;

Velocity feedforward gain: velocity feed forward gain, varying from 0 to 2;

Input saturation: limits the input of the position loop integrator by setting the input saturation;

Output saturation: limits the output of the position loop integrator by setting the output saturation;

Acceleration feedforward gain.

VELOCITY CONTROLLER

Proportional gain: proportional gain of the velocity controller;

Integral gain: integral gain of the velocity controller varying from 0 to 1000;

Observer type: the observer passes the velocity feedback through a PID control loop that behaves like a low-pass filter with the specified bandwidth; if the *No observer* option is selected, the observer is not part of the control loop;

Observer gain

Observer bandwidth: see the description of *Observer type* for this;

CURRENT CONTROLLER

Imax +: maximum positive peak current percentage value;

Imax –: maximum negative peak current percentage value (it has to be written with sign);

Proportional gain: proportional gain of the current loop controller;

FORCE CONTROLLER

Proportional gain: proportional gain of the force controller;

Integral time constant: integral time constant of the force controller;

Derivative time constant: derivative time constant of the force controller;

Spring effect compensation:

Equivalent rigidity management: the PID control algorithms can take into account the rigidity of the specimen under test: if this option is set to “single value”, the rigidity is expressed by a unique number; otherwise, if the “table” option is selected, the rigidity table displayed on the right of the interface is considered;

Equivalent rigidity: if the equivalent rigidity management is set to “single value”, this is the rigidity of the specimen to be tested expressed in the [load / position] standard unit;

Nr. of samples for derivative/feedforward/input/output filter moving avg.: moving average windows of the various filters available for the PID control.

EXTERNAL CONTROLLER

Proportional gain: proportional gain of the external controller;

Integral time constant: integral time constant of the external controller;

Derivative time constant: derivative time constant of the external controller;

Spring effect compensation:

Equivalent rigidity management: the PID control algorithms can take into account the rigidity of the specimen under test: if this option is set to “single value”, the rigidity is expressed by a unique number; otherwise, if the “table” option is selected, the rigidity table displayed on the right of the interface is considered;

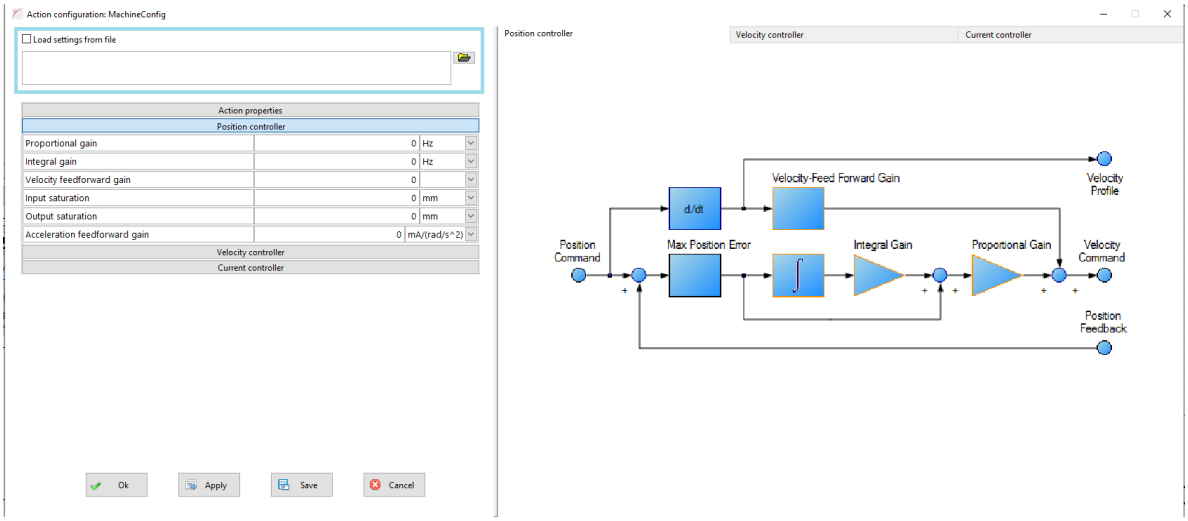
Equivalent rigidity: if the equivalent rigidity management is set to “single value”, this is the rigidity of the specimen to be tested expressed in the [feedback channel unit / position] standard unit;

Nr. of samples for derivative/feedforward/input/output filter moving avg.: moving average windows of the various filters available for the PID control.

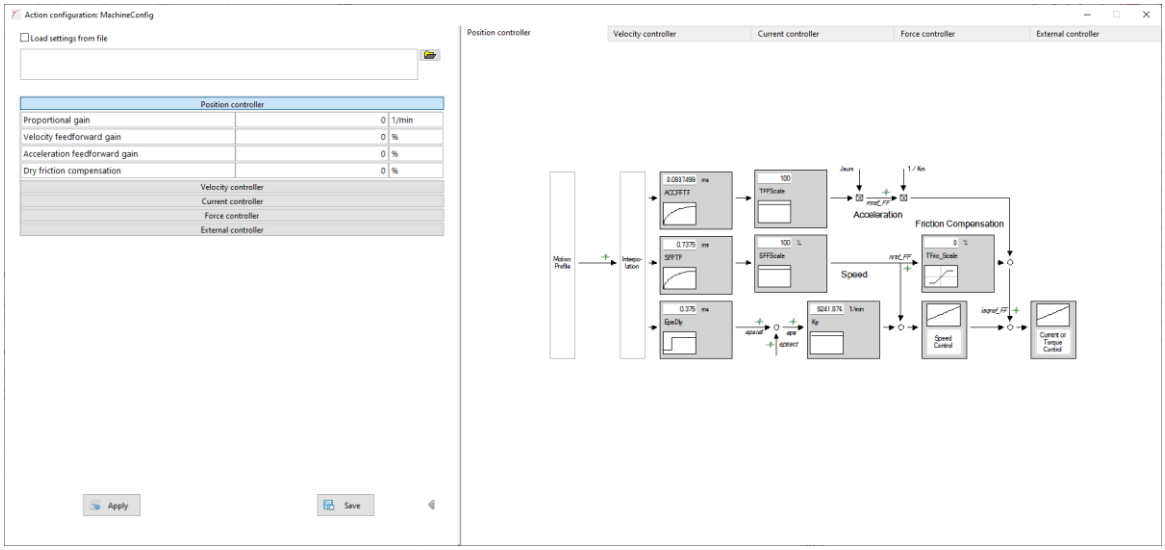
On the right side of the window there is a section dedicated to display a graphical scheme of the controller, which displayed page is automatically adapted to the section of the menu that the user selects.

Finally, on the top-left corner of the interface displays a checkbox and a file path: if the checkbox is activated, the configuration parameters are not taken from the user interface, but read from an existing file that becomes the reference for the configuration: this means that if the template configuration file is changed, each action pointing to that file will have its parameters updated; the file is not read when the action is loaded or when its path is specified, but when the test that includes the action is started; in this way the software can be sure that the parameters are updated when needed.

If a *MachineConfig* action is saved as a template action through the dedicated button on the right of the test interface, it becomes a template file that can be loaded for the *load settings from file* options.



4.4.13. MachineConfig action (version 3)



A *MachineConfig* action is an action through which the user can configure the PID controller of the machine and its layout depends on the hardware installed on the system.

> ***Caution:*** *the PID configuration must be restricted to highly qualified users since editing these parameters improperly may be cause of damages in the machine components.*

The specific parameters that can be configured for a *MachineConfig* action are:

POSITION CONTROLLER

Proportional gain: proportional gain of the position controller;

Velocity feedforward gain: velocity feed forward gain, varying from 0 to 2;

Acceleration feedforward gain:

Dry friction compensation:

VELOCITY CONTROLLER

Proportional gain: proportional gain of the velocity controller;

Tn:

Observer filter time:

CURRENT CONTROLLER

Torque positive peak: maximum positive peak torque percentage value;

Torque negative peak (absolute value): maximum negative peak torque percentage value (it has to be written as an absolute value);

Proportional gain: proportional gain of the current loop controller;

ki: integral time of the current controller;

Drive peak current: read-only field which displays the maximum current managed by the drive.

FORCE CONTROLLER

Proportional gain: proportional gain of the force controller;

Integral time constant: integral time constant of the force controller;

Derivative time constant: derivative time constant of the force controller;

Spring effect compensation:

Equivalent rigidity management: the PID control algorithms can take into account the rigidity of the specimen under test: if this option is set to “single value”, the rigidity is expressed by a unique number; otherwise, if the “table” option is selected, the rigidity table displayed on the right of the interface is considered;

Equivalent rigidity: if the equivalent rigidity management is set to “single value”, this is the rigidity of the specimen to be tested expressed in the [load / position] standard unit;

Nr. of samples for derivative/feedforward/input/output filter moving avg.: moving average windows of the various filters available for the PID control.

EXTERNAL CONTROLLER

Proportional gain: proportional gain of the external controller;

Integral time constant: integral time constant of the external controller;

Derivative time constant: derivative time constant of the external controller;

Spring effect compensation:

Equivalent rigidity management: the PID control algorithms can take into account the rigidity of the specimen under test: if this option is set to “single value”, the rigidity is expressed by a unique number; otherwise, if the “table” option is selected, the rigidity table displayed on the right of the interface is considered;

Equivalent rigidity: if the equivalent rigidity management is set to “single value”, this is the rigidity of the specimen to be tested expressed in the [feedback channel unit / position] standard unit;

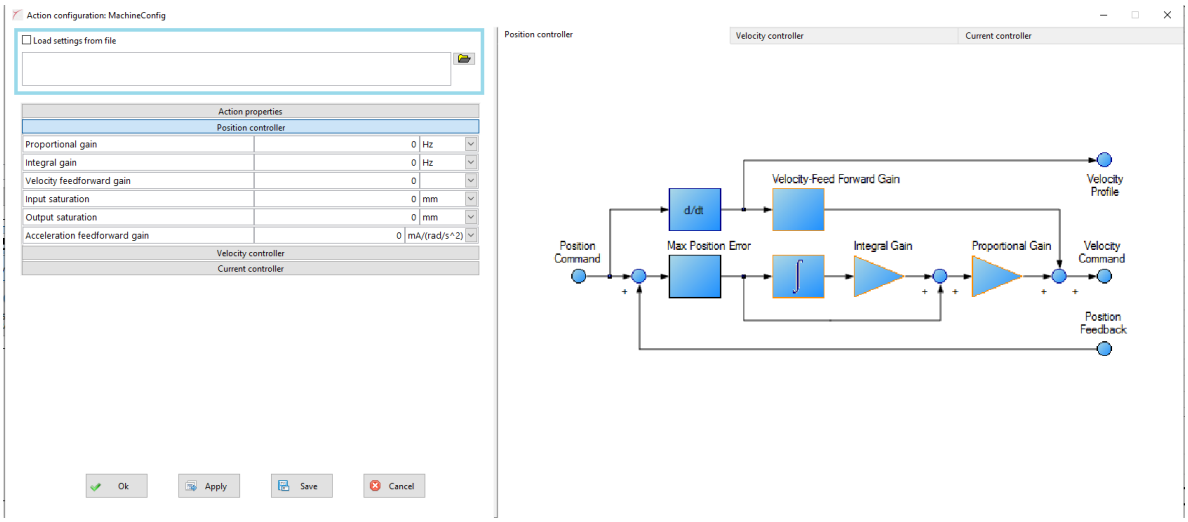
Nr. of samples for derivative/feedforward/input/output filter moving avg.: moving average windows of the various filters available for the PID control.

On the right side of the window there is a section dedicated to display a graphical scheme of the controller, which displayed page is automatically adapted to the section of the menu that the user selects.

Finally, on the top-left corner of the interface displays a checkbox and a file path: if the checkbox is activated, the configuration parameters are not taken from the user interface, but read from an existing file that becomes the reference for the configuration: this means that if the template configuration file is

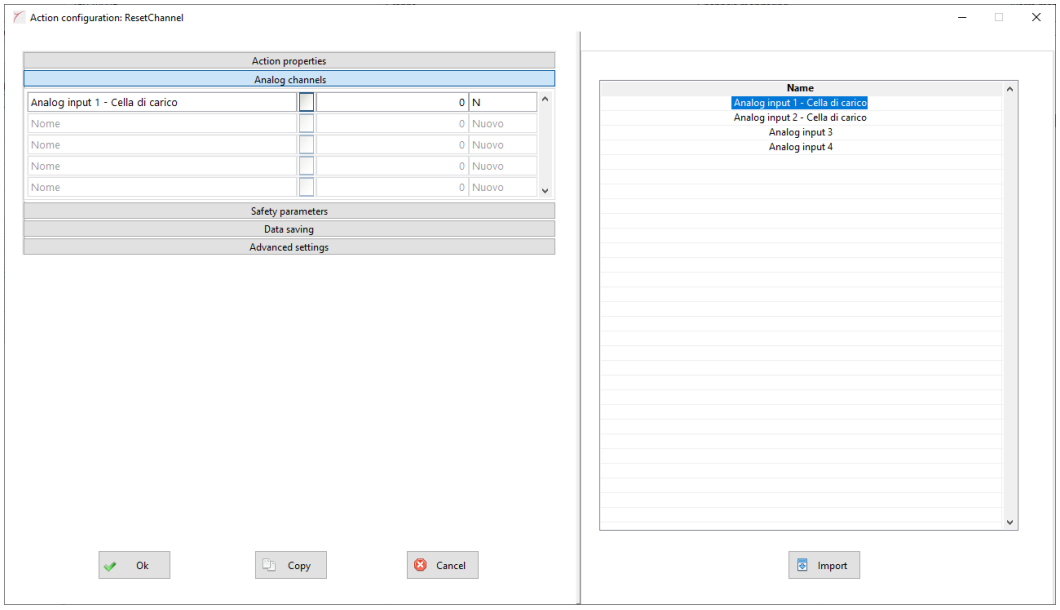
changed, each action pointing to that file will have its parameters updated; the file is not read when the action is loaded or when its path is specified, but when the test that includes the action is started; in this way the software can be sure that the parameters are updated when needed.

If a *MachineConfig* action is saved as a template action through the dedicated button on the right of the test interface, it becomes a template file that can be loaded for the *load settings from file* options.



4.4.14. ResetChannel action

The *ResetChannel* action allows to execute a reset of one or more analog channels at the same time during the execution of the test.



The specific parameters that can be configured for a *ResetChannel* action are:

ANALOG CHANNELS

This is a list of the channels to be reset during the execution of the action: channels can be selected from the table on the right of the interface and imported into the list through the *Import* button; in order to delete an element from the list, it is necessary to right-click on it and select *Delete element*.

For each element of the list there are a checkbox and a numeric value: if the checkbox is off, the controller resets the channel to 0 when the action executes; if the checkbox is on, the controller writes as new channel signal the value specified in the numeric control.

The unit of measurement is automatically taken from the respective analog channel settings.

ADVANCED SETTINGS

Time to wait before the reset: it is possible to command the action to wait for a certain time before resetting the channel(s), in order to stabilize the signals acquired from the field.

4.4.15. SetVariable action

The *SetVariable* action allows to change – during the test sequence – the value of up to 5 of the *global variables* available for the test.

The specific parameters that can be configured for a *SetVariable* action are:

VARIABLE 1 CONFIGURATION

Variable to configure: it is necessary to choose which of the available variables has to be updated; the choice can be made through the drop-down menu or by selecting an element of the table on the right side of the interface and pressing the  button;

Operation type: the update can be mode in several ways: a mathematical operation, the setting of a system parameter (for instance, the current load value is written into the variable), the setting of a parameter from the test sequence (for instance, the target speed of a certain *displacement* action is written into the variable);

Constant value to write from: if the *operation type* is set to *write variable from a constant value*, this is the constant value to write into the variable as soon as the action is executed;

Operation (ordered), Constant value, Variable 1, Variable 2: for operations that need multiple parameters (sum, subtraction, string concatenation…), it is possible to configure whether they are variables or constants and their order.

Example: if the chosen operation is a subtraction, then:

- *Constant,Variable* means that the output value will be *Constant – Variable 1*
- *Variable,Constant* means that the output value will be *Variable 1 – Constant*
- *Variable,Variable* means that the output value will be *Variable 1 – Variable 2*

The variables configuration (drop-down menu and button to paste from the table) is the same as in the *Variable to configure*.

Action of the sequence, Parameter subgroup, Parameter: in case the *operation type* is *write from a parameter of the test sequence*, these settings allow to locate the specific action and its parameter that has to be written into the global variable;

Immediate value update from field: global variables are generally read automatically from the controller with a period of some seconds; it is possible to command the software to read the updated value immediately, so that the TestCenter HMI is aligned with the change faster.

VARIABLE 2-3-4-5 CONFIGURATION

The same as *Variable 1*, but providing also the possibility to activate or deactivate the setting of the parameter.

4.4.16. ChangeParameters action

The *ChangeParameters* action allows to change – during the test sequence – the value of a test parameter (up to 5 test parameters can be customized at the same time for each action).

The specific parameters that can be configured for a *ChangeParameters* action are:

PARAMETER 1-2-3-4-5

Parameter writing activation: this setting has to be turned to ON in order to use all the other options; otherwise, the writing of the parameter is deactivated;

Parameter origin: the parameter that has to be changed can be part of the multiaction test sequence or part of the measurements configuration in case the *Multiaction with measurements* test is in use;

Action of the sequence/Parameter subgroup (or Measurement element), Parameter: the combination of these settings identifies the parameter to be updated in real time;

Operation type: it is possible to set a constant value, execute a mathematical operation on the current parameter value (for instance, increment the current value by 1 each time the action is executed) or write the value of a global variable into the test parameter;

Math operation to execute, Operation constant value member, Global variable to write from: depending on the choice for *operation type*, these settings complete the configuration of the operation to execute to get the output parameter.

4.4.17. SignalGeneration action

A *SignalGeneration* action is an action through which an external device, if present, can be commanded to generate a signal (for example a current signal).

The specific parameters that can be configured for a *SignalGeneration* action are:

MAIN SETTINGS

Analog output to command: the user has to specify which analog output has to be commanded by the action among those that are available;

Profile shape: the user can choose the type of profile to command; in case *stop* is selected, the action stops any running signal generation; depending on the chosen profile shape, different parameters need to be configured; on the right side of the interface a graphs previews the profile composed of the inserted settings;

Maximum setpoint value: the maximum value that has to be reached by the signal in every cycle; the unit of measurement is taken from the selected channel automatically, once the *analog output to command* is specified;

Minimum setpoint value: the minimum value that has to be reached by the signal in every cycle; the unit of measurement is taken from the selected channel automatically, once the *analog output to command* is specified;

Delay: if 0, the signal generation starts as soon as the controller enters this action; if the user sets a greater value, it works as a timeout between the first processing of the action and the actual beginning of the signal generation;

Profile generation duration: if the dedicated checkbox is activated, the numeric value associated sets how long the signal generation has to last; otherwise it continues indefinitely until a *stop* action is encountered;

Transition duration: (only for ramp profiles) this parameter determines how fast the signal has to go from the previous value to the final value (set through the *maximum setpoint value*);

Frequency: (only for wave profiles) this is the frequency of the signal cycles;

Phase: (only for wave profiles) it is possible to set a phase different than 0 for the cycle by editing this value;

Duty cycle: (only for rectangular wave profiles) this is a percentage that sets the ratio between the time spent at the maximum setpoint value and the overall cycle time within a single cycle;

Mean value and RMS: (only for wave profiles) these two read-only fields show the user the mean and RMS value of the configured cycle.

ADVANCED SETTINGS

Signal start together with next action: if activated, this option allows to synchronise the beginning of the signal generation with the beginning of the action that follows the *signalgeneration* in the sequence;

Action enqueueing: this setting is used by the controller if it is already generating an output signal from one of these actions when another one starts executing; if it is set to ON, the next command is kept in memory to be executed as soon as the previous is finished; if it is set to OFF, the controller stays in the action as long as the previous command is concluded before starting the new output.

4.4.18. DigitalOutput action

The *DigitalOutput* action allows to command a digital output, if the machine is equipped with these devices.

The specific parameters that can be configured for a *DigitalOutput* action are:

COMMAND 1..5 FOR DIGITAL OUTPUT

Each of these actions allows to command up to 5 digital outputs at the same time; for each of them, the following options are available:

Command activation: if this setting is ON, the command is active and all the following elements are visible;

Digital output to command: through this drop-down menu it is possible to choose which digital output has to be commanded;

Value to write: apart from the standard “false” and “true” options, it is possible to switch the value of the output (the value commanded is the opposite of the current value) or to write the value of a specific variable into the output;

Variable to use: if the “value to write” is set to “value taken from global variable”, this drop-down menu allows to choose which is the global variable that has to be written into the digital output.

4.4.19. ActuatorCmd action

The *ActuatorCmd* action is built to give some general commands to the actuator.

The specific parameters that can be configured for an *ActuatorCmd* action are:

MAIN SETTINGS

Operation to execute: this is the operation to be commanded to the actuator, that has to be selected among those that are available;

Target load (with sign): this setting is visible only if the operation is set to *static load compensation* (open or closed loop) and sets the static load value that has to be object of the compensation;

Target load reaching threshold: this is a tolerance to consider the target load reached, and it is visible only if the operation is set to *static load compensation* (open or closed loop);

Channels which calibration has to be inverted: this element is only visible for operations to assign the polarity (or sign convention) of the actuator, and contains a list of all available analogue inputs: those whose sign the software must change (automatically inverting all calibration points) when changing polarity must be selected; it is mandatory to invert the sign of at least the channel relative to the load cell which detects the force in the actuator's direction of motion, since the sign convention of this sensor must always agree with that of the actuator itself.

4.4.20. Wait action

A *Wait* action is a timeout that has to elapse before the sequence moves on.

The specific parameters that can be configured for a *Wait* action are:

ACTION END CONDITIONS

Break threshold: this is a percentage, calculated with respect to the current load, that defines a limit difference between two consecutive load values to detect a fracture through a dedicated automatic algorithm;

Timeout: this parameter sets how long the wait action has to last.

4.4.21. GenericPID action

A *GenericPID* action can be used to command a PID regulator of generic type to generate a signal on one of the analog output channels available.

The specific parameters that can be configured for a *GenericPID* action are:

MAIN SETTINGS

Action to command: the *GenericPID* action can activate several types of actions on the controller:

- Command stop: if there is another command running, it is interrupted;
- Gains adjustment: the action is only used to change the PID regulation parameters that are being applied to the current command;
- Constant value command: a constant setpoint value is commanded to the target channel;
- Point-by-point setpoint command: the user can configure a random profile by inserting various points (time;setpoint) in a dedicated table

Analog feedback channel: this is the input channel that represents the feedback channel of the setpoint regulation; analog inputs and mathematical channels are included among the choices;

Analog channel to command: this is the analog output channel on which the controller commands the setpoint regulation;

Final setpoint value: if the operation is set to *constant value command*, this is the value of the setpoint that has to be applied;

Tolerance around setpoint: if the operation is set to *constant value command* or *point-by-point setpoint command*, the controller interprets the setpoint as reached once the feedback signal stays within a certain tolerance around the configured setpoint for a specified time interval: this is the tolerance value;

Time in tolerance to consider setpoint reached: if the operation is set to *constant value command* or *point-by-point setpoint command*, the controller interprets the setpoint as reached once the feedback signal stays within a certain tolerance around the configured setpoint for a specified time interval: this is the time interval.

PID CONFIGURATION

Proportional gain, Integral time and Derivative time: these are the common PID regulation parameters to be used by the controller to command the setpoint value;

Maximum saturation and Minimum saturation: if activated through their checkboxes, these are the maximum and minimum values that shall not be exceeded by the commanded output channel;

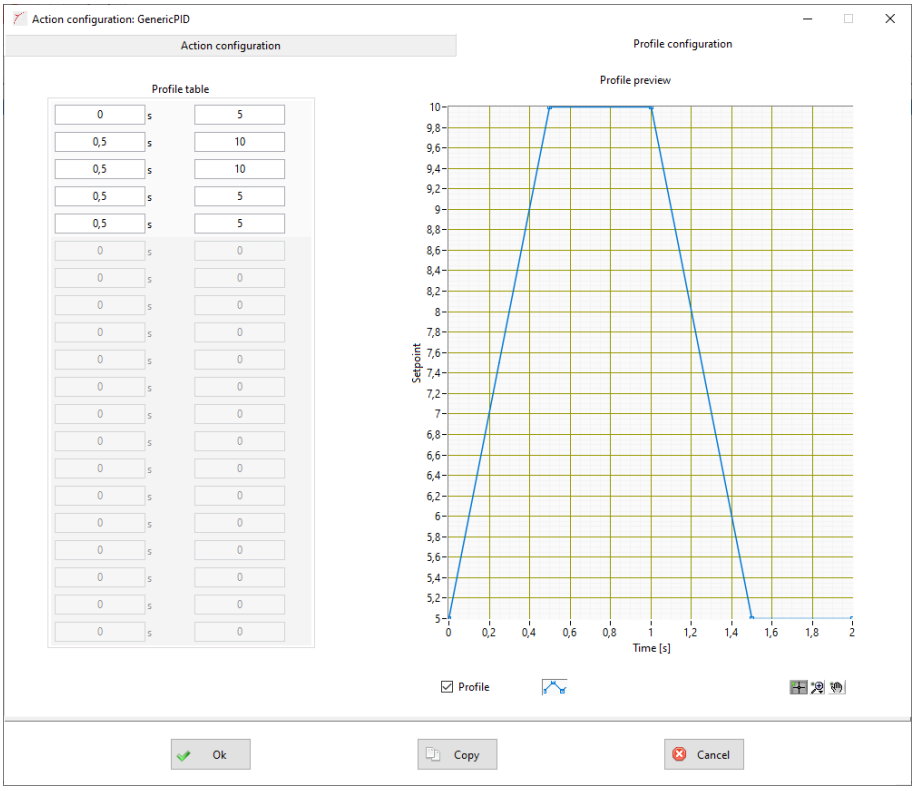
Feedforward gain: feedforward gain of the PID controller.

ACTION END CONDITIONS

Timeout: maximum duration of the action.

Action sequence management: if this control is configured to *stay in current action until command finished*, the sequence of actions is not continued until the setpoint has been reached, or the profile to be tracked is completed; instead, if the *give command and move to next action* option is selected, the controller activates the setpoint command with the configured settings, and then moves to the next action of the sequence while the regulation keeps running in background.

If the operation to be commanded by the action is set to the application of a point-by-point setpoint profile, another tab called *Profile configuration* is displayed on the top of the action interface: it contains a table to configure all the points of the profile to track, and a preview graph:



The time values are interpreted as incremental.

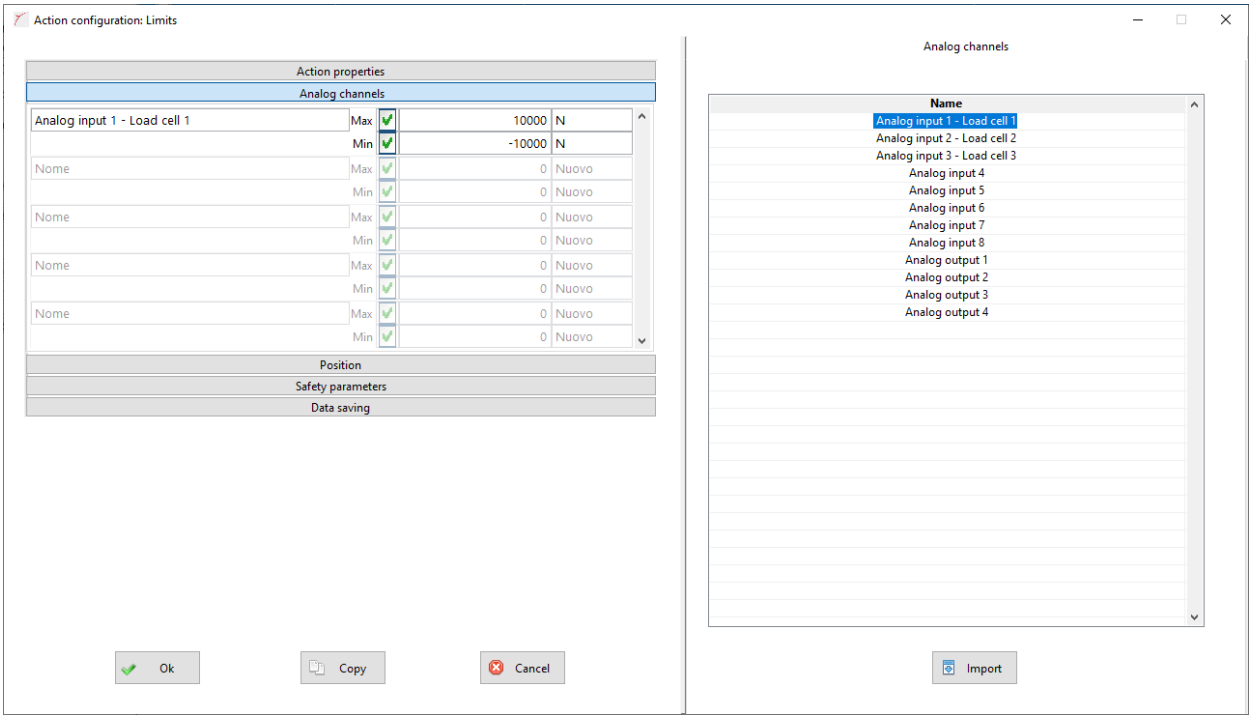
4.4.22. Limits action

The *Limits* action can be used to set dynamically the operation limits of sensors and analog channels while a test is running, in addition to the default load limits that can be set for each action of the sequence.

The specific parameters that can be configured for a *Limits* action are:

ANALOG CHANNELS

This section allows to set the maximum and minimum limits for one or more analogue channels of your choice: on the right-hand side you will see a list of available channels, which can be imported by pressing the Import button located below the list.



Once a channel becomes included in the menu, the action is going to configure its limits as soon as it executes, according to the values the user writes in the field dedicated to the maximum and minimum and to the activation checkboxes state (if a checkbox is activated, its limit is activated).

POSITION

Limits activation in coordinate system in use: if this item is set to ON, position limits in the coordinate system in use are assigned to the actuator when the action is executed, according to what is set in the next control;

Position in active coordinate system: the items in this control are only visible when the previous option is active, and allow to choose which position limits the actuator must adhere to in relation to the coordinate system in use, otherwise the test is terminated; the checkboxes for maximum and minimum

values are used to decide whether to assign a value (checkbox activated) or to disable the position limit check (checkbox deactivated);

Limits activation in absolute coordinates: if this item is set to ON, position limits in its absolute, invariant co-ordinate system are assigned to the actuator when the action is executed, as set by the following control;

Position in the absolute coordinate system: the items of this control are only visible when the previous option is ON, and allow to choose which position limits the actuator must adhere to in relation to its absolute and invariant coordinate system, otherwise the test is terminated; the checkboxes for maximum and minimum values are used to decide whether to assign a value (checkbox activated) or to disable the position limit check (checkbox deactivated).

4.4.23. TDMS action

The *TDMS* action allows to close the current results file and create a new one during the test sequence.

The specific parameters that can be configured for a *TDMS* action are:

MAIN SETTINGS

Folder path where to save the file: this is the folder where the new TDMS file is going to be saved;

File name origin: it is possible to select whether the name to be given to the new file has to come from a text inserted manually or from the value of a global variable;

TDMS file name: the name of the new file created by the software when executing this action: depending on the setting of the previous parameter, it can be a fixed text or a drop-down menu to choose a global variable; in the latter case, the name of the new file is the value of the variable at the moment of the execution of this action;

Add timestamp to file name: if ON, the timestamp of the file creation is automatically added to the new file name to preserve its uniqueness;

Add specimen name to file name: if ON, the specimen name is automatically added to the new file name.

4.4.24. For loop actions

A *for loop* is defined by two different actions, the first one called *for loop start* to begin the loop and the second named *for loop end* to conclude it: the only action provided with a configuration interface is *for loop start*; the software repeats as many times as the user configures the sequence of actions included between the *for loop start* and the *for loop end*.

While the *for loop end* action has no dedicated interface (nevertheless, a *for loop end* has to be places in the sequence for each *for loop start*), except for the configuration of standard properties such as label, description, etc., the specific parameters that can be configured for a *For loop start* action are:

ACTION END CONDITIONS

Number of repetitions: this parameter states how many times the sequence between the selected *for loop start* and its correspondent *for loop end* has to be repeated.

SAVING OF ACTIONS INSIDE LOOP

Periodic saving of actions inside loop: if this setting is configured to OFF, all the actions inside the for loop always keep their saving settings; in other words, their data are either always saved or always not saved every time they are called during the loop execution; instead, if the choice is ON, only a certain percentage of the loop executions is stored on disk (see the description of the following parameter);

% of repetitions to save: through this parameter the user sets how many of the iterations get saved on the results file (for instance, 5% = 1 for each 20); within an iteration that has to be saved, only the data of the actions for which the saving is activated are logged on disk.

ADVANCED SETTINGS

Cycle count management: this option determines the behaviour of the action cycle counter:

- Resume at start & reset at successive iterations: when the action is executed for the first time since the test start, the cycle counter is preserved; if the development of the sequence leads to performing the same action again (for instance, through a for loop or a *GoTo* action), the cycle counter is reset in the successive beginnings of the action;
- Always reset when entering action: the cycle counter is reset every time the action begins;
- Never reset when entering action: the cycle counter is never reset when the action begins.

4.4.25. Temperature action

If the system includes a climatic chamber, the *Temperature* action allows users to control temperature and humidity (if available) during the test sequence.

The specific parameters that can be configured for a *Temperature* action are:

MAIN SETTINGS

Operation type: this field is dedicated to choosing whether to use the action to command a temperature (and/or humidity, if available) setpoint or to turn the chamber off automatically;

Temperature: if the dedicated checkbox is activated and the *operation type* is set to *conditioning*, this is the temperature value that the machine will reach and maintain until another action of the same type gives a different command;

Temperature rate (without sign): this parameter may be present or not depending on the system configuration; if available, it has to be always positive and states how quickly the chamber will aim at the target temperature;

Relative humidity: this parameter may be present or not depending on the system configuration; if the dedicated checkbox is activated and the *operation type* is set to *conditioning*, this is the target relative humidity that the machine will reach and maintain until another action of the same type gives a different command;

Humidity rate (without sign): this parameter may be present or not depending on the system configuration; if available, it has to be always positive and states how quickly the chamber will aim at the target humidity;

Wait for the set parameters to be reached: if this field is set to *ON*, the controller waits for the target temperature and/or humidity, according to what the user configures, to continue the test sequence; if it is set to *OFF*, the command is given and the sequence instantly goes on while the climatic regulation is in progress.

4.4.26. DataAcq action

The *DataAcq* action is specifically built to read a large amount of data from the field in a single, time-limited acquisition at a sampling rate that can be higher than the one of the standard data; a *DataAcq* action must be used twice in order to be effective: the first instance of the action configures the acquisition so that the controller can store the data during the motion of the machine, and finally log them when the action is used for the second time.

The specific parameters that can be configured for a *DataAcq* action are:

MAIN SETTINGS

Operation to execute: through this option the user chooses if the action is used to prepare the acquisition or to sample the data from the field once they have been collected; if the *Data acquisition* option is selected, no other parameter of this section needs to be configured;

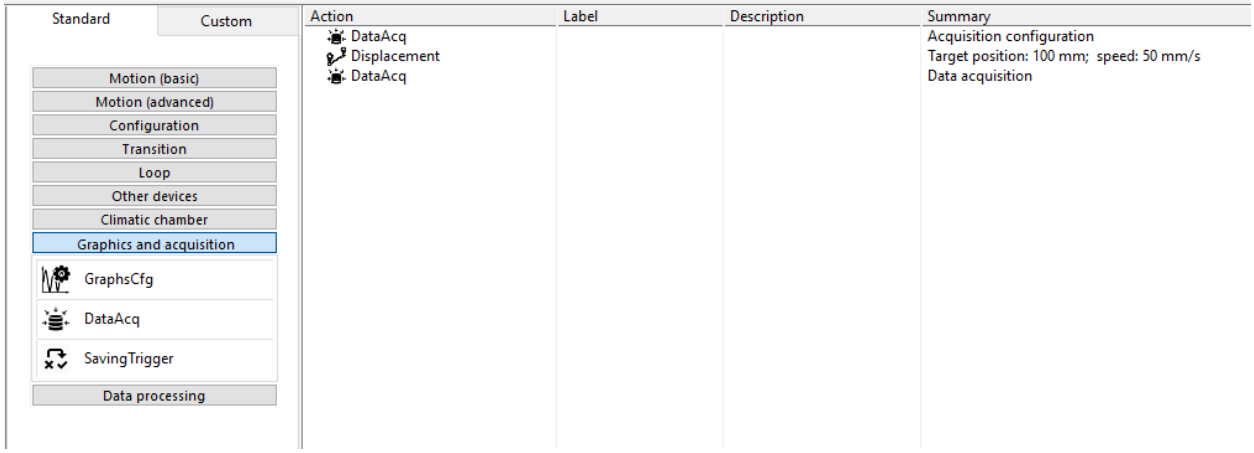
Acquisition trigger channel: this is the sampled channel that commands the trigger for the high speed acquisition to begin;

Acquisition trigger value: if the checkbox is activated, this field states the comparison that activates the trigger: the channel selected in the option just above is continuously monitored and, as soon as the configured condition takes place, the acquisition begins; if the checkbox is left blank, the acquisition begins immediately after the *DataAcq* action used to prepare the data sampling;

Pre-trigger time for acquisition: if the dedicated checkbox is activated, the software dedicates a part of the available memory to store data that happened before the trigger occurred, according to the amount of time chosen by the user;

Total acquisition duration: this is the overall duration of the data acquisition, including the pre-trigger time.

The following picture shows an example of how to use the *DataAcq* action: in this case, the trigger for data saving is supposed to occur during the *Displacement* action.



In the procedure displayed here, the program begins to check for the trigger occurrence as soon as the sequence moves from the *DataAcq – Acquisition configuration* action to the following, which in this case is the *Displacement*.

4.4.27. Report action

This action can be used to generate a test report automatically within the sequence.

The specific parameters that can be configured for a *Report* action are:

MAIN SETTINGS

Path of the .setup file with the configuration to load: in order to generate a report, a post-processing configuration file (*.setup*) has to be loaded: this path control allows the user to configure which one; for further details about the usage of *.setup* files, please check the software manual of the TDMS viewer environment;

Select file path at report generation: if this setting is OFF, the output report path has to be configured within the interface of the action through the controls named *folder path where to save the test report* and *report file name (without extension)*; if it is ON, the two elements mentioned before are hidden and the user is asked to choose for the file path manually (through a *file explorer* window) at the moment of the report generation;

Save report in the same folder as TDMS file: if this option is active, the *Folder path where to save the test report* control is automatically hidden and the report is saved into the same folder where the TDMS file generated by the test is located;

Folder path where to save the test report: the user can select the destination of the report that will be generated;

Report file name origin: users can configure a fixed name for the report, or choose a variable that will define the name of the report according to its value when the action is executed, or finally select the option *Same name as TDMS file* to give the report the same name as the results file generated by the test;

Report file name (without extension): this is the name that will be given to the resulting document; the software will add the extension by itself;

Variable to use to define report file name: the drop-down menu allows to select the variable whose value when the action is executed will define the name of the report file;

Generate PDF report, Open PDF report after generation, Open Excel report after generation, Open report destination folder: through these additional fields, the user is able to customize the behaviour of the action according to his/her specific desires.

ADVANCED SETTINGS

TDMS file to use for the report: this option allows the user to choose whether to execute the report from the current TDMS results file or to load an eventual previous file in case the sequence includes a *TDMS* action.

4.4.28. Measurement start/stop actions

These actions are significant only if the active test protocol is *Multiaction with measurements* because they command the start and the stop of the measurements acquisition.

Both of them have in the *main settings* the configuration of the measurement graph to start or stop (only one graph for each action): the sampling of the data for the measurements goes from the start moment to the stop moment, all the other data are discarded by the controller.

The *measurement start* action provides also these other parameters:

MAIN SETTINGS

Trigger type: the user can choose a trigger based on time or on a sampled channel in order to make the measurement start; if no trigger is configured, the measurement is started as soon as the action is executed;

Trigger activation delay: this control is visible only if the *trigger based on time* is configured, and states the time that has to elapse from the execution to the *measurement start* action to the actual start of the measurement;

Channel to monitor for trigger activation and Channel value that activates the trigger: if the *trigger based on acquisition channel* is chosen, these two controls allow to set the channel that has to be monitored and the condition that activates the trigger;

HMI synchronization timeout: this is a timeout value used to make sure that the TestCenter software is up-to-date with the action sequence state, so that old data are not cleared before making sure they have been completely collected.

4.4.29. ExportResults action

This action can be used to export test results through software utilities outside the TestCenter.

The specific parameters that can be configured for an *ExportResults* action are:

MAIN SETTINGS

Folder path where to save the file: the directory where the output file has to be saved;

Export file name (with extension): the name to be given to the output file, including its extension;

Add timestamp to file name: if this option is activated, the software adds the timestamp to the output file name automatically, in order to avoid overwriting eventual other files with the same name in the same directory;

Exporting tool to run: this is the path of the external software utility to be launched by the main software when this action is executed; the software utility includes the source code to export the data in the desired way, and has to be a .lvlibp format file.

ADVANCED SETTINGS

TDMS file from which to export data: if an action is used throughout the sequence to change the active results file at a certain point, this option allows the user to choose whether to run the exporting utility on the current file or on the previous file, which stays in memory until another file change is executed.

4.4.30. Measurement save action

The *Measurement save* action is generally used to save measurements data into the TDMS results file or to export them in other formats when running *multiaction with measurements* tests, but it can be also be helpful to save global variables value as part of standard multiaction tests.

The specific parameters that can be configured for a *Measurement save* action are:

RESULTS SAVING CONFIGURATION

This is the section that commands the saving of the measurement results into the current TDMS file of the test.

Measurement results saving activation: this is used to activate or deactivate this section; all the following parameters are hidden if this control is set to OFF;

Saving group name origin: it is possible to choose whether to provide the group name manually or to use a global variable to set it;

Saving group name: this is the name of the group where the data are going to be saved, inside the TDMS results file of the test (the same file that includes the data of the action); it can be a constant text field if *saving group name origin* is set to *constant*, and a drop-down menu with the list of the global variables if the *saving group name origin* is set to *variable*;

Specimen name origin and Specimen name: the same concept as *saving group name origin* and *saving group name*, but in this case applied to a column that is added by default to the data, in order to log the specimen name.

RESULTS EXPORT

This is the section that commands the export of the measurement results into an external CSV file.

Results export on text file: this is used to activate or deactivate this section; all the following parameters are hidden if this control is set to OFF;

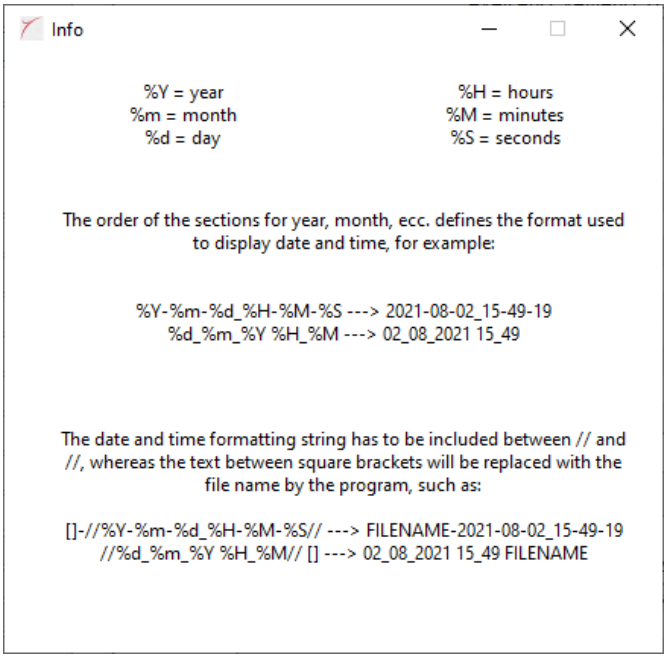
Folder path where to save the file: this is the path of the folder where the csv file is going to be stored;

Same file name as the TDMS of the test: if this is ON, the external csv is automatically given the same name as the active TDMS results file; otherwise, the user is asked to insert the file name manually;

Export file name: this is the name that has to be assigned to the csv file, in case the option *same file name as the TDMS of the test* is OFF;

Add date and time to file name automatically: in order to make sure that the file is unique, it is possible to add a timestamp to its name automatically;

File name format: the user has some degrees of freedom in the configuration of the file name with respect to the possibility of having the timestamp automatically added; please refer to the software help window about this:



If the syntax is incorrect, the text gets coloured in red;

File name preview: a read-only section that provides the preview of the resulting file name with the current settings;

File extension without period: the extension of the file, without any period (for instance: csv, not .csv);

Column separator: the character to be used as column separator in the export file;

Number of digits of precision to export: it is possible to customize how many digits of precision to keep for the numbers that will be exported into the file.

GRAPH SAVING CONFIGURATION (EXPORT OF FILE WITH DATA)

This is the section that commands the export of the entire data sets of the measurement graphs into an external TDMS file.

Data graph saving activation: this is used to activate or deactivate this section; all the following parameters are hidden if this control is set to OFF;

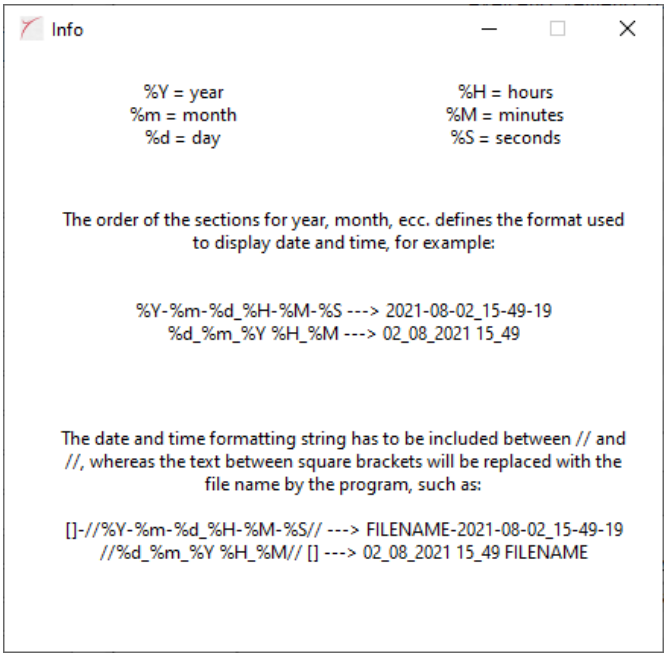
Folder path where to save the file: this is the path of the folder where the tdms file is going to be stored;

Saving file name origin: it is possible to choose whether to provide the file name manually or to use a global variable to set it;

Saving file name: this is the name of the file where the data are going to be saved; it can be a constant text field if *saving group name origin* is set to *constant*, and a drop-down menu with the list of the global variables if the *saving group name origin* is set to *variable*;

Add date and time to file name automatically: in order to make sure that the file is unique, it is possible to add a timestamp to its name automatically;

File name format: the user has some degrees of freedom in the configuration of the file name with respect to the possibility of having the timestamp automatically added; please refer to the software help window about this:



If the syntax is incorrect, the text gets coloured in red;

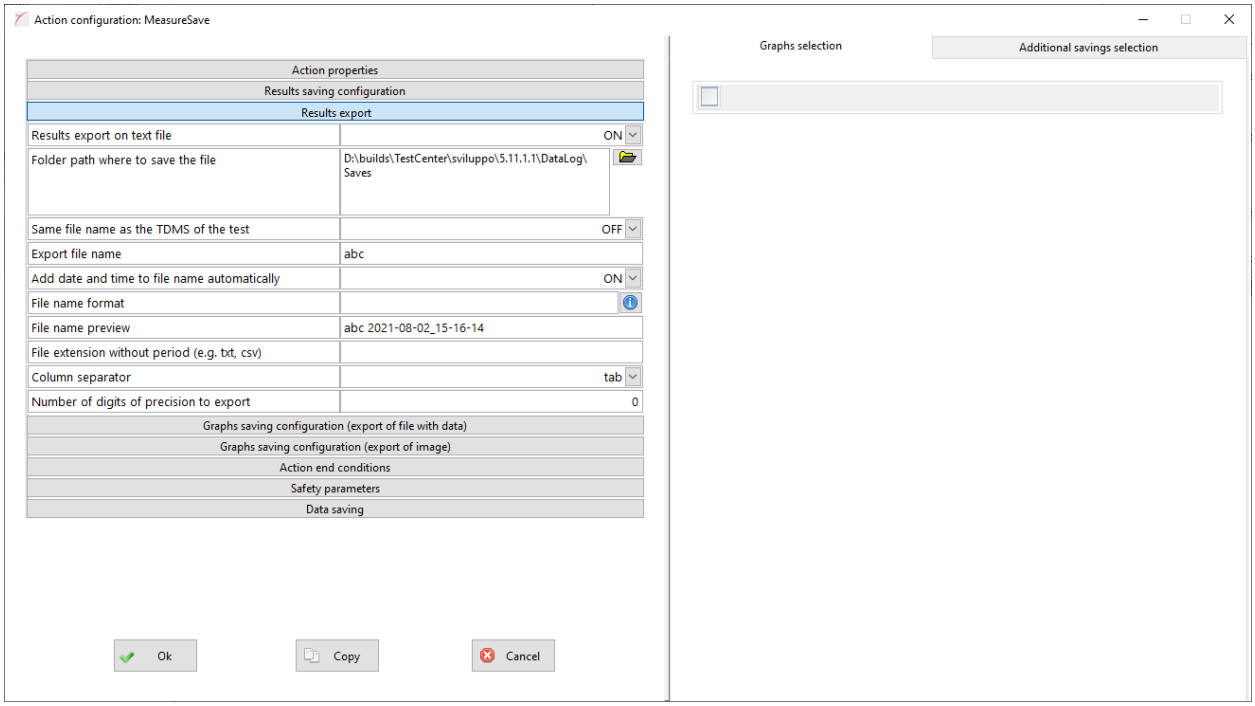
File name preview: a read-only section that provides the preview of the resulting file name with the current settings.

GRAPH SAVING CONFIGURATION (EXPORT OF IMAGE)

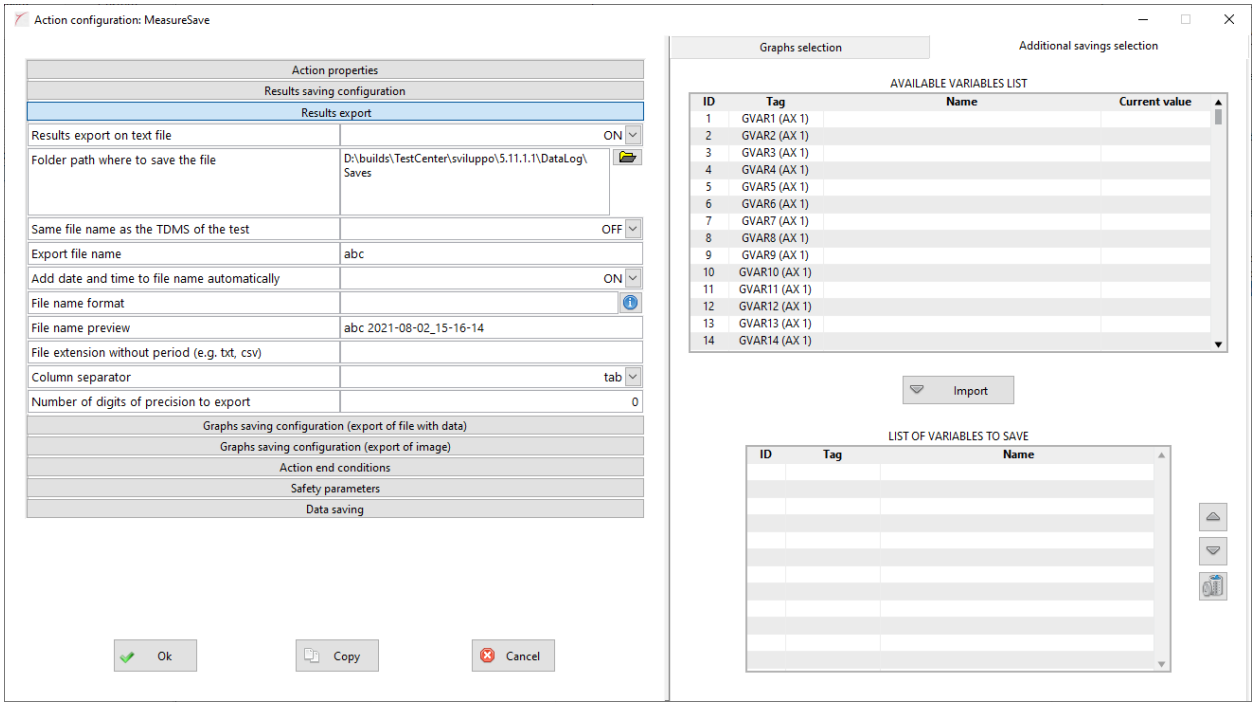
This is the section that commands the export of the measurement graphs as screenshots. In case this option is active, the user has to remember that only the first graph checked in the list is exported as image (in order to export more graph, other actions of these type are needed in the sequence).

The parameters to be configured for this functionality are the same as in the *GRAPH SAVING CONFIGURATION (EXPORT FILE WITH DATA)*.

On the right side of the interface there is a tab control with two pages:



The first tab is dedicated to the configuration of the measurement which data are going to be exported in the various shapes allowed by the action.



The second page provides the possibility to choose some global variables to be saved together with the measurement results, referring to the *RESULTS SAVING CONFIGURATION* and *RESULTS EXPORT* functionalities: the variables have to be selected among the list above (*AVAILABLE VARIABLES LIST* –

multiple selection is allowed by holding the CTRL key) and imported in the *LIST OF VARIABLES TO SAVE* through the “Import” button. The three buttons on the right of the *list of variables to save* permit to

- Move a variable upwards in the order of the *list of variables to save* (arrow up button)
- Move a variable downwards in the order of the *list of variables to save* (arrow down button)
- Delete a variable from the *list of variables to save* (bin button)

4.4.31. If – Else – End if actions

These three actions provide the possibility to vary the test sequence introducing logical conditions through which the controller is able to choose whether to do or not to do something.

For each *If* action, an *End if* has to be placed in the test sequence, whereas the *Else* is not compulsory.

While the *End if* action has no interface and the *Else* action only provides the general settings (label, description…), the *If* action configuration interface is composed of the following elements:

MAIN SETTINGS

Variable 1: this is the first variable that builds the logical condition; it can be chosen among the *system parameters* and the *global variables* through the lists on the right of the interface and the  button;

Operation: this is the comparison to make between *Variable 1* and *Variable 2* to generate the output (TRUE or FALSE); =| means "not equal to";

Variable 2 element type: the second parameter of the comparison can be either a constant value or another variable chosen among the system data or the global variables;

Constant value: if *Variable 2 element type* is *constant*, this is the constant value to which *Variable 1* has to be compared;

Variable 2: if *Variable 2 element type* is *variable*, this is the second parameter (to be chosen among the system variables or the global variables in the same way as *Variable 1*), to be compared to *Variable 1*.

4.4.32. GoTo action

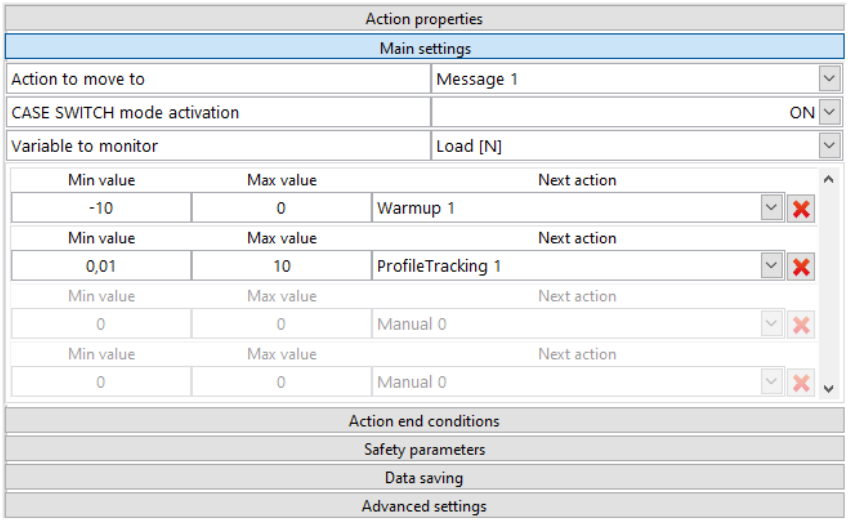
The *GoTo* action allows to force the test sequence into a certain action instead of automatically moving to the next one.

The specific parameters that can be configured for a *GoTo* action are:

MAIN SETTINGS

Action to move to: this is the action that will become the next of the sequence: the *GoTo* action moves to the action selected here; if the *CASE SWITCH* mode is activated, the action configured here is the default case to be applied if none of the other conditions is met;

CASE SWITCH mode activation: the software provides the possibility to configure a table of possible conditions that a certain variable (*Variable to monitor*) may meet, and the respective following action:



In the case exemplified here, the sequence will:

- Move to *Warmup 1* if the current load value at the moment of the action execution is between –10 N and 0;
- Move to *ProfileTracking 1* if the current load value at the moment of the action execution is between 0.01 and 10 N;
- Move to *Message 1* in none of the above occurs.

4.4.33. ExitRoutine action

If an *ExitRoutine* action is placed in a test sequence, it automatically becomes the next action to execute in case of any possible alarm condition that generally forces the test to stop; in this way it is possible to create custom sections of the multiaction sequence that are going to be executed only in case an alarm occurs.

There is to remember that no more than one *ExitRoutine* action has to be placed in a test sequence, and that this action does not operate if the user stops the test manually through the *Stop* button:

4.4.34. Message action

The *Message* action allows to display a popup message at a certain point of the actions sequence.

The specific parameters that can be configured for a *Message* action are:

MAIN SETTINGS

Sequence stop until user input: if this option is set to ON, the test is stopped until the user closes the message popup window; otherwise, the action sequence continues after displaying the popup;

Message: this is the text to display in the popup window;

Save message with results as property: it is possible to log the message in the results file together with the other data;

Monitor on which the message has to be displayed: if the computer has multiple screens, there is the possibility to customize in which one the popup appears.

TEXT FORMATTING

This section is dedicated to the various graphic properties that define the text layout.

BUTTONS FORMAT

Number of buttons to display: it is possible to choose a single-button or multiple-button layout for the popup window;

Button on which key focus has to be applied: unless this setting is configured to *none*, it determines the button that can be activated by simply pressing the "enter" key without the need to click on it with the mouse;

Button test, button width, button height: these options are repeated as many times as the number of buttons chosen to display; the buttons are automatically distributed on the interface of the popup window by the software.

ADVANCED SETTINGS

Message window behaviour: from here it is possible to define the behaviour of the window that will appear to show the message, choosing among

- Default: a window with default behaviour allows normal interaction with all the other windows of the programme during the time it remains open, and can move into the background;
- Default with periodic return to foreground: the window retains its default behaviour but, in addition, the programme automatically and periodically brings it to the foreground to give it more prominence;

- Floating: a window with floating behaviour allows the user to interact with the rest of the programme, but always remains in the foreground for as long as it remains open;
- Modal: a modal window always remains in the foreground and does not allow the user to interact with other windows of the programme as long as it is open.

Periodicity of window return to foreground: this option is only visible if the behaviour set is *Default with periodic return to foreground*, and defines every how many seconds the software must return the window with the message to the foreground for as long as it remains open.

Finally, at the bottom of the *Message* action interface, along with the *Ok*, *Copy* and *Cancel* buttons (common to the other actions) there is also a *Preview* button that allows to simulate the opening of a pop-up window reproducing the message according to the settings chosen for the text, buttons and window formatting: in this way, users can easily visualise the result of the action configuration without having to run a test sequence.

4.4.35. Print action

The *Print* action allows the software to interface with a Dymo LabelWriter 550 printer or another device for which the same communication library is compatible.

In order to have this action functioning, the TestCenter installation folder has to contain also additional software packages for which it is necessary to contact STEP Lab support.

The specific parameters that can be configured for a *Print* action are:

MAIN SETTINGS

Path of the template to use: the printing requires a label model in which text and images can be loaded by the TestCenter software; the model has to be prepared through the dedicated software of the printer; it can include up to 3 text fields and up to 3 image fields;

Number of copies to print: it is possible to print multiple copies of the same label at once.

TEXT FIELD 1..3

Text field writing activation: if this is ON, the text field is in use;

Name associated to the text field: this is the name given to the text field by the software of the printer, and can be found by opening the label template through a text editor;

Number of elements to use to build the text: the text can be composed of up to 5 subsets, which are concatenated by the software; for each subset it is possible to configure:

Origin of the text to use: the text can be a constant string, a global variable (in this case the text is the value of the global variable at the moment when the action is executed), or a combination of a constant string and a global variable in the desired order;

Text: if the constant string is in use as origin of the text or part of it, this is the text to be included in the label;

Variable containing the text to use: if a global variable is in use as origin of the text or part of it, this is the drop-down menu to select which variable has to be written;

Finally, the last set of parameters is about text wrapping:

Enable text wrapping: if this option is enabled, the text composed for the label is wrapped according to the information specified below;

Text area width: the width (in mm, not in pixels) of the area dedicated to the text in the label template used;

Text area height: the height (in mm, not in pixels) of the area dedicated to the text in the label template used;

Custom font name: if nothing is specified here, the software uses a default font;

Font reference size: this is the ideal size of the text, but the software may downsize it automatically to fit it in the available space;

Bold: select this option if the text has to be printed in bold;

Italic: select this option if the text has to be printed in italic.

IMAGES CONFIGURATION

Number of images to load: if no images have to be included, just keep 0; otherwise, select the number of image fields in use from 1 to 3; for each image, it is possible to specify:

Name associated to image: this is the name given to the image field by the software of the printer, and can be found by opening the label template through a text editor;

Path of image: the path of the file to be inserted in the image field of the label.

4.4.36. ProcessVariables action

The *ProcessVariables* action makes it possible to process certain global variables of the program, given as input parameters, to write others as output, through plug-ins that must be developed separately. The use of this action presupposes knowledge of how global variables work and the existence of a plug-in in PPL format (LabVIEW packed project library) to perform the desired operations, whether developed independently by the user or commissioned to STEP Lab.

The specific parameters that can be configured for a *ProcessVariables* action are:

MAIN SETTINGS

Path of the file to use to process the data: here the user specifies the path to the PPL file (with the extension .lvlibp) that will be invoked when executing the action to transform input variables into output variables. If you wish to develop one of these files yourself, you should refer to STEP Lab's technical support for information regarding the features the plug-in must have.

INPUT PARAMETERS

This is the list of global variables that will be passed as input to the calculation plug-in; each item in the list can be populated using the drop-down control or by importing a variable from the list on the right using the  button; by clicking on an item in the list, it is possible to move it up or down using the two arrows visible immediately to the right of the list itself; the order of the items is important for the execution of the PPL plug-in.

OUTPUT PARAMETERS

This is the list of global variables that will be written by the PPL plug-in from the values of the input variables; its composition works in the same way as that of the input parameter list.

ADVANCED SETTINGS

Global variables update waiting: if this option is set to ON, when the *ProcessVariabiles* action is executed, an update of the values of all global variables from the field is commanded before the PPL plug-in is invoked, so that there is no risk of performing calculations with obsolete input values.

4.4.37. Recipe action

The *Recipe* action allows a recipe file to be loaded, i.e. having the extension .rcp, which contains a set of test parameters collected together via the customised interface called *Dynamic HMI*.

The specific parameters that can be configured for a *Recipe* action are:

MAIN SETTINGS

Recipe loading mode: if the configured recipe loading is manual, the action works as follows: at each execution, it checks whether the user has in the meantime manually loaded a recipe via the *Dynamic HMI*, and if so, it stops the test sequence to allow the parameters of the loaded recipe to be sent to the controller (in this way, the test variables are modified under controlled conditions), and then resumes it when this operation is finished; if, on the other hand, the action is used with automatic loading, the programme automatically loads the desired recipe (defined via the following fields) each time the *Recipe* action is executed;

Recipe path assignment mode: visible only in the case of automatic loading, allows you to choose whether to directly set a predefined and always the same path for the recipe file to be loaded, or to produce it dynamically by specifying a source folder and a variable defining the name of the recipe file;

Path of the recipe to load: if the action has been configured to automatically load a recipe with a pre-assigned path, from here the user can choose which recipe file to use;

Starting folder path: if the action has been configured to automatically load a recipe with a dynamically composed path, this defines the source folder in which the software should search for the recipe file; all subfolders of the specified folder are also included;

Variable giving the name of the recipe to load: if the action has been configured to automatically load a recipe with a dynamically composed path, this is the variable that defines the name of the recipe: when the action is executed, the software searches in the source folder and any subfolders thereof for an .rcp file with a name exactly equal to the value that the chosen variable has at that moment.

ADVANCED SETTINGS

Saving of recipe into TDMS file: If this option is enabled, the software saves the loaded recipe in a dedicated group of the TDMS file containing the test results;

Name of the group where to save the recipe: if saving the recipe is enabled, here the user can specify what name to give to the group in which this is to be done.

4.4.38. LoadTDMS action

The *LoadTDMS* action allows a TDMS file to be automatically loaded into the post-processing environment, possibly opening the latter if it is closed when the action is executed.

The specific parameters that can be configured for a *LoadTDMS* action are:

MAIN SETTINGS

File to load: from here, it is possible to choose whether the file to be loaded into the post-processing environment through the execution of the action should be:

- The TDMS file in use at the time the action is executed;
- The TDMS file prior to the one in use at the time the action was performed (this option can only work if a *ChangeTDMS* action is used prior to the *LoadTDMS* action to archive one TDMS file and start writing another);
- A TDMS file which path is statically specified within the action configuration interface;
- A TDMS file of which the user will be asked to choose the path via a dialogue box that the software will open when the action is performed.

Path of the TDMS file to load: if the action is configured to load a file whose path is statically defined within the action itself, from here the user can set this parameter;

Management of possible files already in memory: it is possible to choose whether to keep open any other TDMS files already loaded in the post-processing environment at the time the action is executed, or to automatically command them to be closed;

Path of the setup file to load: the post-processing environment gives the possibility of saving files with the extension .setup, containing the configuration of custom channels and graphics; by means of this control, it is possible (but not mandatory) to choose one of these files, which the DataViewer will automatically load together with the desired TDMS file.

DATAVIEWER CONFIGURATION

DataViewer settings configuration: by activating this option, the following parameters are displayed and it is also possible to use the action to send configuration settings to the DataViewer software; it should be noted that these commands are only sent if the *LoadTDMS* action opens the post-processing environment when it is executed, not if it is already open;

Visualization type: the user can choose which type of display to adopt from those available (for more information on display types, see the DataViewer manual and tutorials);

Max nr of TDMS files to keep open (0=inf): from here the user can choose a maximum number of files to be kept in memory by the DataViewer: if a non-zero number X is specified, starting with the X+1-th file that is loaded, the DataViewer will automatically begin to close the file whose loading is the oldest, so that the number of files opened together never exceeds X.

ADVANCED SETTINGS

Wait for procedure conclusion: by activating this function, the test sequence stops in the *LoadTDMS* action until the loading of the file into the post-processing environment is completed.

4.4.39. SaveVariables action

The *SaveVariables* action allows any number of global programme variables to be saved within the TDMS file in use.

The specific parameters that can be configured for a *SaveVariables* action are:

MAIN SETTINGS

Saving type: it is possible to choose whether the variables saved in the TDMS file by this action are to be written in the form of channels or properties: channels are data sets similar to those saved during test execution for quantities such as load, position and velocity; properties are individual values (a property is composed of a name and a value), written in the form of strings, which can be associated with the file, a group of it or a channel of it; to explore these concepts in more detail, it is necessary to consult documentation on the structure of TDMS files, such as the following page: https://www.ni.com/en/support/documentation/supplemental/06/the-ni-tdms-file-format.html

File element to which the property belongs: if variables are to be saved as properties, here it is possible to choose whether they will be properties belonging to the file, to a group (in which case it must be specified which group) or to a channel (in which case it must be specified which group and which channel in that group);

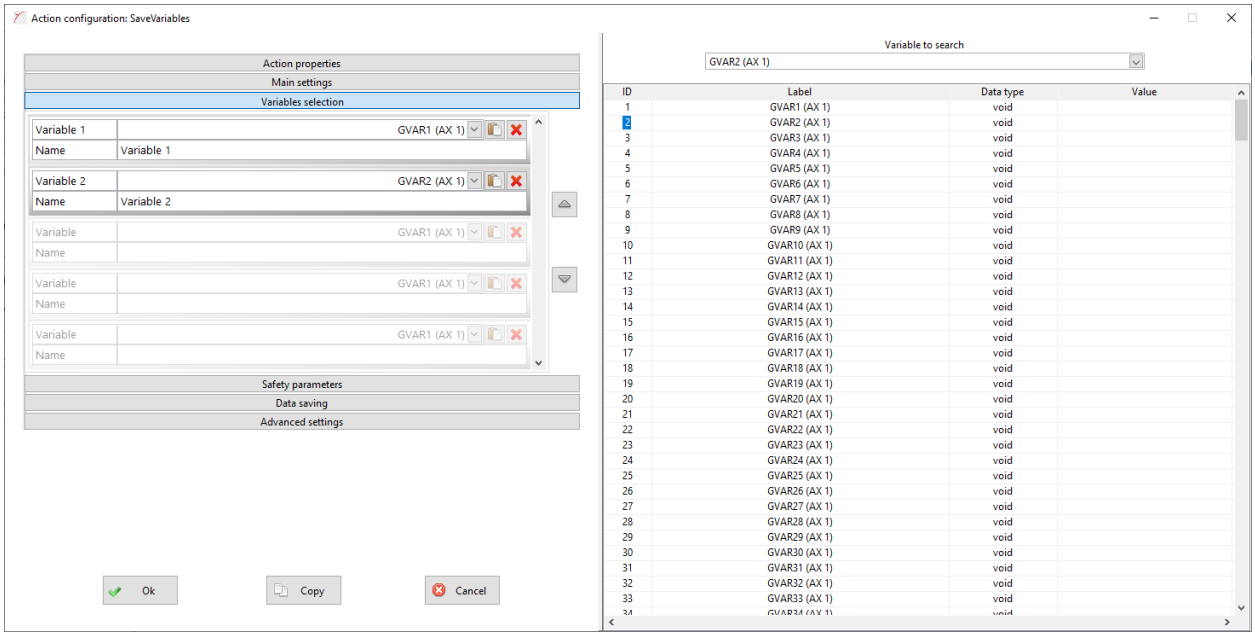
Group name: this field is visible in three cases:

- If saving as a channel is selected: then it defines the name of the group within which the channels for the variables to be saved will be created;
- If saving as a group property is selected: then it defines the name of the group to which the properties consisting of the names and values of the saved variables will belong;
- If saving in the form of a channel property is selected: then it defines the name of the channel group to which the properties consisting of the names and values of the saved variables will belong.

Channel name: this field is only visible if saving in the form of a channel property is configured, and defines the channel to which the properties consisting of the names and values of the saved variables will belong.

VARIABLES SELECTION

This section of the menu represents the list of variables to save:



Variables can be imported into the list from the list visible on the right by pressing the buttons available for each item, or removed from the list using the buttons; using the two arrows on the side, you can change the order of the list by clicking on a variable (whose frame will light up a darker grey colour than the others) and then pressing the up or down arrow: the variables are saved in the order in which they appear in this list; finally, each variable can be associated with a name, which will be the name of the channel or property (depending on how saving is configured) relating to the variable; if no name is configured, the software uses the name assigned to the global variable.

ADVANCED SETTINGS

Global variables update waiting: if this option is enabled, the software will read the values of the global variables from the field before saving, in order to guarantee that what is written in the file is up-to-date; it is possible to disable this function in order to speed up the execution of the action, but you must be sure that the variables to be saved will have up-to-date values when the action is executed (it must be taken into account that generally in the TestCenter the updating of global variables by reading from the field occurs approximately every 2 seconds).

4.4.40. SaveDatabase action

The *SaveDatabase* action allows to save any number of parameters within a SQL database table. In order for this action to work properly, you must have installed the Microsoft SQL Server Management Studio environment and configured the STEP Lab database and the communication between the TestCenter and this database (this is done via configuration files to change which requires STEP Lab technical support).

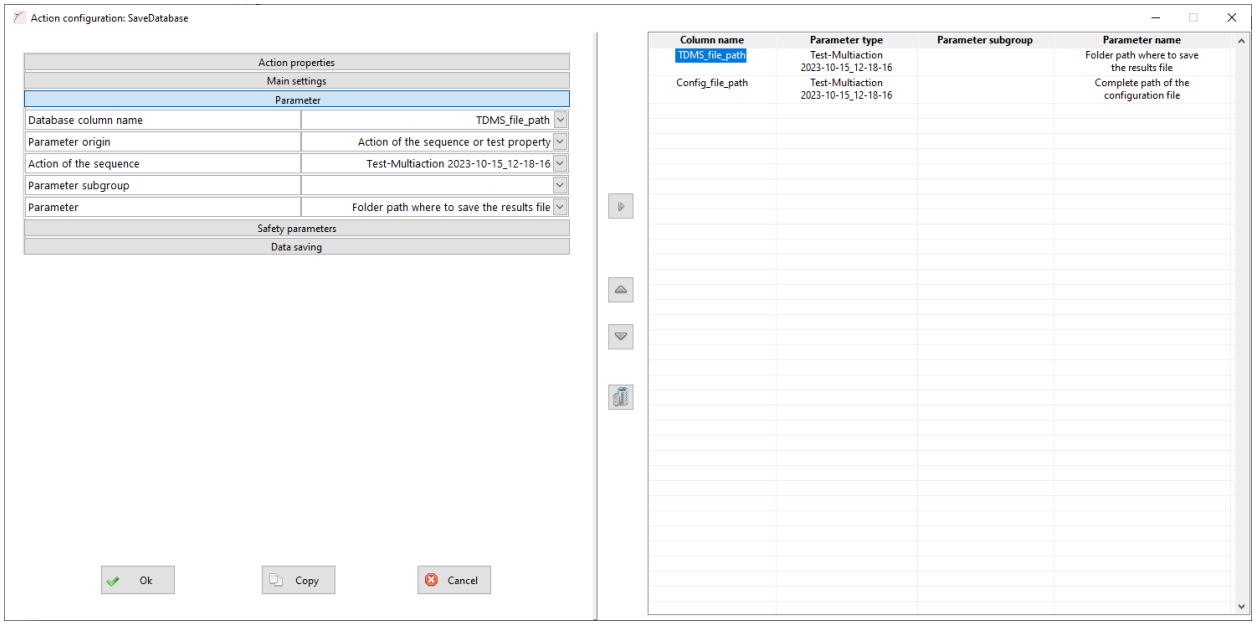
The specific parameters that can be configured for a *SaveDatabase* action are:

MAIN SETTINGS

Name of the database table to write: here must be written the name of the table, belonging to the STEP Lab SQL database, in which the parameters are to be saved; the table must already be physically present in the database with this name, it is not created by the TestCenter software; when this name is configured, the TestCenter automatically queries the SQL database to find out the names of the columns of which the table is composed, and sends an error message if it does not find valid names.

PARAMETER

From this section of the menu, the parameters to be saved in the database are configured, one at a time: once the configuration of a parameter is complete, it can be saved in the table on the right by pressing the button with the arrow pointing to the right; the elements in the table can then be reordered or deleted by clicking on them and using the buttons with the arrow and basket icons; if a parameter with the same column name as another already selected is imported into the table, the latter is overwritten.



Database column name: this drop-down menu is automatically populated by the software with the names of the existing columns in the table that is selected from the primary settings field;

Parameter origin: the parameter that has to be changed can be part of the multiaction test sequence or part of the measurements configuration in case the *Multiaction with measurements* test is in use;

Action of the sequence/Parameter subgroup (or Measurement element), Parameter: the combination of these settings identifies the parameter to be updated in real time.

4.4.41. AOCoupling action

The *AOCoupling* action allows to configure a coupling between an analog output channel and an acquisition channel: after its execution, the controller commands the analog output a value that is a function of the coupled acquisition channel, according to an association table that is configured within the action itself.

This action can only be used if the machine has analog outputs.

The specific parameters that can be configured for an AOCoupling action are:

MAIN SETTINGS

Analog output to command: this drop-down menu is automatically populated with all available analog output channels, and allows the user to select the one to act on;

Operation to command: it is possible to set the association with an acquisition channel, but also to interrupt it;

Acquisition channel to associate: this drop-down menu, visible only if the action is used to make an association and not to end it, is automatically populated with all the acquisition channels available to be associated with the analogue output.

ASSOCIATION TABLE

This table, represented in the graph on the right in the form of a preview, represents the law that links the value of the associated acquisition channel (independent variable) to the value to be commanded for the set analog output (dependent variable): for as long as the association is active during a test, the value of the analog output is commanded in real time according to the paired acquisition channel following this law.

Association table origin: the table can be constructed by writing the individual points in it one by one, or imported from a file previously saved using the *Save association table as file* button visible immediately below the table itself;

File to load: if the association table is to be imported from file, this is the path to the file to be loaded;

Immediately below the table are two buttons: the one called *Sort table* sorts the configured points by increasing values of the independent variable (for reading convenience only), while the one called *Save association table as file* is used to save an xml-format file containing the configured table, which can then be reused by loading it into another *AOCoupling* action without the need to rewrite it each time by hand.

4.4.42. DOCoupling action

The *DOCoupling* action allows to configure a coupling between a digital output channel and an acquisition channel: after its execution, the controller commands the digital output a value that is a function of the coupled acquisition channel, according to an association table that is configured within the action itself.

This action can only be used if the machine has digital outputs.

The specific parameters that can be configured for an *DOCoupling* action are:

MAIN SETTINGS

Digital output to command: this drop-down menu is automatically populated with all available digital output channels, and allows the user to select the one to act on;

Operation to command: it is possible to set the association with an acquisition channel, but also to interrupt it;

Acquisition channel to associate: this drop-down menu, visible only if the action is used to make an association and not to end it, is automatically populated with all the acquisition channels available to be associated with the digital output; it should be noted that the software filters the digital output channels by populating the menu only with those for which it detects the possibility of pairing, so it is possible that this drop-down menu does not contain a complete list of all digital output channels;

Hysteresis value: this value, expressed in the unit of measurement used by the software for the associated acquisition channel, defines the deviation from each switching value that must be reached before another command against the digital output can be given; in other words, if an ON-OFF or OFF-ON switching is set for an acquisition channel value of 100, with a hysteresis of 10, after having exceeded (in any direction) the value of 100 and having given the digital output the required value, no other command will be given until the value of the associated channel becomes lower than 90 or higher than 110; the purpose of this parameter is to prevent a fluctuation of the signal around a switching value from causing high-frequency opposite commands to be sent to the digital output, which could damage the hardware.

ASSOCIATION TABLE

This table, represented in the graph on the right in the form of a preview, represents the law that links the value of the associated acquisition channel (independent variable) to the value to be commanded for the set digital output (dependent variable): for as long as the association is active during a test, the value of the digital output is commanded in real time according to the paired acquisition channel following this law.

Association table origin: the table can be constructed by writing the individual points in it one by one, or imported from a file previously saved using the *Save association table as file* button visible immediately below the table itself;

File to load: if the association table is to be imported from file, this is the path to the file to be loaded;

Immediately below the table are two buttons: the one called *Sort table* sorts the configured points by increasing values of the independent variable (for reading convenience only), while the one called *Save association table as file* is used to save an xml-format file containing the configured table, which can then be reused by loading it into another *DOCoupling* action without the need to rewrite it each time by hand.

ADVANCED SETTINGS

Minimum wait between two DO switches: if this wait is not null, after each switchover of the selected output the controller does not command any other switchover for a time equal to the value set here.

4.4.43. VariablesInterface action

The *VariablesInterface* action allows an interface to be opened during the execution of the test sequence, through which the user is asked to manually enter as input data certain test properties and/or global variables that are chosen through a special configuration file; only after the user has entered the required data and closed the window, can the test sequence resume.

The specific parameters that can be configured for a *VariablesInterface* action are:

MAIN SETTINGS

Interface configuration file path: from here the user must select the path to the configuration file containing information on which variables are to be set via the interface that will open when the action is executed; the interface is developed for the purpose of reading and interpreting this type of file, generating a list of parameters composed of description and value, which the user must set.

ADVANCED SETTINGS

Global variables update waiting: if this option is enabled, the software will read the values of the global variables from the field before saving, in order to guarantee that what is written in the file is up-to-date; it is possible to disable this function in order to speed up the execution of the action, but you must be sure that the variables to be saved will have up-to-date values when the action is executed (it must be taken into account that generally in the TestCenter the updating of global variables by reading from the field occurs approximately every 2 seconds).

4.4.44. Email action

The *Email* action allows e-mail messages to be sent at the point in the action sequence where it is used, provided the computer has an active Internet connection.

The specific parameters that can be configured for an *Email* action are:

MAIN SETTINGS

Message text: the text of the message to be sent;

Subject: the subject of the message to be sent;

Recipients: it is possible to enter more than one e-mail address to which to send the same message;

cc: it is possible to enter more than one e-mail address to put in the "knowledge copy" of the message;

Attachments: files can be attached to the message, which can be selected via this path list.

ADVANCED SETTINGS

Save message as property into TDMS file: if this functionality is activated, the software also saves the configured message text in the TDMS results file, in the form of the save group properties relating to this action;

Use custom outgoing mail server: it is possible to use the default outgoing mail server configured for the programme or one of the user's choice, provided it has been appropriately parameterised in the *Software options* section of the main menu.

4.4.45. SavingTrigger action

The *SavingTrigger* action allows a condition to be configured that, if subsequently verified, triggers a data saving event in the TDMS results file.

The specific parameters that can be configured for a *SavingTrigger* action are:

MAIN SETTINGS

Dedicated saving group name: for each save due to the occurrence of the trigger, the software creates a new group in the TDMS file, the name of which is defined by this field; any multiple saves are distinguished by adding a date and time to the group name to make it unique;

Comparison that activates the trigger: by activating the appropriate tick, from here the operation is defined which, carried out in real time on the selected channel, defines the occurrence of the save trigger: for example, with sign > and numerical value 3, the trigger occurs as soon as it is detected that the monitored channel has assumed a value greater than 3;

Recording time before trigger: the software constantly keeps a limited amount of data in memory so that it is also possible to record a certain interval of time immediately prior to when the trigger occurs; if the tick contained in this field is disabled, only data from when the trigger occurs will be recorded, ignoring all previous triggers; the maximum value that can be set in this field depends on the parameters contained in the TestCenter configuration files: if the user wishes to extend the permitted range, it is necessary to contact STEP Lab technical support to check whether the request can be fulfilled;

Post-trigger acquisition duration: this value defines how long the software must save data from the moment the trigger occurs;

Saving management mode: the *SavingTrigger* action can be configured to save data only at the first occurrence of the set trigger, or each time such an event occurs since the action itself is executed;

Behaviour with respect to active triggers: since several *SavingTrigger* actions can be used in the same sequence, each can be configured to add to or replace any triggers already in memory due to possible *SavingTrigger* actions already executed.

DATA SAVING WHILE THE ACTION IS EXECUTING

This is the common section of the menu that is generally called *DATA SAVING* for the other actions, but in this case it is necessary to distinguish it from the section which configures the saving related to triggers, which is the following.

DATA SAVING RELATED TO THE TRIGGER

This section determines the save configuration that occurs when the trigger occurs.

Sampling frequency: this ring menu is effective only if saving is activated for the action, and sets the sampling frequency of the data for logging during that action; the frequency can be chosen only among the given values;

Saving group name: if the user writes some text here, this becomes part of the data group dedicated to the current action on the results file, together with the action label;

Include action label in group name: if this option is active, the label of the action becomes part of the group name, otherwise it is ignored.

4.4.46. PSD action

The *PSD* action allows a profile to be reproduced, which, unlike the *ProfileTracking* action (in which the time profile is composed by points by the user or imported from an external file), here is automatically generated by the software in such a way as to obtain the desired PSD, which is the input parameter for this action.

The specific parameters that can be configured for a *PSD* action are:

MAIN SETTINGS

Total procedure duration: here the time is specified, in hours, minutes and seconds, during which profile reproduction continues; after this time, the action ends;

Feedback type: the user can choose whether to work in load or position control;

Acquisition phase duration: in order to understand this parameter and the following one, it is necessary to know how the profile generation and reproduction algorithm works: initially, a profile section is generated such that it reproduces the desired PSD and consists of three sections a transition phase, which serves to pass from the condition of the actuator stopped to the next phase, which is called acquisition phase, because it is the portion of the profile from which the software will carry out the feedback, i.e. it will process the corrections to be made to the profile itself in order to make the PSD as close as possible to the desired one; the third section, called calculation, has this name because it is the section of the profile during which the TestCenter is processing the data detected during the acquisition phase and is calculating a new profile with the corrections necessary to obtain a PSD as close as possible to the objective; In this way, a new section of time history is obtained, which in turn is composed of a transition (to connect the end of the previous section to the new acquisition phase), an acquisition phase and a calculation phase during which the TestCenter processes when obtained from the acquisition phase to construct the next section of the profile, making the cycle begin again;

Calculation phase duration: in addition to what has been described above, the calculation phase must last a sufficient time to ensure that the TestCenter can process the new temporal history section and send it to the controller before the calculation phase ends; the most important variable for an order of magnitude of this time is the duration required for the transition (parameter described later): with durations of less than 0.5 s, it is generally possible to limit oneself to calculation phases of around 20 seconds, but this indication should only be taken as an order of magnitude: depending on the characteristics of the computer used, any other operations that the software may be carrying out in parallel, and other factors that may affect performance (such as the presence of other programmes running), it is possible that this time may be shortened or may have to be lengthened;

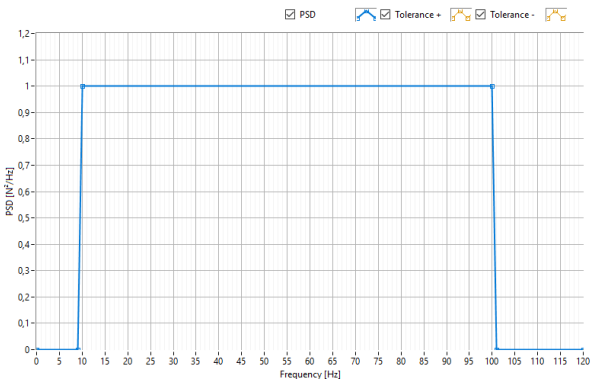
PSD definition mode: similar to the profiles of the *ProfileTracking* action, the PSD can also be constructed by points, or imported from a file;

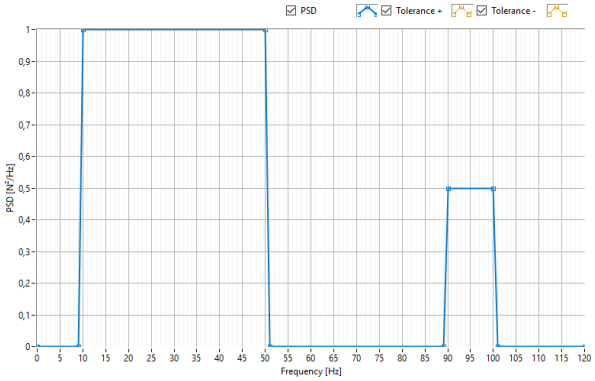
Offset to apply to time history: since the algorithm for generating the time history from the given PSD always obtains a profile with 0 mean, this parameter allows mean value to be varied: for example, setting a PSD in load with an offset of 500 N, a value of 500 N will be added to the entire load history obtained.

FEEDBACK ALGORITHM CONFIGURATION

Duration of transition between two profile sections: as described above, the duration of the transition segment between two consecutive portions of the profile is a critical parameter in terms of the time the programme takes to compute the time history: in general, there is no particular advantage in setting high values for this time, which can normally be kept below 0.5 s;

Type of feedback algorithm to use: it is possible to choose whether to use a PID- or FIR-type algorithm to correct the time history in order to bring the obtained PSD as close as possible to what is configured; while the FIR-type algorithm converges faster, it's not suitable to regulate PSD targets where the excited frequencies are not continuous: in this case the PID algorithm works better; the following two images show examples of PSD with continuous (on the left) and non-continuous (on the right) excited frequencies, respectively:





Proportional gain: this parameter, only visible when selecting a PID-type feedback algorithm, represents the proportional gain of the algorithm;

Filter order: this parameter, only visible when selecting a FIR-type feedback algorithm, represents the order of the filter to be used, a higher order usually allows for a better convergence between feedback and target PSDs at the cost of a longer computation time, mind that after a certain value (depending on PSD target and feedback) increasing the order has no effect on convergence and only increases computation times. The value must be greater or equal to 2, we suggest to use a value between 64 and 256, the default value is 128;

Forgetting factor: this parameter, only visible when selecting a FIR-type feedback algorithm, represents the filter forgetting factor, i.e. the factor greater than 0 (0 is not an admissible value) and less or equal to 1 that gives exponentially lower weight to the oldest error values, default is 0.95;

Regulation factor: this parameter, only visible when selecting a FIR-type feedback algorithm, represents the filter adjustment factor used to initialize the FIR estimation; regulation factor must be greater than zero, it's suggested to keep its value near to zero, default value is 10^{-5}.

DATA SAVING

In addition to the normal saving parameters common to the other actions, this one provides two others, based on the fact that data recording for this action takes the form "saving data for a certain time x at each time interval y":

Duration of each saving: this is the duration of each saving interval;

Saving periodicity: this is the time interval between the start of two consecutive saves.

To give a numerical example, by setting these two parameters to three minutes and one hour respectively, the software will save data in the TDMS results file for three minutes consecutively for each hour of testing.

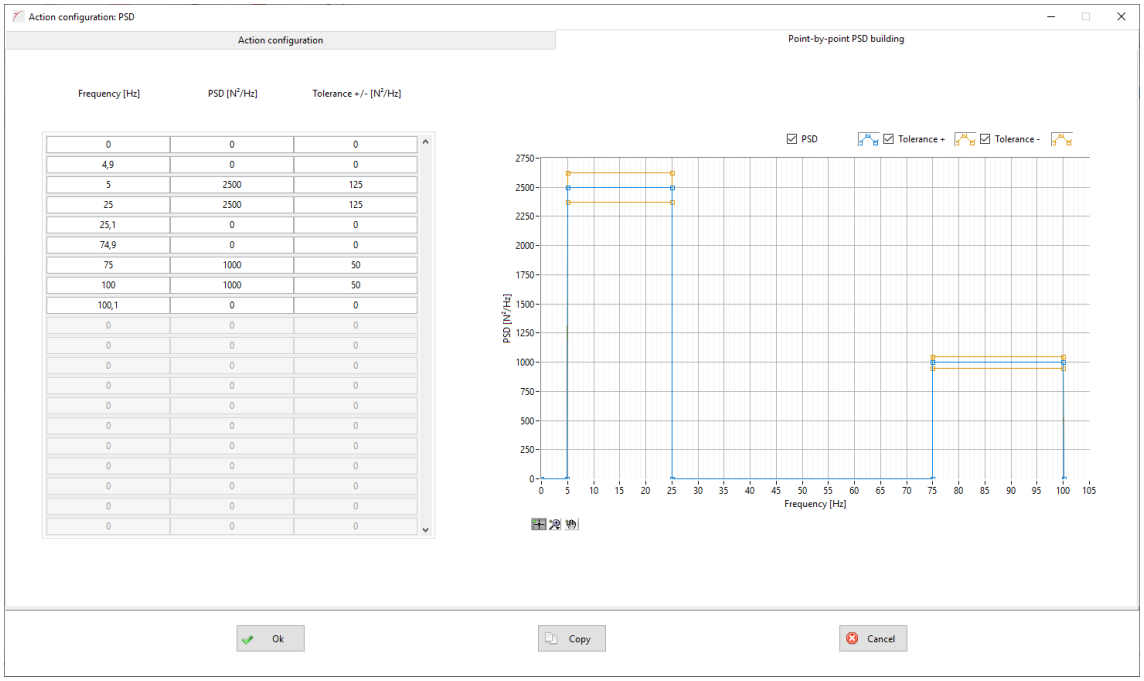
PRELIMINARY AUTO-TUNING

Preliminary auto-tuning activation: If this option is set to ON, the actuator performs a preliminary calibration before starting to reproduce the profile obtained from the PSD to be reproduced; this procedure is always recommended when working in load control, while it is not necessary when feedback is set in position;

Setpoint range % to use for auto-tuning: if the auto-tuning is active, this parameter sets the range to be used for this procedure, which is calculated with respect to the setpoint range (difference between maximum and minimum value of the setpoint in the time history to be reproduced) and centred on the initial value of the time history itself, similarly to the *Pattern* and *ProfileTracking* actions;

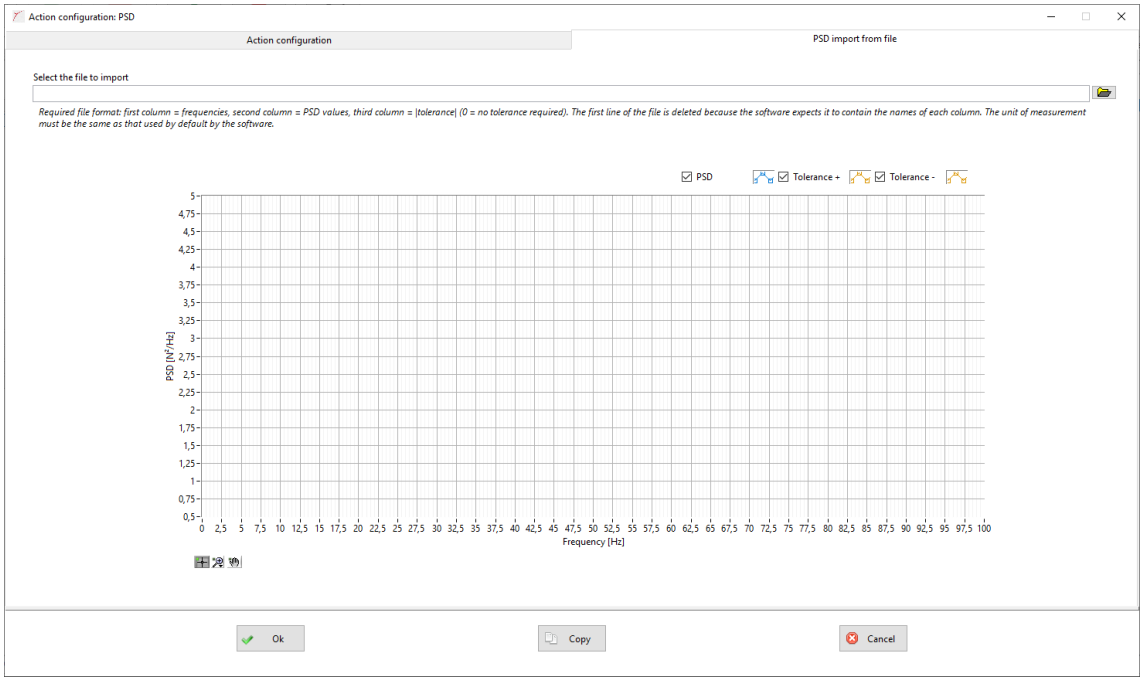
Speed for auto-tuning and movement to initial setpt.: this is the speed of the actuator during auto-tuning movements and during the movement which brings it to the initial setpoint value of the profile to be reproduced.

If the user chooses to configure the PSD by points, the second tab of the action interface is presented with a table on the left and a preview graph on the right:



In the table it is possible to write the PSD values to be associated with each frequency, together with tolerances that define the attainment of the commanded PSD; for each PSD value, the respective tolerance is unique and is added to obtain the upper limit and subtracted to obtain the lower limit. The preview graph automatically shows the values entered in the table, with one set of data for the commanded PSD and one for each tolerance (upper and lower).

Instead, if the action is used with the intention of importing a PSD from file, the second tab of the interface looks like this:

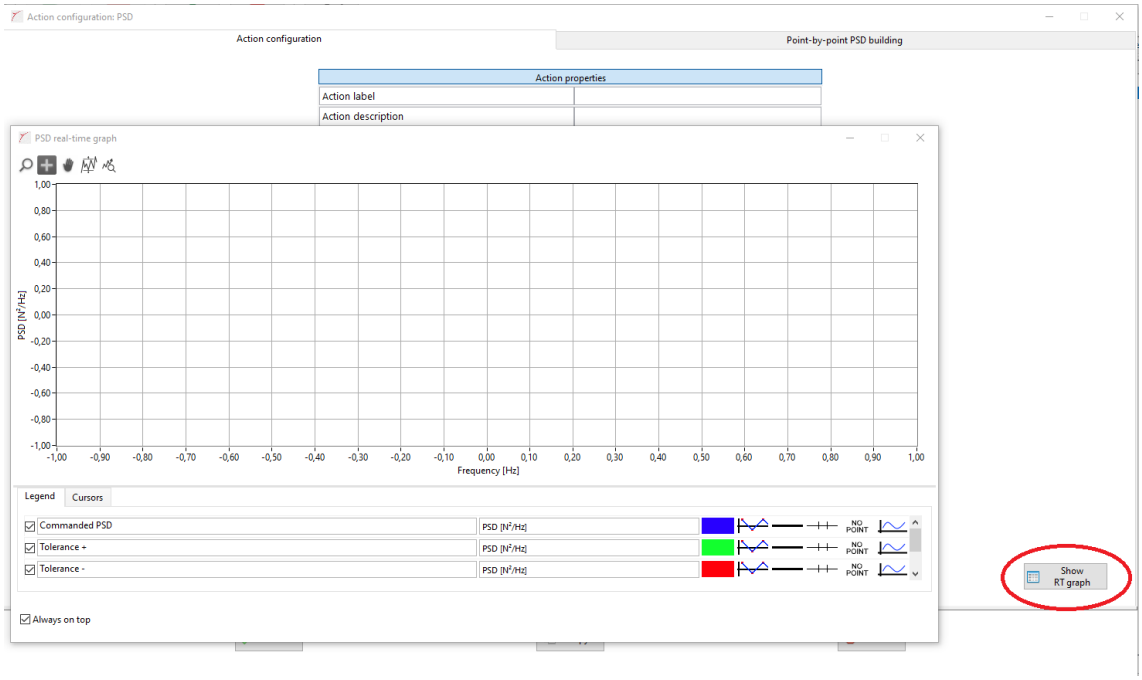


At the top is the path to the file to be imported, and in the middle a preview graphic. Immediately below the file path, a short message summarises the characteristics required for a file to be imported:

Required file format: first column = frequencies, second column = PSD values, third column = |tolerance| (0 = no tolerance required). The first line of the file is deleted because the software expects it to contain the names of each column. The unit of measurement must be the same as that used by default by the software.

In other words: the content of the first line does not matter, because the software deletes it a priori expecting it to contain the names of the data series; in the first column there must be the frequency values, in the second the PSD values and in the third the tolerances in absolute value; in addition, the separator between the columns must be a tab character. You can import files in txt or csv format, but you must respect the unit of measurement that the software uses by default for each channel (load → N; position → mm).

Starting with version 7.2 of the software, there is also a button in the *PSD* action configuration interface that shows a graph with the commanded PSD and the PSD obtained from real-time data processing; this graph is only updated during the execution of the action itself.



4.4.47. BarCode action

The *BarCode* action is used to control the acquisition via a barcode and/or QR code scanner, depending on the type of device.

It is necessary to refer to STEP Lab technical support to verify compatibility between a generic scanner and the TestCenter software.

The specific parameters that can be configured for a *BarCode* action are:

MAIN SETTINGS

Device selection: the scanners connected to the computer are not automatically recognised by the TestCenter software: the list of devices that appears here results from the configuration of a program file that requires STEP Lab's support in order to be modified; if several scanners are connected at the same time, if the “all” option is selected, the software waits for the first useful reading, from whichever of the devices it comes from;

Scan mode: this function is always visible but requires a scanner that supports the reception of automatic commands from the software, as it allows the user to choose whether to manually perform the reading or whether the software should drive an automatic read command to the scanner (which would simulate the user pressing the typical read button on the device).

ADVANCED SETTINGS

Removal of any spaces from the read code: it may happen that some types of scanners, especially those that reproduce the behaviour of a virtual keyboard, output alphanumeric codes containing unexpected spaces; this option can be used to make the software remove them automatically.

4.4.48. Side button bar

On the right side of the multiaction interface there are some buttons, with the following functionalities:

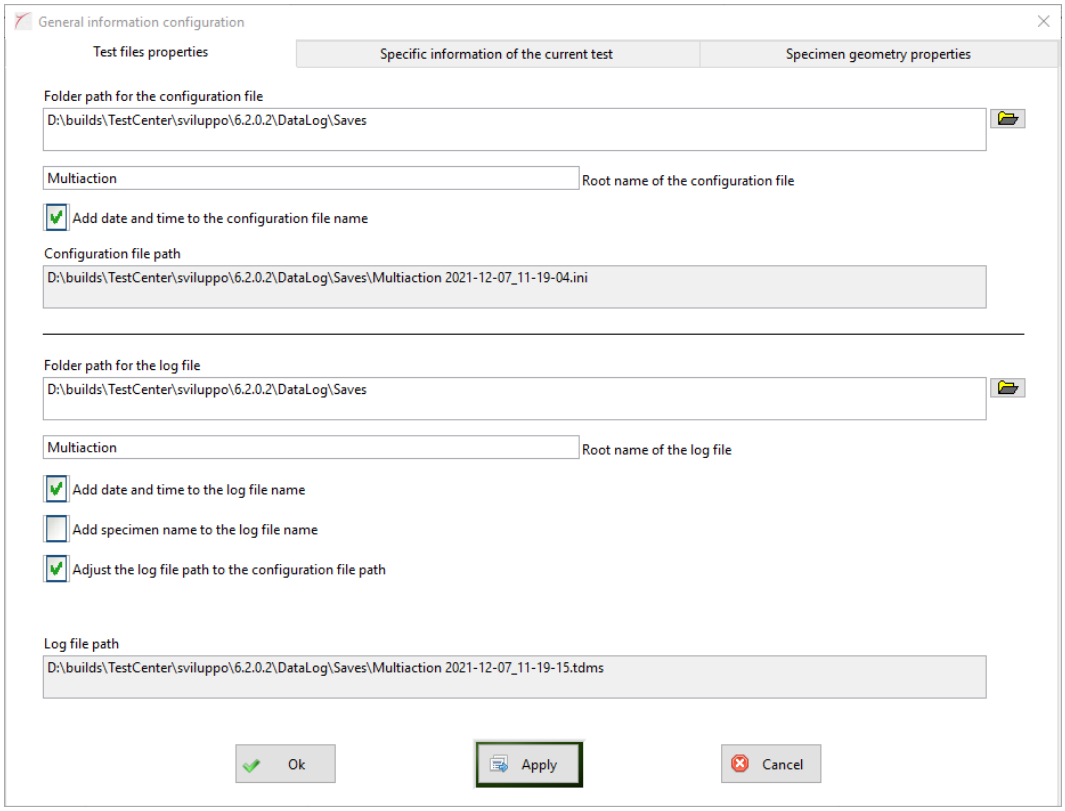
This button deletes the action that is currently selected on the tree menu.

This button allows the user to save the action that is currently selected on the tree as a template action, so that it can be recalled with all the parameters already configured as they are at the moment of saving (similarly to what happens with the test templates).

This button allows to deactivate various actions of the sequence without needing to delete them; deactivated actions are not executed by the controller during the test, and are displayed in italic and grey in the tree:

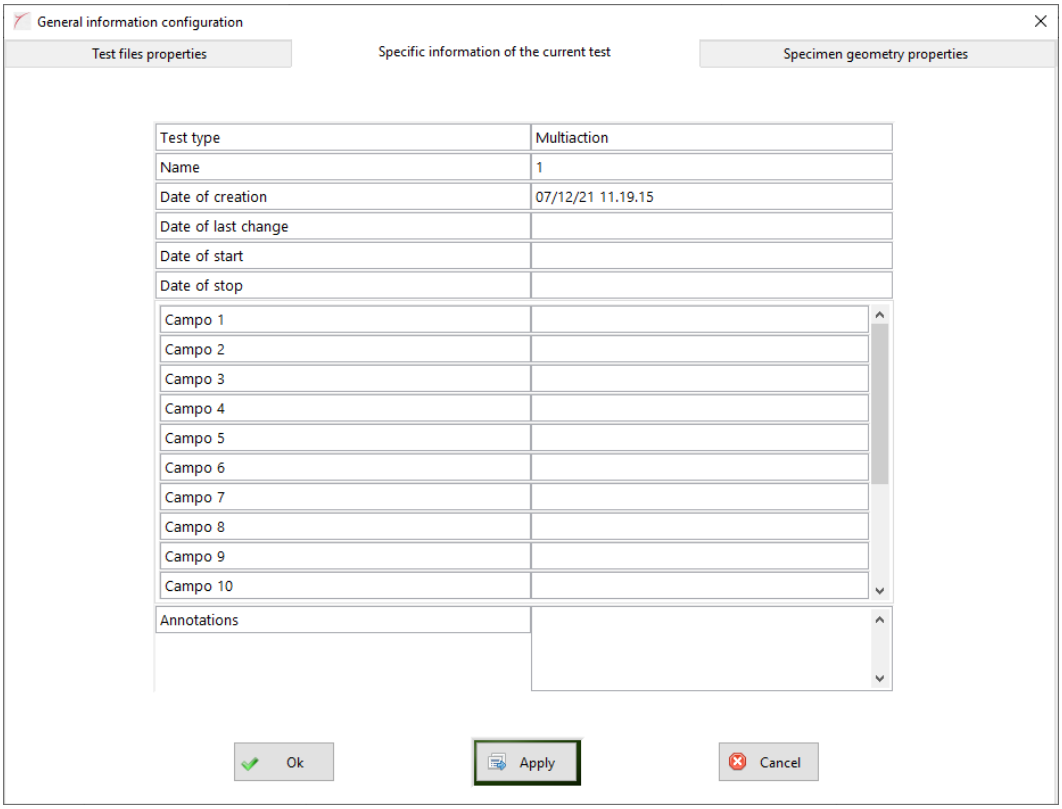
The same button, used on a deactivated action, reactivates it.

This button opens the interface used to configure the general properties of the specimen and the file paths options:

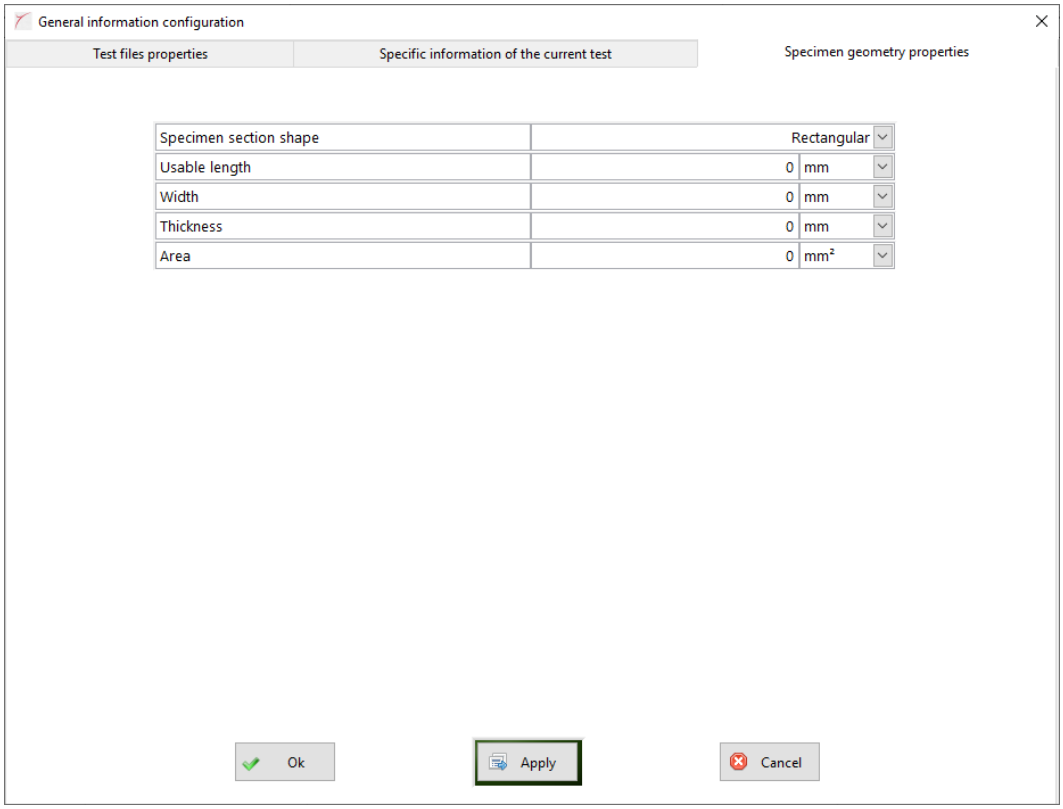


The first page of the interface is dedicated to the file paths settings: the upper section is for the test configuration file (.tc_mth), whereas the lower is dedicated to the test results files (.tdms); Please note that in multiaction tests a new tdms file is generated for each specimen created within the same configuration file, so a single *ini* file corresponds to as many *tdms* files as the specimens included in it are.

For each of the two files it is possible to select the folder path and the file name, and also choose whether to include the creation timestamp in the file name to make sure it is unique. Moreover, for the results file it is also possible to choose whether to include the specimen name in the file name, and there is also an option to automatically adapt the path and name of the results file to those chosen for the configuration file, so that it is not necessary to make the same modification twice if the user wants to keep the two files in the same place and with the same name.



The second page of the *General information configuration* interface is dedicated to the general properties of the selected specimen: as it happens in the other test protocols, here the user can choose a name for the specimen and write several custom fields in which the description can be also changed, so that everyone can insert what fits best for their needs (product code, operator, manufacturer, etc.).



The third page of the window is dedicated to the specimen geometry properties, which will be also saved in the TDMS results file by the software; it is possible to choose a section shape and configure the usable length, width and thickness. The area is calculated automatically for rectangular and circular shapes, whereas the user has to type it manually if a different shape option is selected.

These two buttons can be used to indent and outdent the elements in the tree; their functionality is only related to the graphic appearance of the action sequence and has nothing to do with the operation of the machine.

These two buttons can be used to expand or collapse the entire hierarchy of the multiaction test sequence tree.

This button opens the interface used to configure the *global variables* of the test.

This button opens an interface that can be used to search for the various global variables in the actions sequence of a multiaction test.

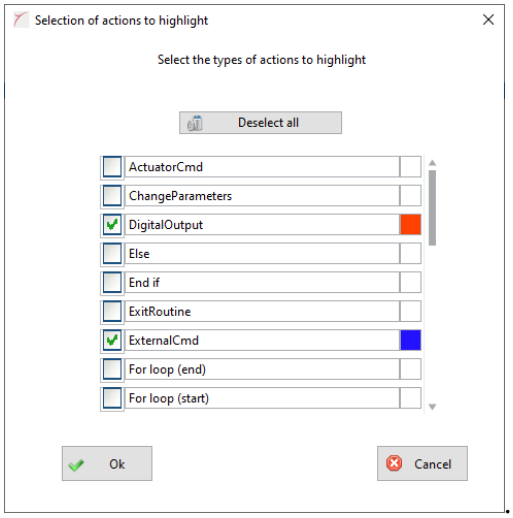
This button is needed only in the *Multiaction with measurements* test in order to open the measurements configuration interface.

This button opens the *dynamic HMI*, or *DHMI*, which is a custom interface that can be created in association to a multiaction test sequence including custom controls, indicators and graphs.

This button runs a procedure to check if the action sequence is fine or if it includes some errors: the output is provided to the user as a popup message.

This button opens an interface through which all the action of the sequence that belong to certain types can be highlighted in different colours (just to make it easier for the user to read the sequence, there is no effect on the machine operation).

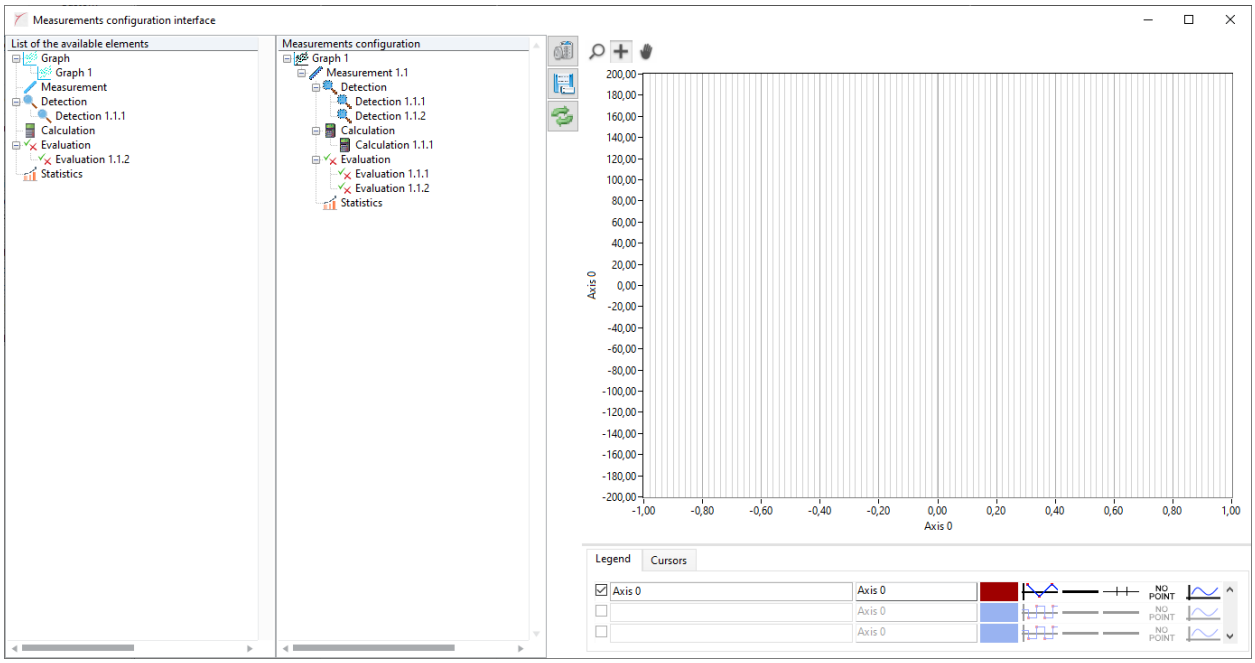
For example, in the case below all the *DigitalOutput* actions are going to be highlighted in red and all the *ExternalCmd* actions in blue. In order to clear any highlighting colour from an action type, just uncheck the checkbox dedicated to it.



This button allows to delete the entire sequence of action configured in the tree.

4.5. *Multiaction with measurements test*

The *Multiaction with measurements* test is a multiaction test that includes also the possibility to configure quality control measurements. The tree of the action sequence, all the actions, the side button bar and the test configuration is the same as the standard multiaction test; additionally, a separate window is provided for the configuration of the measurements:

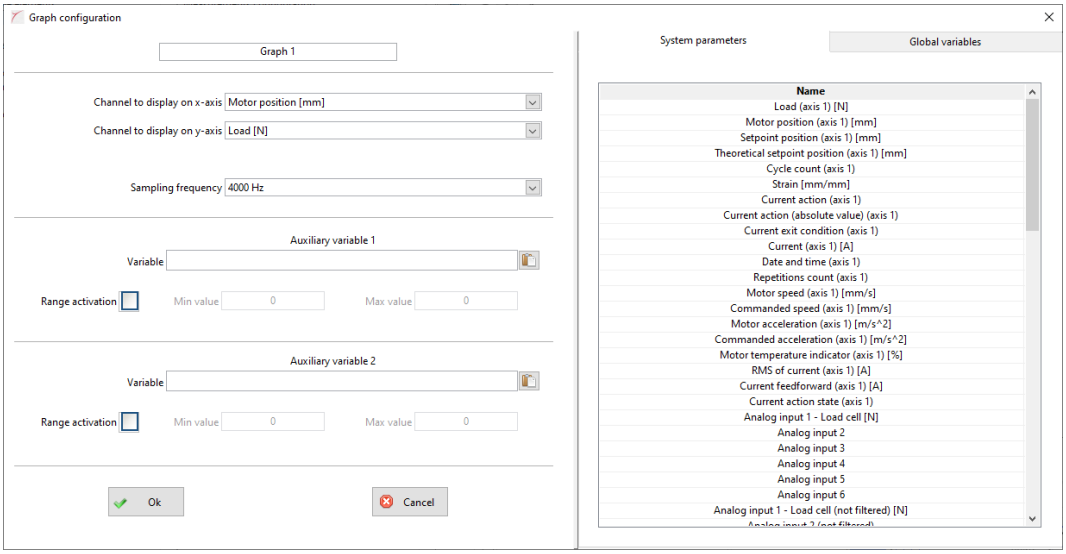


The quality controls that can be configured are articulated into *graphs*, *measurements*, *detections*, *calculations*, *evaluations* and *statistics*. The tree menu displayed on the left of the interface includes all the available elements of the various types, whereas the central tree menu is the configuration of the quality controls to execute; as in the multiaction test sequence, dragging and dropping from the left tree to the central tree is the procedure to add elements; if the root element of each category is selected for drag&drop, a default element of that type is inserted in the central tree; instead, the second level elements of the left tree (in the picture above, *Graph 1*, *Detection 1.1.1* and *Evaluation 1.1.2*) are templates saved by the user from existing items, that allow to include pre-configured elements into the central tree.

A *graph* can be dropped only at the root level of the central tree, not inside another graph; a *measurement* is the first-level child of a graph and it is composed by detections, calculations, evaluations and statistics; each of them can be dropped only in the third-level section dedicated to its type inside a measurement (for instance, it is not allowed to drop a calculation into the section dedicated to the evaluations).

By double-clicking an item of the central tree, its configuration interface is displayed.

4.5.1. Graphs

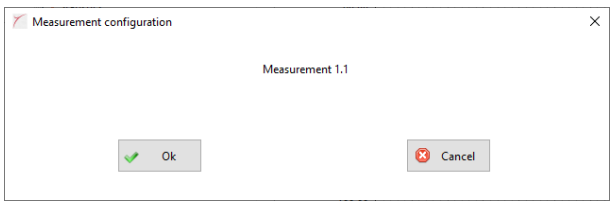


A graph is the fundamental element of a quality control, from which everything begins; its most important configuration parameters are the channels for x and y axes and the sampling frequency: all the controls that belong to the graph act on the channel that are chosen as x and y here, and the data to process are sampled at the frequency specified here.

The monitoring of the selected quantities and the execution of the various detections, calculations, evaluations and statistics starts with the *Measurement start* action and continues as long as the *Measurement stop* action on the same graph is not called in the multiaction sequence.

There is also the possibility to configure two *auxiliary variables* among those available on the *system parameters* and *global variables* lists, by selecting the desired variable on the right of the interface and copying it into the *Variable* text control through the  button: if the *range activation* checkbox is activated and two values are specified as *min value* and *max value*, the data processing of the graph requires the specified channel(s) value(s) to be inside the selected range(s).

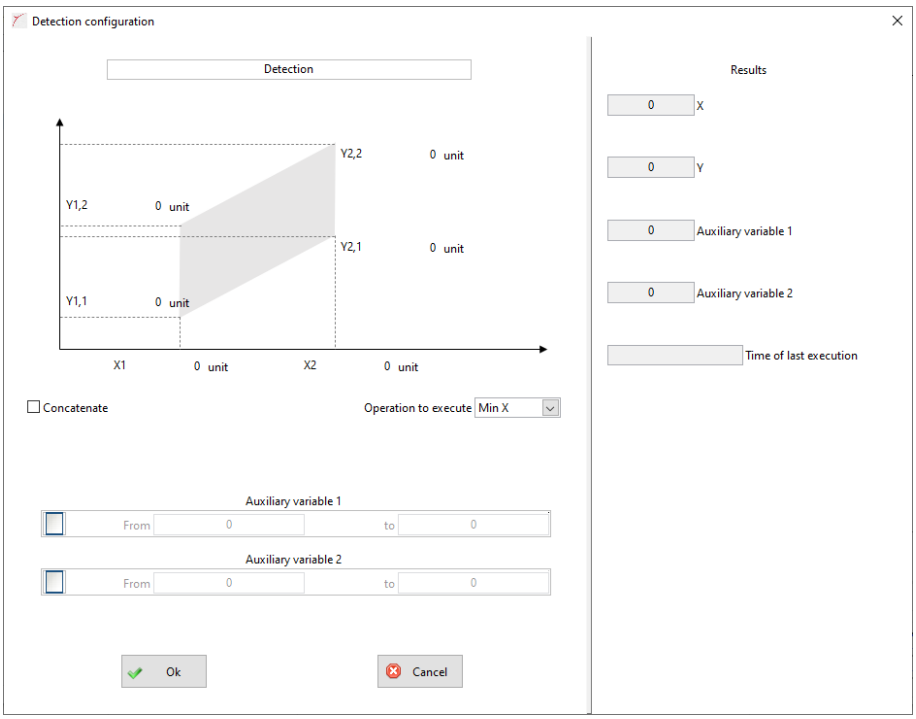
4.5.2. Measurements



Measurements are conceptual items only needed to gather the detections, calculations, evaluations and statistics that belong to the same control. There is actually no difference in creating 10 different measurements within one graph, each of them with only one detection and evaluation, or creating only one measurement with 10 detections and 10 evaluations; it is only a matter of how the items appear on the

interface and how the user feels more comfortable. This is the reason why the *measurement* configuration interface only provides a label.

4.5.3. Detections



A detection is an operation that provides a value as output, identifying it according to the parameters specified in the interface above; a measuring window has to be identified, the controller will isolate the data that belong to the window and run on them the operation chosen through the *Operation to execute* drop-down menu.

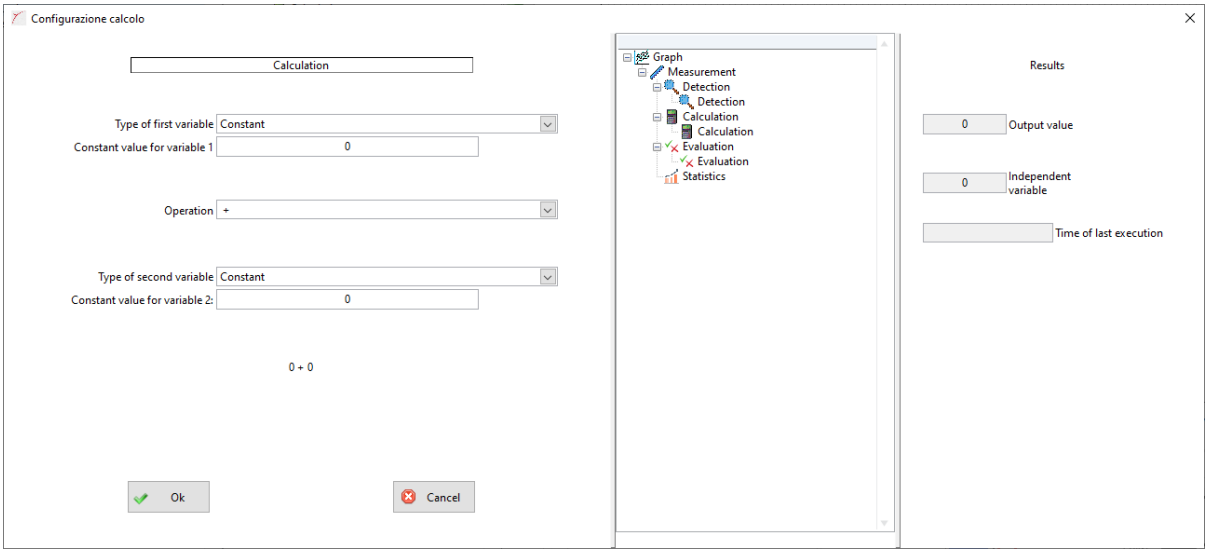
- X1, X2: initial and final values of the measuring window for the channel selected as X in the graph;
- Y1,1 and Y1,2: minimum and maximum initial values of the monitoring window for the channel selected as Y in the graph; unless the detection requires to identify a flux tube, these are also the final values of the region (which is therefore a rectangle; in other words, the two parameters of the following points are not in use);
- Y2,1 and Y2,2: minimum and maximum final values of the monitoring window for the channel selected as Y in the graph, only if the detection requires to identify a flux tube – otherwise Y1,1 and Y1,2 are used as final values;

> ***Information:*** *INITIAL and FINAL mean that the graph has to ENTER the measuring region through the INITIAL coordinates and EXIT through the FINAL coordinates, so it may be possible that the FINAL values are smaller than the INITIAL values.*

- "Concatenate" checkbox: if this is unchecked, in case the detection is executed multiple times only the last result counts; otherwise, the data provided by all the execution are taken into consideration for the final output;
- Operation to execute: it is possible to choose which type of identification has to be run:
    - Min X: calculates the minimum X value within the measuring window
    - Max X: calculates the maximum X value within the measuring window
    - Min Y: calculates the minimum Y value within the measuring window
    - Max Y: calculates the maximum Y value within the measuring window
    - Avg X: calculates the average X value within the measuring window
    - Avg Y: calculates the average Y value within the measuring window
    - X@Y: gets the X value at a specified Y coordinate, that has to lay inside the measuring window
    - Y@X: gets the Y value at a specified X coordinate, that has to lay inside the measuring window
    - Interval: collects all the data inside the measuring window
    - Max X&Y: calculates the maximum X and the maximum Y values within the measuring window
    - Min X&Y: calculates the minimum X and the minimum Y values within the measuring window
- Auxiliary variable 1 and 2: it is possible to choose a range for the two auxiliary variables configured in the *graph* interface, which will be a necessary condition for the detection to be executed; in other words, the detection will execute on all the data inside the measuring window acquired when the two auxiliary variables values were inside the ranges specified here.

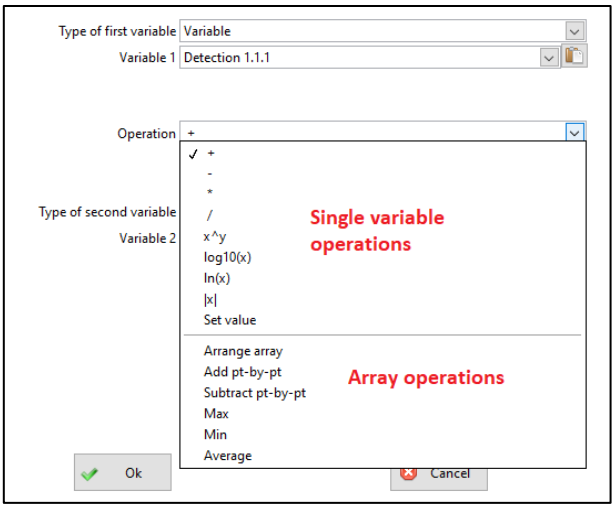
On the right of the interface, the results of the latest execution of the selected detection are displayed.

4.5.4. Calculations



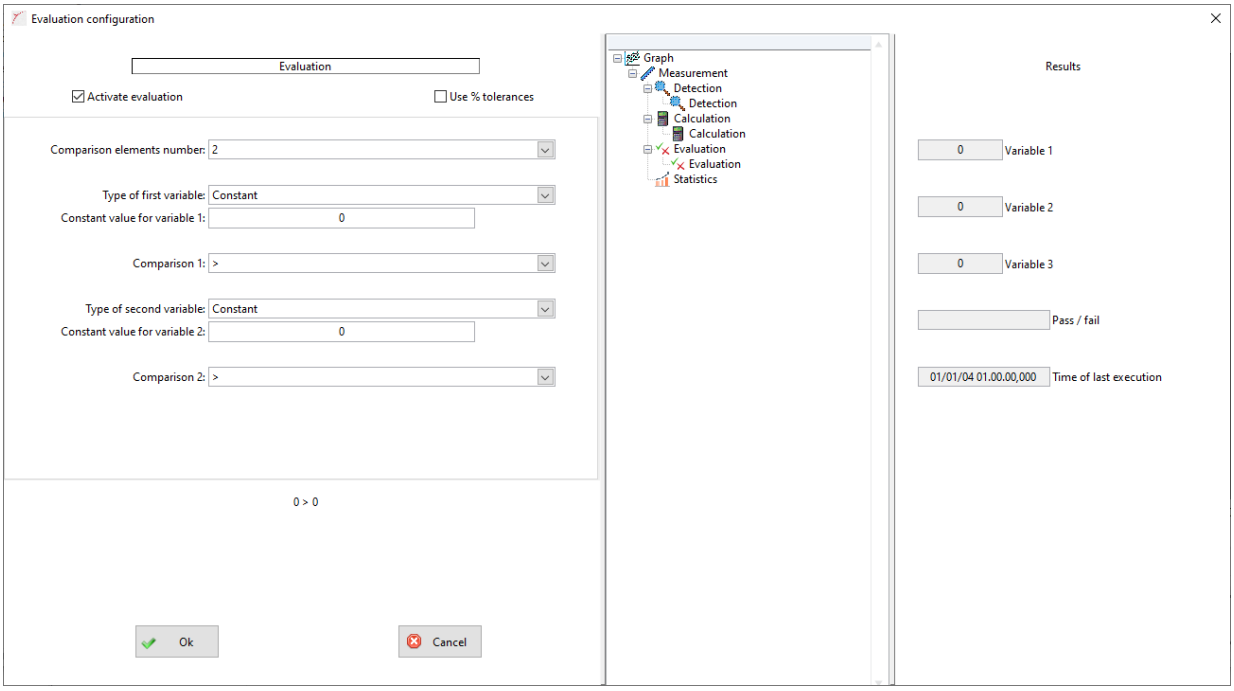
A calculation is a mathematical operation executed between detections or other calculations; for each of the variables involved it is possible to choose if it is a constant value or another measurement item, in which case it has to be selected in the tree on the right and copied into the *Variable (1 or 2)* control through the  button (the drop-down menu is disabled).

The operation to execute has to be configured through the dedicated drop-down menu that stays between the first and the second variable; the menu is divided into single variable operations, that act on values, and array operations, that act on arrays (i.e. results of *Interval* detections or other calculations operated on arrays).



On the right of the interface, the results of the latest execution of the selected calculation are displayed.

4.5.5. Evaluations



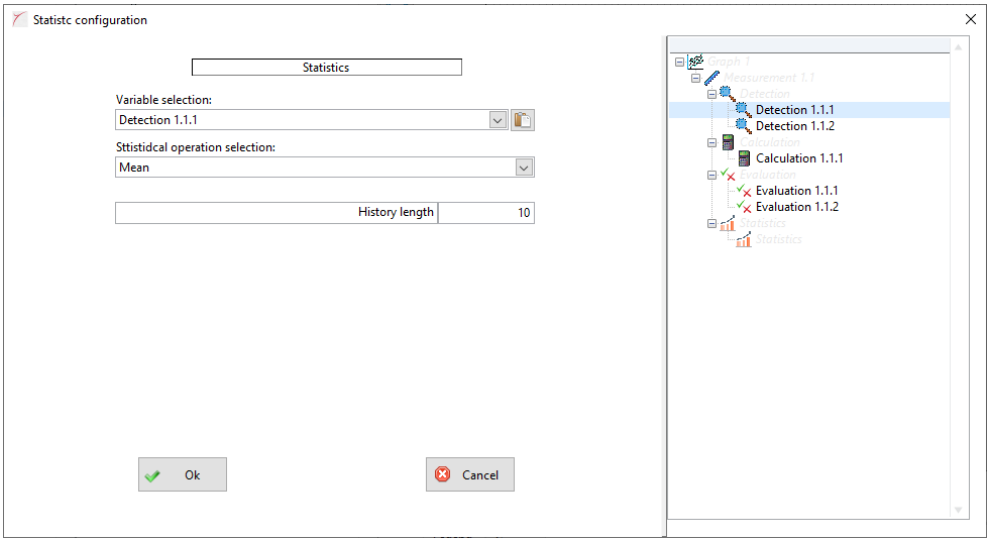
The evaluation is the final element of the quality control, the one that provides the pass/fail output. An evaluation can be composed of two or three operands, each of them configurable as a constant value or a variable parameter (detection, calculation or other evaluation result) to be chosen from the tree menu on the right and copied through the  button. The formula written on the lower section of the interface provides the preview of the comparison that will give the pass/fail result.

It is also possible to deactivate an evaluation through the *Activate evaluation* checkbox located below the name of the element.

An evaluation can be also configured with percentage tolerances: in this case, as exemplified in the picture below, the user has to set the *variable to compare* (which is the parameter that is going to be evaluated to provide the pass/fail output), the *reference variable* (which is the target value that the *variable to compare* should meet, with the appropriate tolerance window) and the upper and lower tolerance, with absolute values. The preview formula is available also in this case.

On the right of the interface, the results of the latest execution of the selected evaluation are displayed.

4.5.6. Statistics



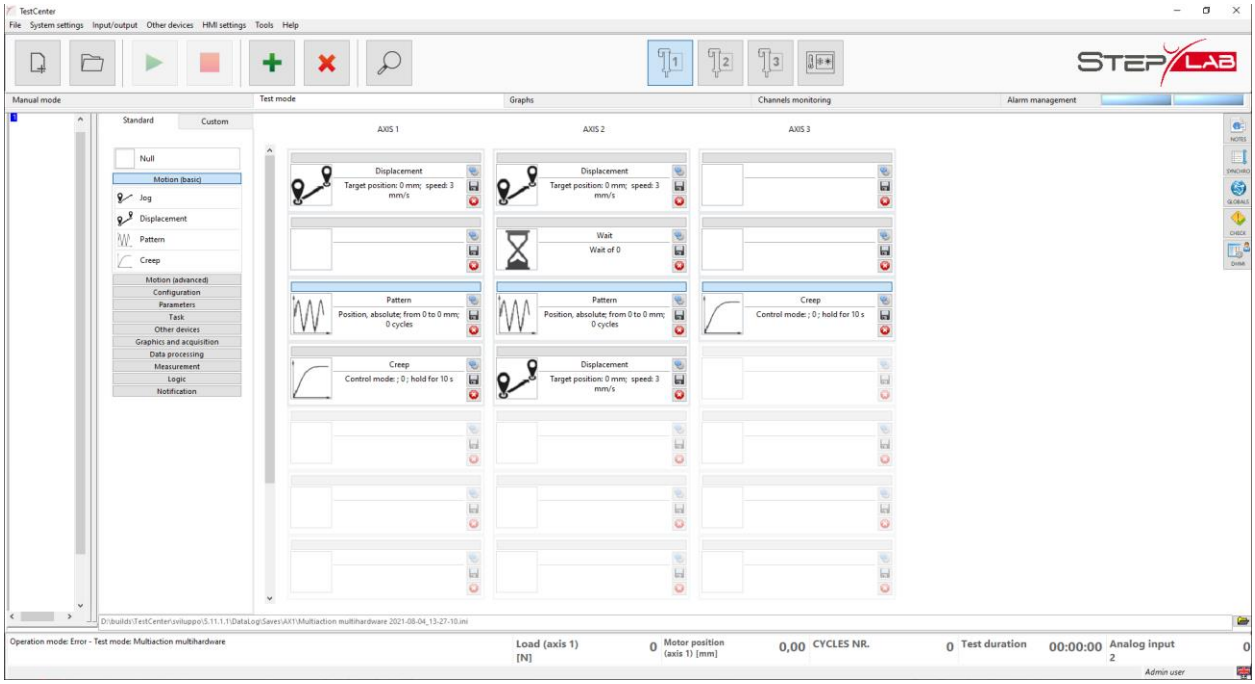
The last functionality of *multiaction with measurements* is statistics computation: the results of the various detections, calculations and evaluations can be analysed to provide some statistic information about their distribution:

- Mean value of the results
- Variance of the results
- Standard deviation of the results
- Maximum value of the results
- Minimum value of the results
- Percentage of the results that correspond to a certain value, provided a tolerance band

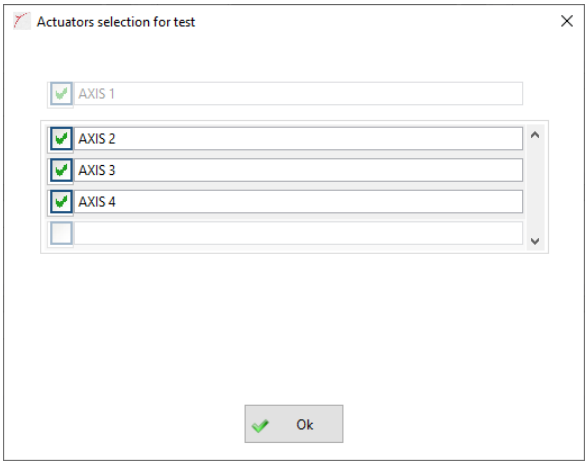
It is also possible to specify the history length of the statistic, which is the number of samples that are considered to process the results.

4.6. Multiaction multihardware test

The *Multiaction multihardware* test is a protocol that can be used in applications which include multiple actuators, in order to run tests that synchronize the various actuators together.



Starting with the software version 7.0, in the event that the application is configured for more than two actuators, when creating a new multihardware multi-action test, the user is given the option of choosing which actuators are involved in the test:

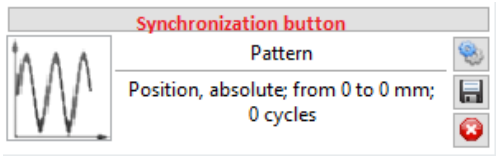


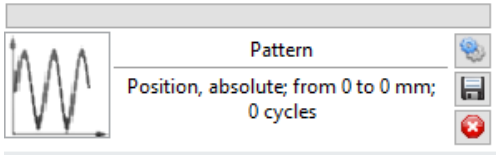
It should be noted that it is only possible to include actuators with an index later than the one used in the test: for example, if the multihardware test is created from actuator number 2, actuator number 1 will not be usable within that test. The actuator with the lowest index, i.e. the one from which the test is configured, is defined as the “master” of the test.

Finally, a test file that was configured for a certain set of actuators can only be reused with those same actuators.

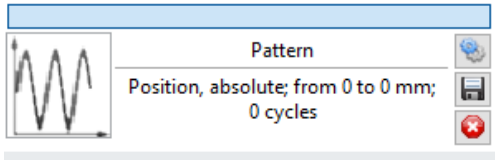
The interface provides the same actions menu of the standard *Multiaction* test on the left, and the top-down sequences of the various actuators on the main section of the interface. As in the single-actuator multiaction tree, the sequences can be composed by dragging and dropping the actions into the desired position.

When two or more actions of various actuators have to start at the same moment, they need to be synchronized: when synchronization is active, all the synchronized actions start at the same time as soon as all the actuators are ready to execute them (in other words, the synchronized actions start when all the actuators finish everything that comes before in their respective test sequences). In order to synchronize actions with one another, the rectangular button on the top of the actions preview has to be activated:





In this case, the *Pattern* action is not synchronized



In this case, the *Pattern* action is synchronized

Actions can be synchronized only with actions of other actuators that lay on the same row of the multi-hardware test sequence; this is the reason why it is possible to drag and drop *Null elements* from the top of the actions list into the sequence, in order to place at the same level the actions to be synchronized.

When two *Pattern* actions are synchronised, the controller sets a link between their frequencies in such a way that, in the event of running changes to the frequency of the master actuator, the ratio between the two frequencies remains the same: for example, if actuator number 1 has an initial frequency of 1 Hz and actuator number 2 of 3 Hz, and during the test the frequency of the former is increased to 2 Hz, the controller automatically increases that of the latter to 6 Hz.

In order to switch the synchronization buttons of the actions, the 

button has to be active; when the synchronization configuration is completed, this button has to be switched back to OFF to "freeze" the synchronizations. After this, the software will keep the synchronized actions at the same level automatically by adding or deleting *Null elements* if the sequence is modified.

For each action, three buttons are available:

-  : the configuration interface of the action is opened (refer to the detailed descriptions of the various actions available in the *Multiaction test* chapter of this user guide);
-  : the selected action is saved as a template action in an xml-format file;
-  : the selected action is eliminated from the sequence.

On the right of the *multiaction multihardware* interface there are some buttons which work the same as in the standard *Multiaction* environment (see paragraph 4.4.48), and the "synchro" button that was described before in this section.

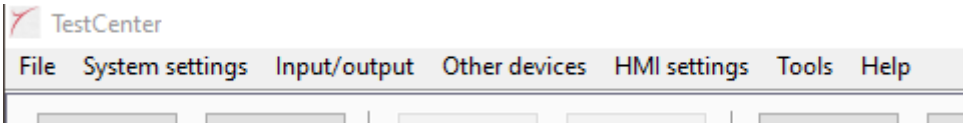
4.7. Quitting a test

A test might finish naturally if the target load/position/cycle count/etc. is reached, or abnormally if a safety limit is exceeded or an alarm happens: it is easy to notice when a test finishes because the buttons on the upper section of the main interface change their enable/disable state, and on the left side of the lower bar (see paragraph 1) the *operation mode* switches from *Auto* to *Standby* or *Error*; in this text field a message with a short description of the procedure end cause can also be found. Below there is a list of the possible exit conditions:

- Target load reached, target position reached: the machine completed the procedure matching the load or the position set by the user for the operation;
- Fracture detected: the break threshold was exceeded, thus the machine stopped automatically because the specimen caved in; if this exit condition intervenes without any actual fracture of the specimen, the test should be repeated or restarted modifying the break threshold value;
- Timeout elapsed; a maximum duration was set for the procedure, and that time elapsed without any other exit condition to happen;
- Target cycle number reached: for dynamic tests, it means that all the required cycles were completed;
- Operation completed: this is a general definition to identify a natural conclusion for the procedure;
- Al.1 (driver error or open doors): this is the main alarm condition, which happens if the emergency button is pressed, if an error state happens for the driver (for example, because the machine was asked for out of reach performances), or if safety doors are opened during an automatic procedure (except for null force maintenance, to which different safety algorithms are dedicated because the main application for this feature – the clamping of the specimen – needs to be done with open doors);
- Al.2a (load/position traction limit): an upper load or position limit was exceeded during the procedure; this exit condition also causes the dedicated alarm to raise only if the corresponding standard machine limit was in use;
- Al.2b (load/position compression limit): a lower load or position limit was exceeded during the procedure; this exit condition also causes the dedicated alarm to raise only if the corresponding standard machine limit was in use;
- Stop from user: the user commanded the machine to stop through the dedicated button on the upper section;
- Exit condition nr X satisfied: this is a general definition that applies when a custom exit condition is detected;
- Undefined exit code: the detected exit condition does not match any of the ones that are automatically recognized.

5. Menu bar

Above the upper bar with the seven main buttons there is a menu with the options displayed below:



Following the descriptions for each functionality related to the different options:

5.1. File *submenu*

File – New test: opens the dialog box through which a new test environment can be loaded into the program; this command is the same as the *New* button on the upper bar.

File – Load test: opens a file explorer window through which the user can select a test (i.e. a configuration file) to load into the dedicated environment, in order to continue a test that was previously created or to load a test template (see the next point): in the first case it is not possible to change the test parameters if the test has already ran (except for the ones that are allowed for run-time modifications), whereas in the second a new file is generated starting from the settings included in the template, but with full editing possibilities; this command is the same as the *Load* button on the upper bar.

File – Save test: gives the software a command to save the current test configuration file (please note that the savings are generally automatic, as soon as some configuration parameter is changed).

File – Rename test: this functionality allows the user to programmatically move and rename the test configuration file and, if there is any and the user wants to do it, also the results file already created for the specimens included in the configuration file; in fact, the software is configured to save each setting and results in file generated automatically with default names: this is comfortable in terms of safety to avoid data being lost, but having lots of files with similar names in the same folder may cause confusion; however, files must not be renamed through the Windows functionality because this would cause the link between the configuration and the results file to be broken, resulting in difficulties to get the correct files in the post-processing environments and even separation of the results of a single test into different files.

Note: this functionality does not create copies of the files, but renames and moves the original ones.

File – Save test as template: saves the current test configuration as a template to be used in the future through the *Load* functionality in order to create new pre-compiled tests without editing the source file: in fact, if the user loads a conventional test file then the original file is committed to memory and modified in case test settings are changed, whereas if a template file is loaded then a new test and a new configuration file are created starting from the settings included in the template instead of the default ones; in this way the original template file is preserved to be re-used whenever it is necessary; the name for the template file can be chosen without restrictions, the software automatically defines a file created through this functionality as a template.

File – Write results on disk: this is a command that only makes sense to give during the execution of a test, and is used to force the writing to the TDMS file of those results that are in the program memory waiting to be saved: in fact, the TestCenter, in order to optimise performance when accessing the TDMS file, does not save the data instantaneously but collects them in memory and writes them to the file periodically; this command overlaps the normal periodic writing and causes the software to record in the file all the data it has in memory at that moment.

5.2. System settings *submenu*

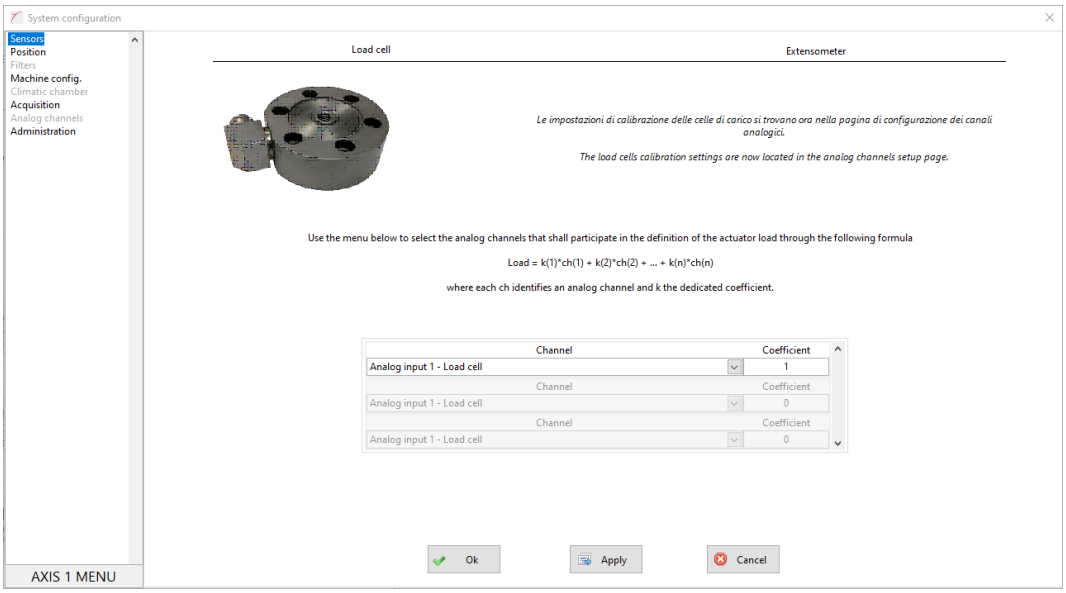
5.2.1. Configuration

The general system configuration window is displayed through this command. This interface is composed of a list (on the left) to navigate between the various sections, and the tab on the right to read and set all the parameters available.

Three buttons are always visible on the bottom of the interface, named *Ok*, *Apply* and *Cancel*: the first one closes the configuration interface saving all the settings; the second saves the settings of the current page without closing the interface; the third closes the interface discarding all the changes. It is not possible to save configuration changes while the machine is in *Auto* or *Manual* operation mode.

Below the descriptions of every section of the configuration interface:

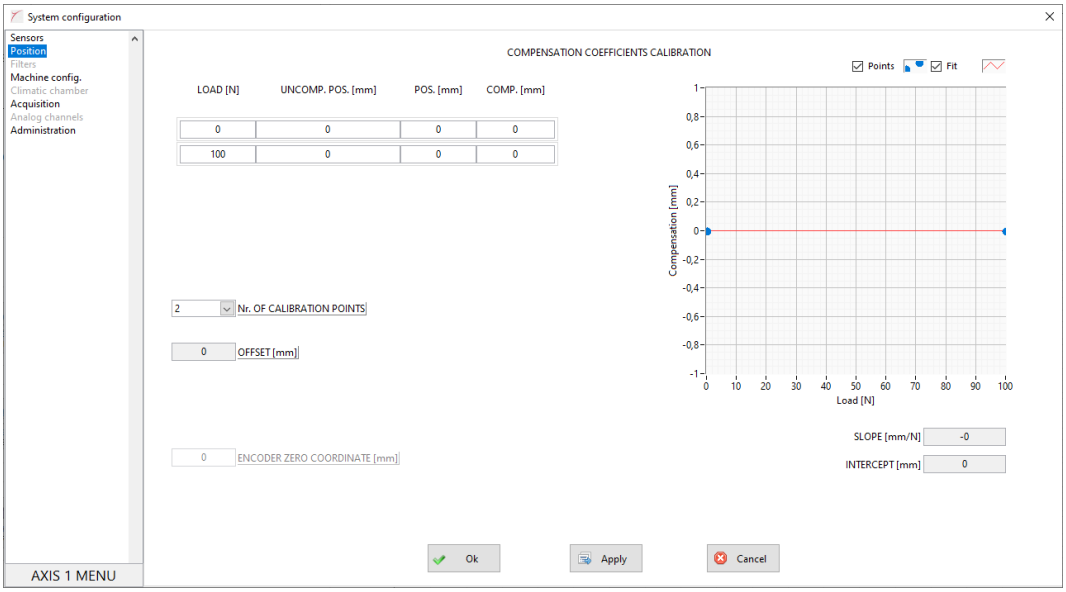
<u>Sensors:</u>



This section includes the configuration of the load and strain channel: the load channel can be obtained by a linear combination of various analog input channels, in case multiple load cells are installed and used at the same time; instead, if only one load cell is in use (as in the majority of the cases), everything that is needed is to choose the load cell channel in the first element of the list (as in the picture above) and set the coefficient to 1 or –1 to have the appropriate load sign rule.

In order to delete an element from the channels list, the user has to right-click on it and select *Delete from array*.

Position:

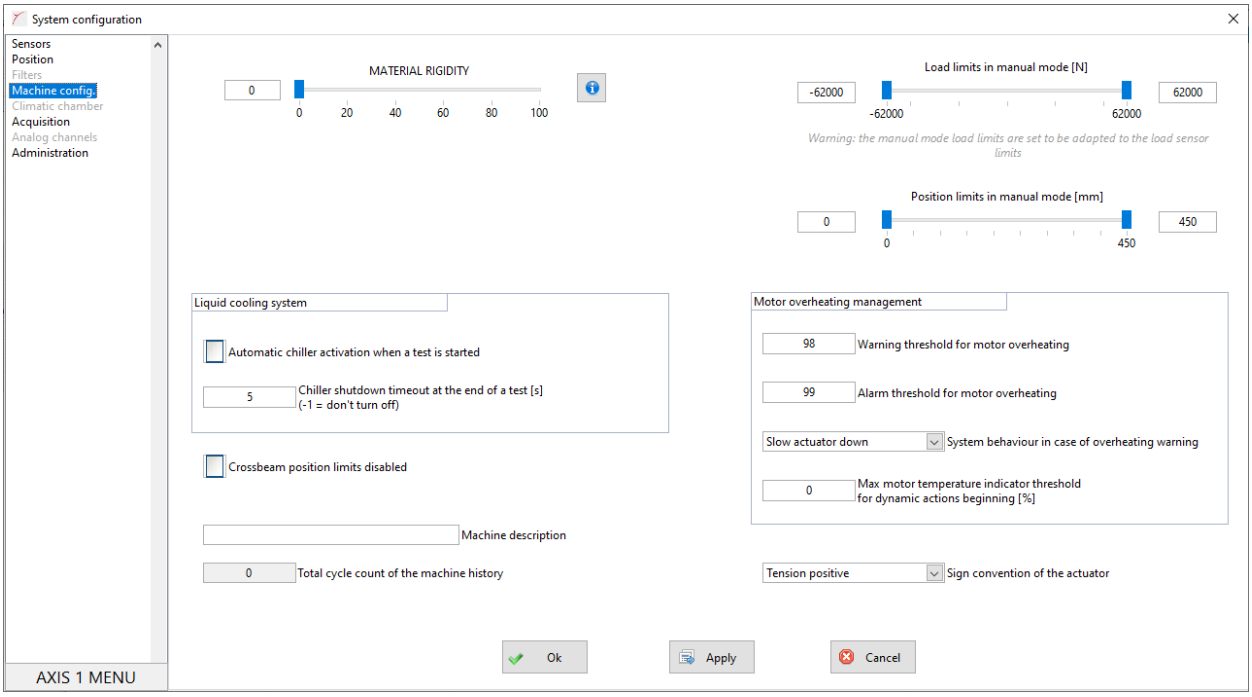


From here it is possible to introduce the points that the program uses to calculate the position compensation coefficient: for each given load there must be the rough position, which can be read from the lower bar of the interface, and the actual position, that must be determined through an external device during the compensation calibration procedure, using an instrument capable of a resolution less than 0,01 mm. From 2 to 5 compensation points can be used, and on the right side of the page there are a preview graph and two read-only numeric indicators with the angular coefficient and the intercept calculated through the inserted points, that are the data sent to the controller once the operation is confirmed through the *Apply* or *Ok* button.

The compensation parameters can be visualized from any user, but only *Advanced users* or superior profiles can edit them.

Finally, on the bottom-left side of the interface there is a numeric control displaying the position offset, which is the difference between the absolute machine position and the user coordinate system position (typically configured through the *reset position* operation).

Machine configuration:



This window provides some controls related to the machine behaviour:

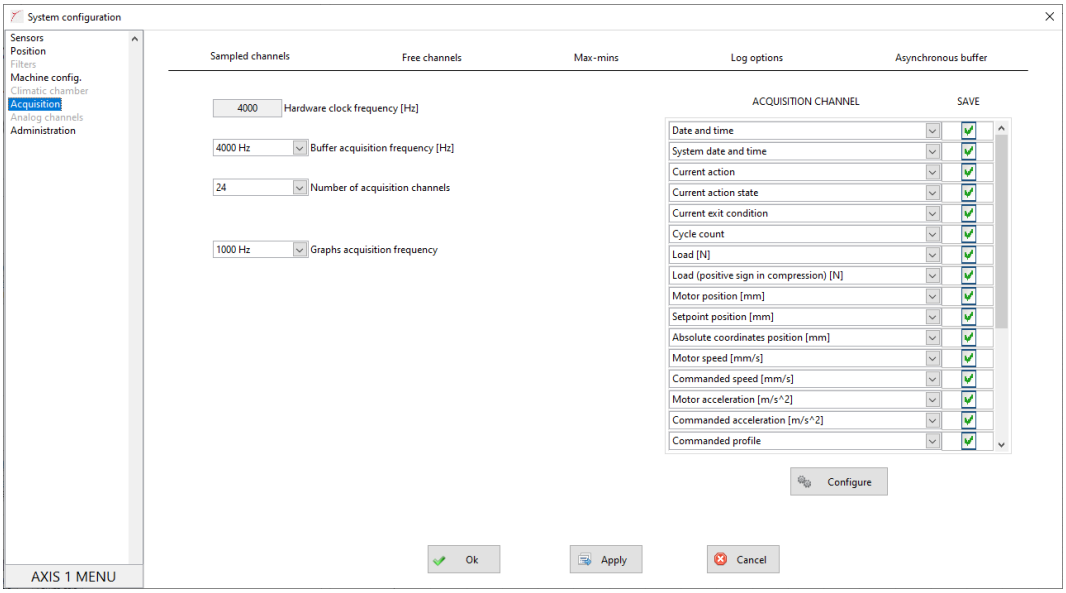
- *MATERIAL RIGIDITY* slide bar: this is a basic setting that operates on the controller to configure the speed and some PID parameters of manual procedures (hold F=0, creep), creep tests and fatigue test preliminary calibration; the more rigid is the specimen, the higher is the value that has to be set to the cursor; some examples are:
    - 0 for rubber or foam
    - Approximately 30-35 for unfilled plastics
    - Approximately 65-70 for composite materials
    - The maximum values (95-100) for metallic materials
- Load and position *limits in manual mode* (cursors + numeric values): it is possible to set a load and stroke range for the manual handling mode that can be stricter than the standard range of the machine;
- *Liquid cooling system* options: if the machine is equipped with a liquid cooling system, it is possible to choose whether to activate it automatically at every test start and whether to turn it off automatically after a certain time when a test finishes;
- *Crossbeam position limits disabled*: only if the machine is equipped with an electric crossbeam capable of acquiring its position value, it is possible to disable the position limits to execute particular setup operations;
- *Motor overheating management* parameters: these are the settings that control how the system behaves in case high performances are required to the motor and for this reason its internal temperature reaches a critical condition;

- *Total cycle count of the machine history*: this indicator displays how many cycles have been executed by the machine;
- *Sign convention of the actuator*: this setting allows to configure the polarity or sign convention of the actuator, i.e. the definition of which direction of motion is associated with positive/increasing position values and which is associated with negative/decreasing values; for linear actuators the available options are "tension positive" (the position value increases if the actuator moves in a direction that would correspond to a tension of a hypothetical specimen positioned between the grips) and "compression positive" (vice versa) in order to manually change the sign convention it is necessary to switch off the motor using the appropriate ON/OFF button on the manual control page; furthermore, at the end of the sign convention change procedure the software proposes a list of the available analogue channels asking the user which are those for which the calibration must be automatically inverted it is compulsory to do this for all load cell channels that detect the load in the direction of actuator movement, because their sign convention (positive tension or positive compression, in this case in terms of load values) must always correspond to that used for actuator movement.

Acquisition:

Minimum user profile required to access this page: Advanced user.

From here it is possible to configure the channels to be acquired from the field.

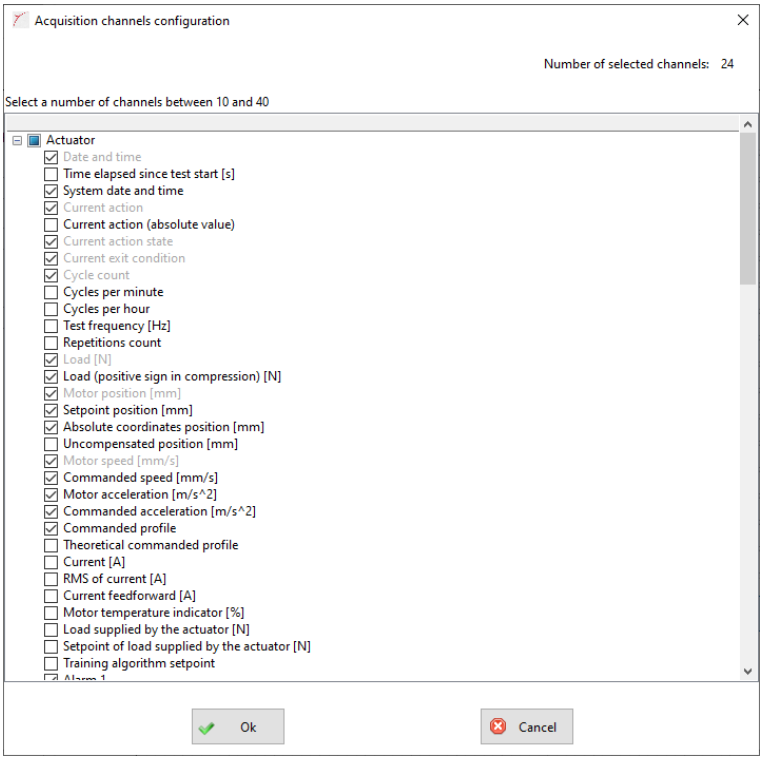


On the first page of this section it is possible to configure the buffer acquisition frequency, its number of channels and the channels to be acquired from the field. Also, starting from software version 6.16, this interface includes a control to set the sampling frequency of the data that are sent to real-time

graphs: this allows to choose between maximising the quantity of data or reducing the use of RAM memory while operating.

For each row of the channels list there is a checkbox to state if the channel has to be saved into TDMS files or not; starting with version 6.24 of the programme, the configuration of channels to be saved is universal and no longer independent for each type of test.

Starting with version 7.0 of the program, a new interface for configuring acquisition channels is available, accessible via the “Configure” button immediately below the channel list:

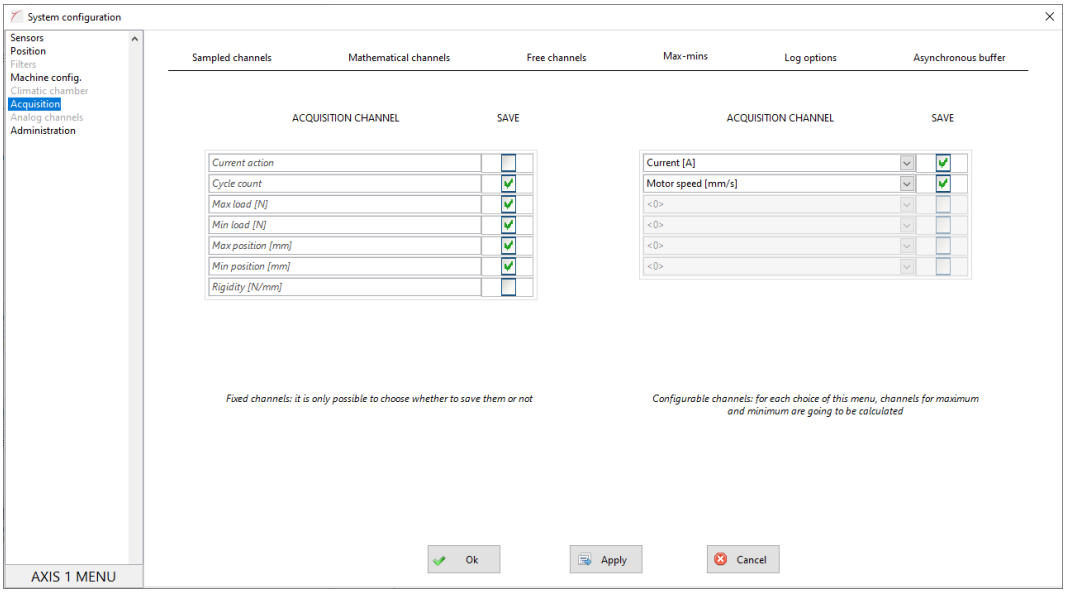


The channels available for acquisition are grouped into categories according to the subsystem to which they refer (actuator, analogue and digital input and output channels, other subsystems, mathematical channels...); the compulsory ones are disabled, meaning that they cannot be deselected and must in any case be included in the list of channels to be acquired.); the compulsory ones are disabled, meaning that they cannot be deselected and must in any case be included in the list of channels to be acquired; furthermore, switching the selection of a “parent” element (i.e. of a subsystem) enables or disables all the channels that are its children in the tree; finally, immediately above the list of channels, a brief text notifies the user what the minimum and maximum limits are with regard to the number of channels that can be selected.

At present, there are these requirements when configuring this setup:

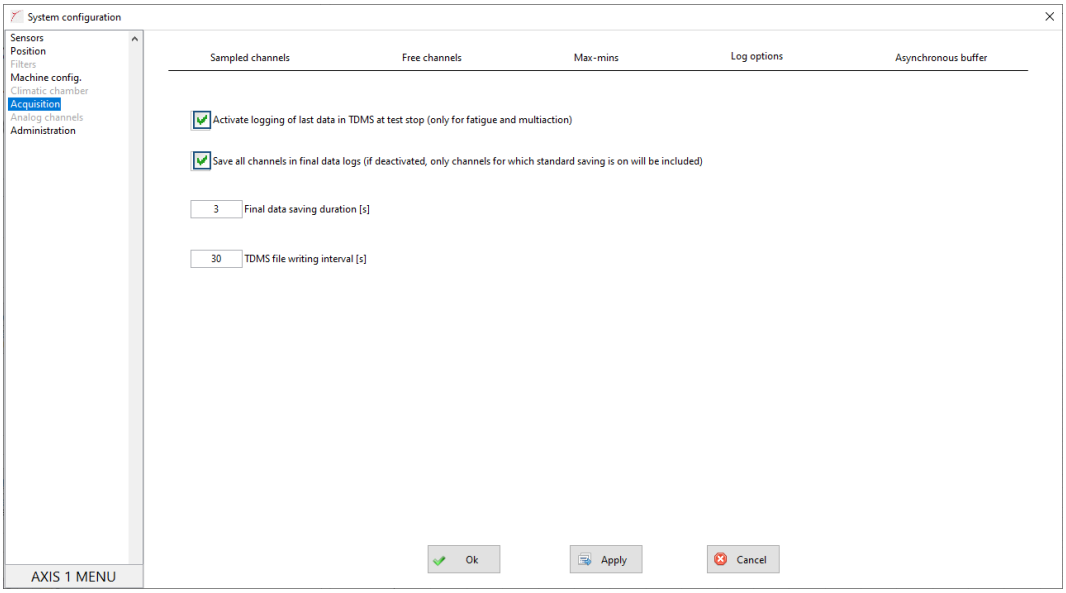
- The following channels must always be included: *Date and time*, *Cycle count*, *Current row index*, *Load*, *Motor position*, *Motor speed*, *Current action state*;
- If a *Multiaction with measurement* test has to be executed, also the channel named *Active measurements state* is needed;

- If there is a *Strain*, it needs to be placed after the *Motor position*; if there is a *Position setpoint*, it needs to be placed after the *Strain* (if present) or after the *Motor position* (if there is no *Strain*).



The *Max-min* tab of the *Acquisition* interface is dedicated to the maximums&minimums buffer, which is the instrument used by the software to acquire maximum and minimum values for each cycles in fatigue tests. This buffer is composed of two sections:

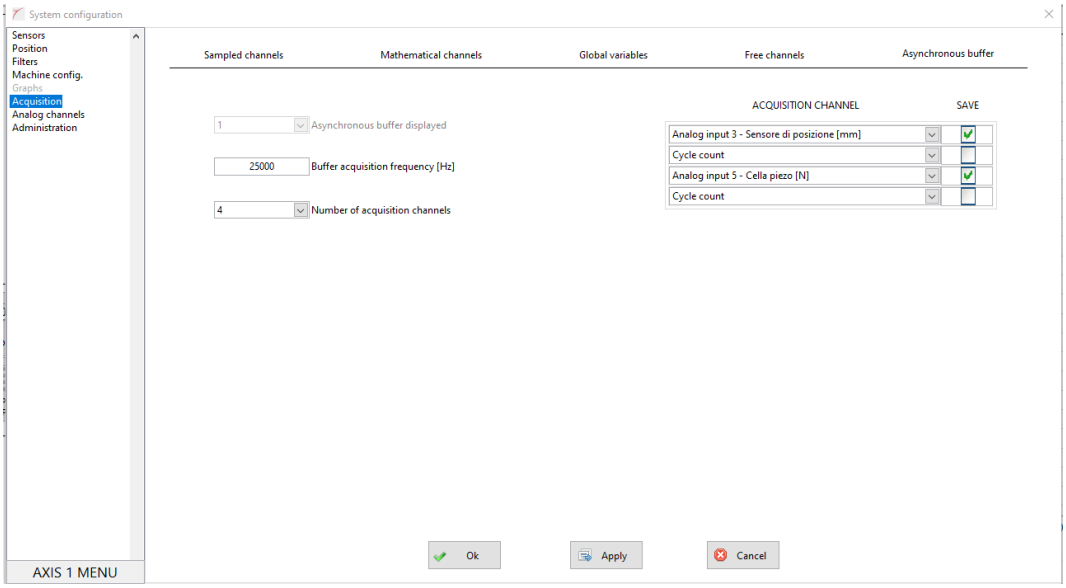
- Some standard channels, displayed on the left of the page: they can't be modified, it is only possible to choose whether to save them or not through the checkboxes;
- Some configurable channels, displayed on the right of the page: for each quantity set by the user in this list, the software generates the maximum and minimum values, providing therefore two channels.



The *Log options* are the software options related to data saving:

- Activate logging of last data in TDMS at test stop (only for fatigue and multiaction): if this setting is active, the software automatically stores a certain quantity of data in the test results file every time the machine stops;
- Save all channels in final data logs (if deactivated, only channels for which standard saving is on will be included): if this and the previous setting are active, the last data of the test that are saved automatically include all the channels configured for the acquisition buffer, instead of only the channels that has to be saved in the standard saving process;
- Final data saving duration [s]: this setting is only valid for multiaction tests and establishes the length of the memory to be allocated for the data that have to be saved automatically when a test is stopped;
- TDMS file writing interval [s]: during a test procedure, the software records data in the RAM memory and saves them periodically in the results TDMS file; this parameter defines this periodicity: a lower value favours the immediacy of the data update, while a higher value helps to make the TDMS file more manageable in the post-processing phase (for the same size and number of groups, the fewer write operations that generated a file, the lighter this file is for the programme to manage).

The last tab of the *Acquisition* page is dedicated to the asynchronous, high-rate sampling buffer which is used in the *DataAcq* action:



In this interface there are:

- The sampling frequency configured for the asynchronous buffer;
- The number of channels of the asynchronous buffer;
- The structure of the asynchronous buffer in terms of channels; for each channel there is a checkbox to save it on disk when a *DataAcq* action is used.

Warning: in this software version, the number of channels and the channels composition are fixed: it is only possible to choose the sampling frequency and which channels to save.

Administration:

Minimum user profile required to access this page: Admin user.

This page includes advanced software configuration functionalities reserved to STEP Lab personnel.

5.2.2. PID parametrization

The PID configuration window is displayed through this command.

Caution: *the PID configuration must be restricted to highly qualified users since editing these parameters improperly may be cause of damages in the machine components.*

The *PID parametrization* option of the *System settings* section of the menu opens the interface dedicated to the manual tuning of the PID controller; the interface layout and the available parameters may vary according to the type of controller installed:

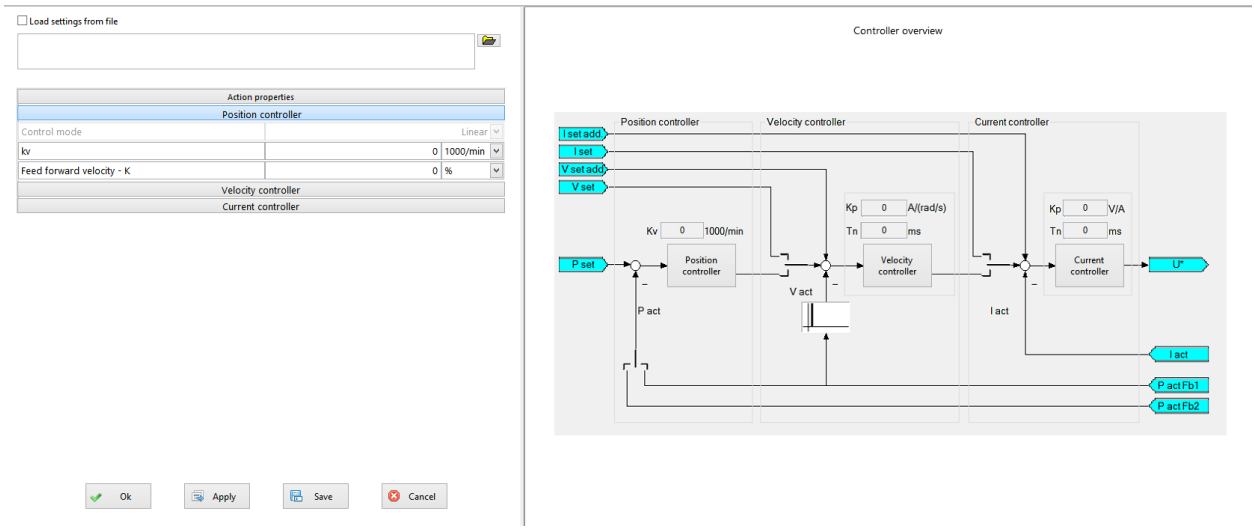


Figure 5.1 - First example of the PID configuration interface layout

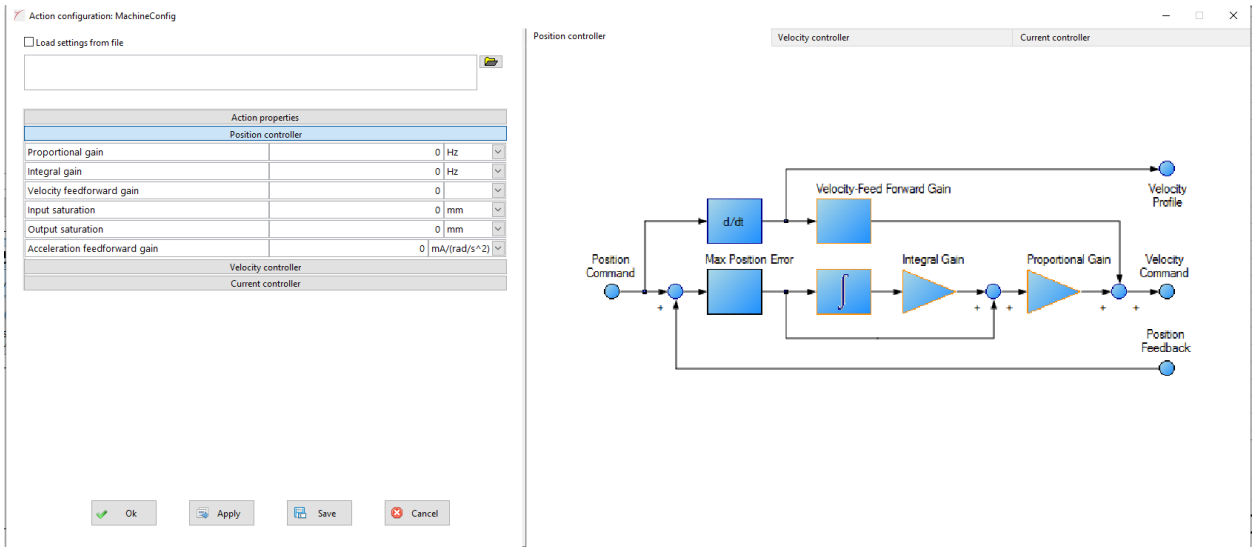
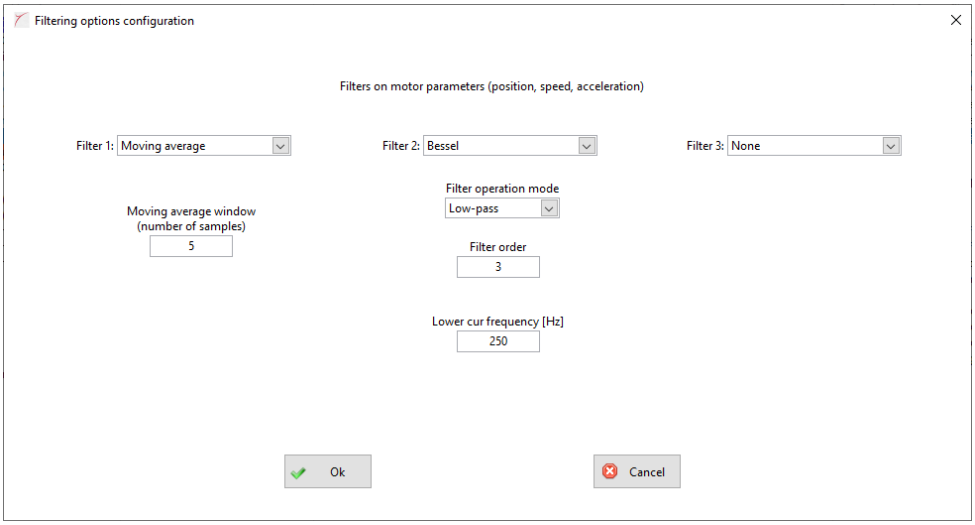


Figure 5.2 - Second example of the PID configuration interface layout

The interface is the same that is loaded for the *MachineConfig* actions, so a more detailed descriptions can be read on paragraphs 4.4.11 and 4.4.12; the only variations with respect to it are the *Apply* and *Save* button: the first is needed to communicate the parameters to the controller, the second can be used to save the current configuration as a reference to be loaded dynamically through the *load settings from file* feature, typically for an action included in a test.

5.2.3. Filtering

The following interface is displayed to configure up to three filters for the motor parameters (position, speed, acceleration):

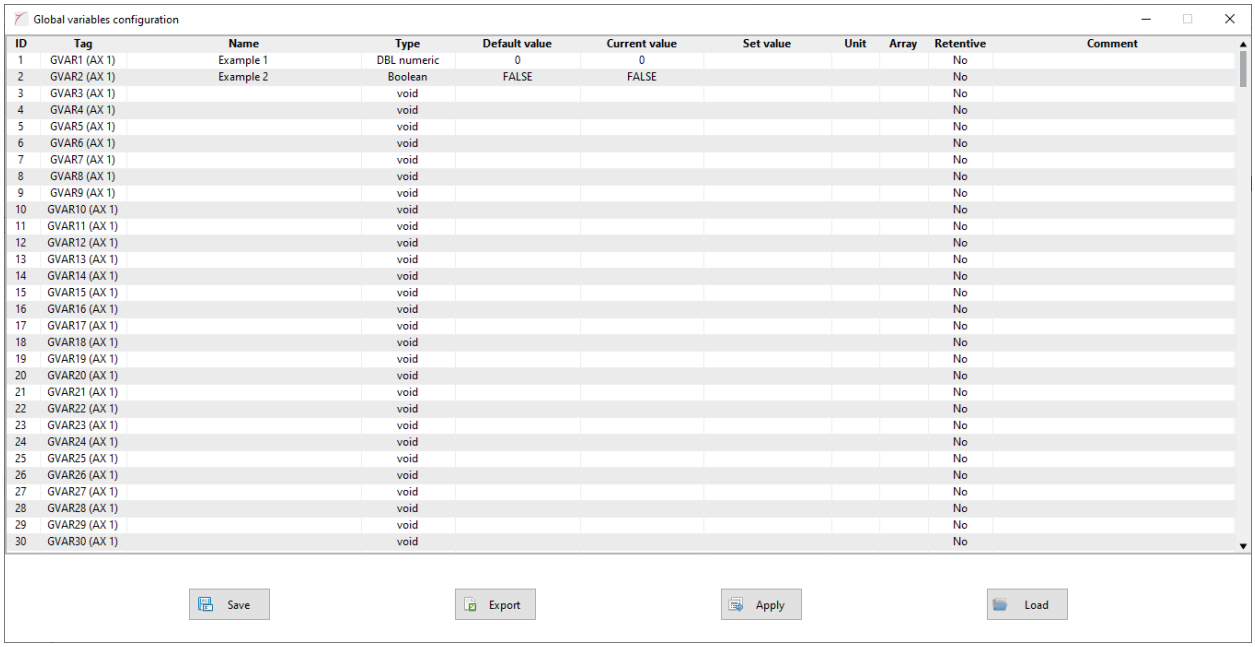


Each of the filters can be set as moving average, low-pass or Bessel filter; depending on the chosen filter type, the characteristics parameters are displayed.

5.2.4. Global variables

Global variables are auxiliary variables that the user has the possibility to customize (choose their type, name, default value, unit of measurement…) and use within the test sequence through actions such as *SetVariable*, *ChangeParameters*, *If*, etc.

The picture here below represents the configuration interface of the global variables:



ID	Tag	Name	Type	Default value	Current value	Set value	Unit	Array	Retentive	Comment
1	GVAR1 (AX 1)	Example 1	DBL numeric	0	0				No	
2	GVAR2 (AX 1)	Example 2	Boolean	FALSE	FALSE				No	
3	GVAR3 (AX 1)		void						No	
4	GVAR4 (AX 1)		void						No	
5	GVAR5 (AX 1)		void						No	
6	GVAR6 (AX 1)		void						No	
7	GVAR7 (AX 1)		void						No	
8	GVAR8 (AX 1)		void						No	
9	GVAR9 (AX 1)		void						No	
10	GVAR10 (AX 1)		void						No	
11	GVAR11 (AX 1)		void						No	
12	GVAR12 (AX 1)		void						No	
13	GVAR13 (AX 1)		void						No	
14	GVAR14 (AX 1)		void						No	
15	GVAR15 (AX 1)		void						No	
16	GVAR16 (AX 1)		void						No	
17	GVAR17 (AX 1)		void						No	
18	GVAR18 (AX 1)		void						No	
19	GVAR19 (AX 1)		void						No	
20	GVAR20 (AX 1)		void						No	
21	GVAR21 (AX 1)		void						No	
22	GVAR22 (AX 1)		void						No	
23	GVAR23 (AX 1)		void						No	
24	GVAR24 (AX 1)		void						No	
25	GVAR25 (AX 1)		void						No	
26	GVAR26 (AX 1)		void						No	
27	GVAR27 (AX 1)		void						No	
28	GVAR28 (AX 1)		void						No	
29	GVAR29 (AX 1)		void						No	
30	GVAR30 (AX 1)		void						No	

Each column of the table has a different meaning:

- ID: incremental identification code of the global variable (cannot be modified);
- Tag: internal name of the global variable (cannot be modified);
- Name: user-specified name of the variable (an interface dedicated to set it is opened by double-clicking on the cell of a variable);
- Type: user-specified data type of the variable (an interface dedicated to set it is opened by double-clicking on the cell of a variable); the data type can be chosen among the following options:
    - Void: no data type specified;
    - Boolean: true or false;
    - DBL numeric: floating point 64-bit numeric value;
    - String: text string limited to 64 characters;
    - Timestamp: date and time;
    - I32 numeric: signed 32-bit integer numeric value;
    - String (extended): text string that can exceed 64 characters, but for which the computation times are longer.
- Default value: this value depends on the data type and it is the default value for the global variable (an interface dedicated to set it is opened by double-clicking on the cell of a variable);
- Current value: this value depends on the data type and it is the current value for the global variable: it is updated in real time as long as the interface stays open;
- Set value: if the user needs to change the value of the variable in real time, it is required to double-click on this cell, set the desired value through the interface that is shown and then transfer it into the field and into the *Current value* cell through the *Apply button*;

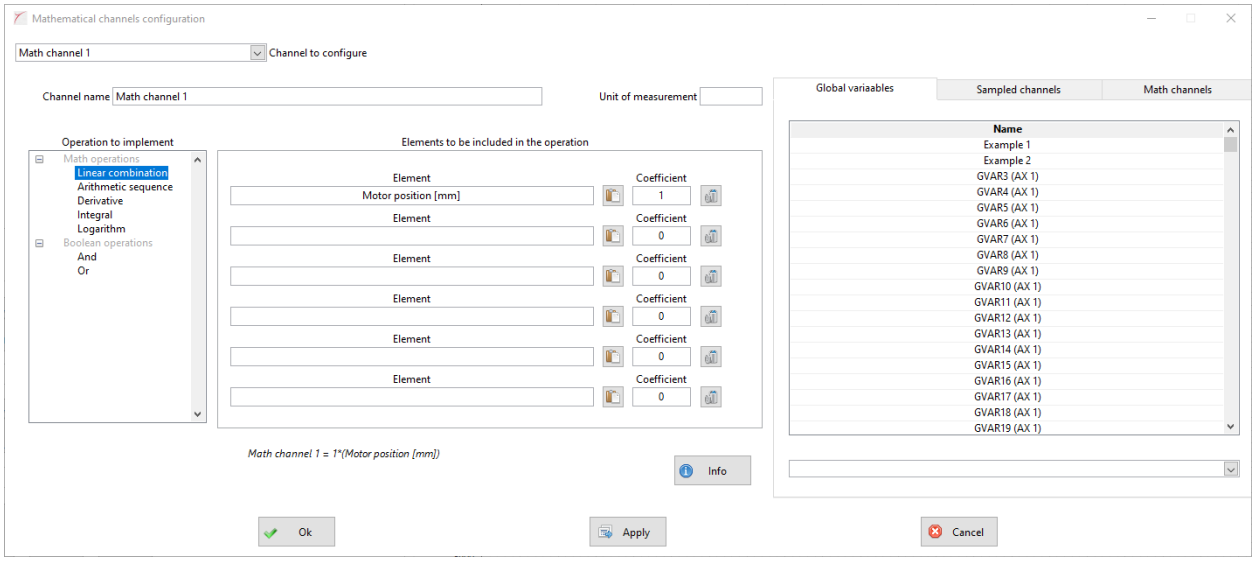
- Unit: permits to set an unit of measurement for the variable (an interface dedicated to set it is opened by double-clicking on the cell of a variable); please note that no automatic conversion is provided (for instance, if the user sets lbf as unit for a global variable that represents a load and then writes into it the current load value through a *SetVariable* action, the written value is still expressed in newton);
- Array: indicates whether the variable is part of an array or not;
- Retentive: global variables are reset when a new specimen is created within a test; if a variable is retentive, the current value is preserved; otherwise, the default value is restored;
- Comment: user-specified annotation for the variable (an interface dedicated to set it is opened by double-clicking on the cell of a variable);

On the bottom of the interface there are four buttons:

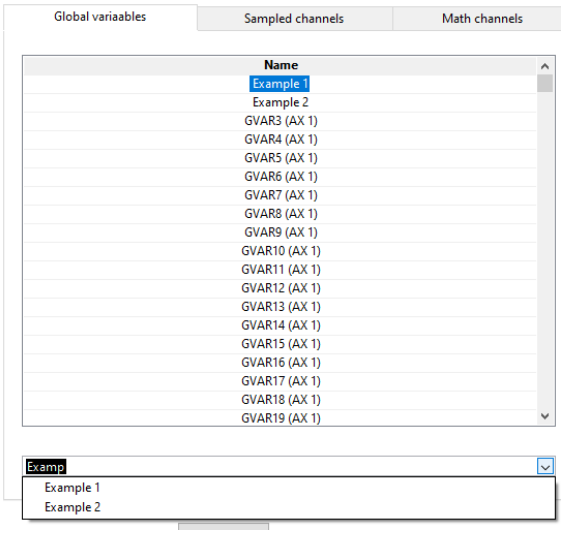
- Save: saves an xml-format file with the current global variables configuration;
- Export: exports the current global variables configuration in a csv-format file;
- Apply: it is needed to apply the pending *set values* to the global variables;
- Load: allows the user to load a different global variables configuration from an xml-format file generated through the *save* functionality or from a csv-format file generated through the *export* functionality.

5.2.5. Math channels

The following interface is displayed to configure some mathematical channels, which are additional channels that are calculated in real time from the other channels data:

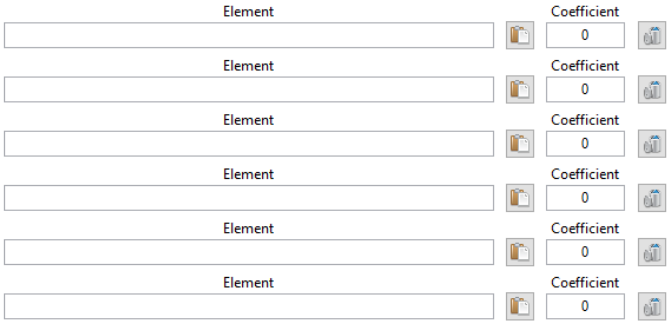


The user can type the channel name and the unit of measurement of each channel, then the *Operation to implement* tree presents multiple choices for the type of operation. For each case, the variables involved in the calculation can be selected from the lists available on the right (*Global variables*, *Sampled channels* and *Math channels* – it is possible to configure mathematical channels basing on other mathematical channels, too) and copied through the  button. Also, in the *Global variables* tab there is a combo box control to search for the desired variable through its name as exemplified in the picture below:

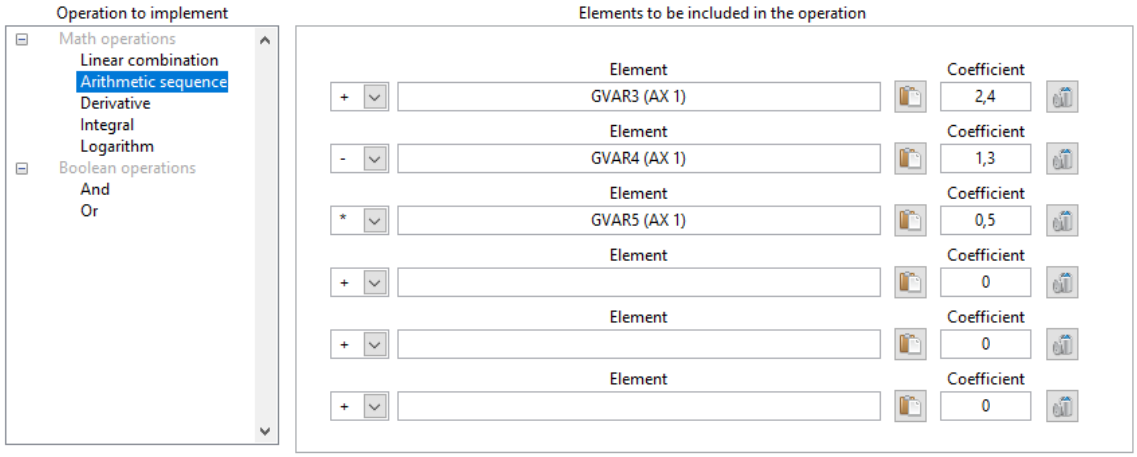


Here below a brief description is given for the various types of mathematical channels:

- Linear combination: some variables/channels can be chosen, each of those with a coefficient, and the result is:
  $Math\ channel = (Variable\ 1 * Coefficient\ 1) + \cdots + (Variable\ N * Coefficient\ N)$

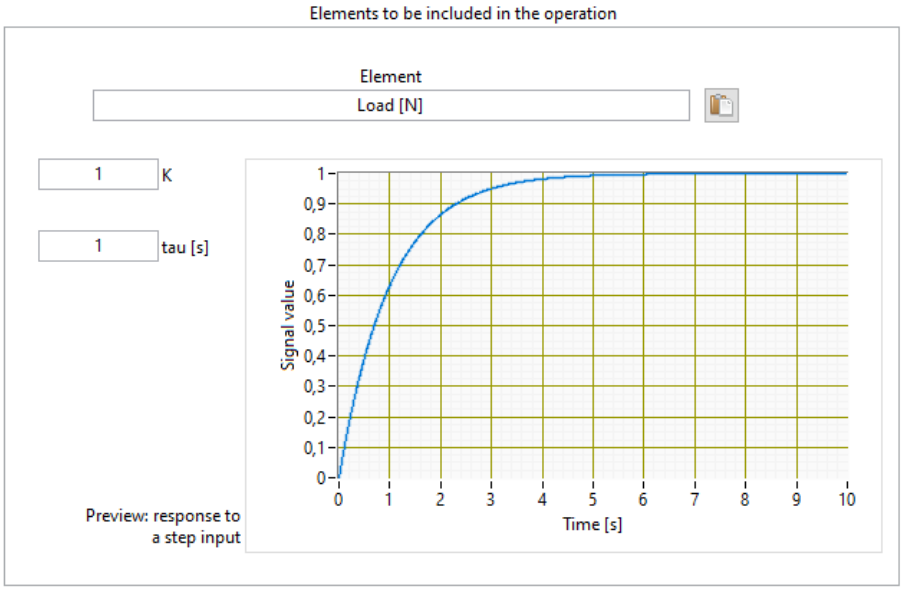


- Arithmetic sequence: a sequence of operations, channels and coefficients has to be configured; there is to remember that the order of the operations matches the one in the configuration interface regardless of the usual priority given to multiplication and division.



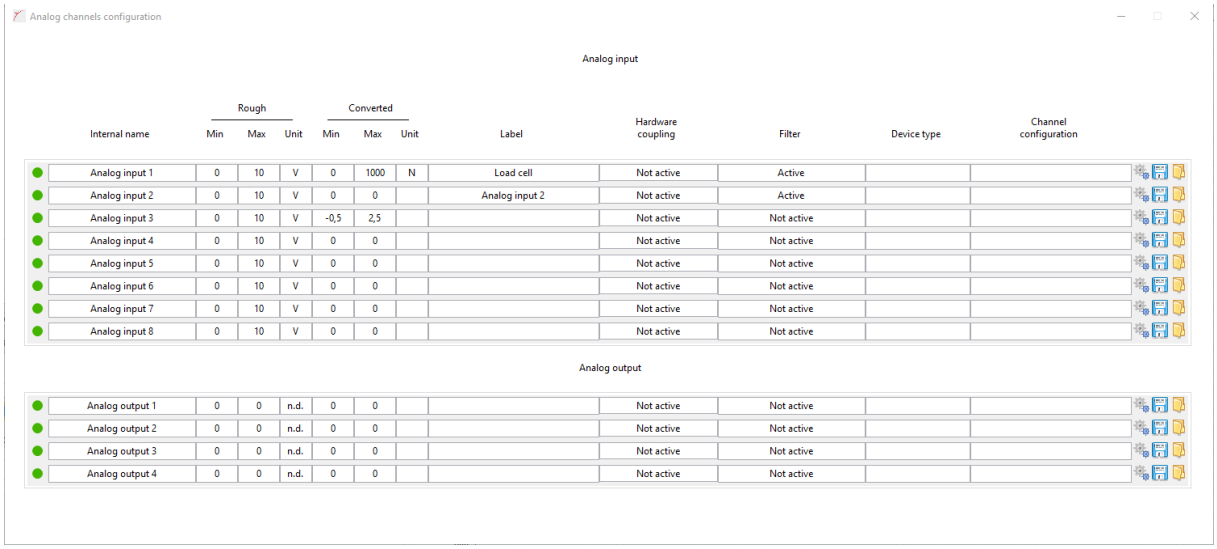
*Math channel 1 = 2,4(GVAR3 (AX 1)) - 1,3(GVAR4 (AX 1)) * 0,5(GVAR5 (AX 1))*

- Derivative: the mathematical channel computes the time derivative of the selected channel;
- Integral: the mathematical channel computes the time integral of the selected channel;
- Logarithm: the mathematical channels computes the natural logarithm of the selected channel;
- And: the mathematical channel executes a logical AND operation between all the selected channels and provides a value that is 0 or 1 according to the result;
- Or: the mathematical channel executes a logical OR operation between all the selected channels and provides a value that is 0 or 1 according to the result;
- First order system: the user has the possibility to reproduce the behaviour of a first-order system begins on an input channel and the values of the gain (K) and the time constant (tau, to be expressed in seconds):



5.3. Input/output *submenu*

Input/output – Analog channels: the following interface is displayed to summarize the properties of all the available analog inputs and outputs:

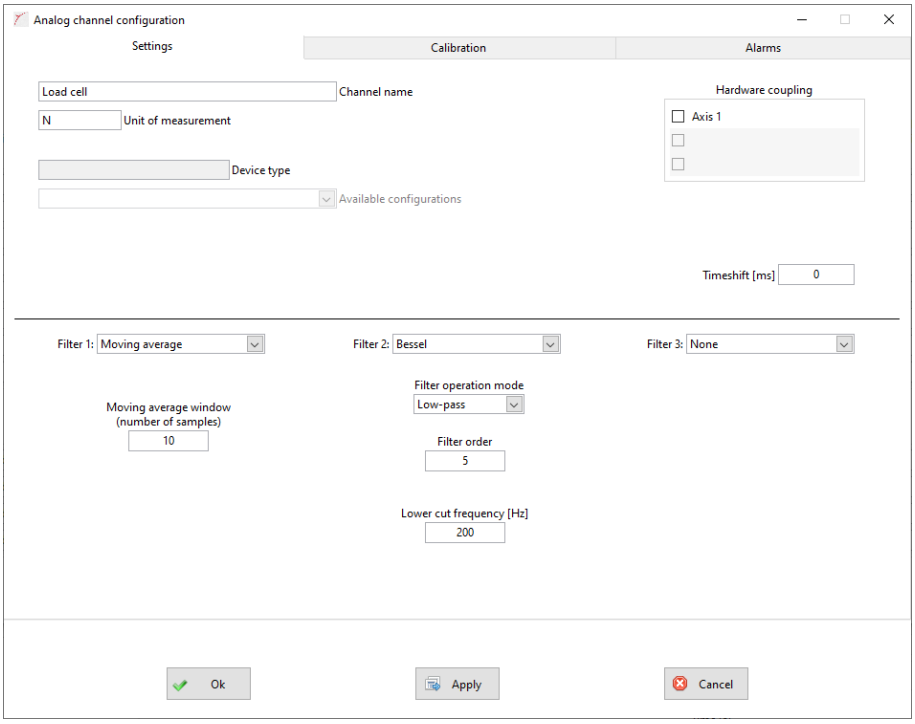


For each channel the software shows:

- Internal name: an internal channel name that can't be edited by the user;
- Rough – Max, min, unit: maximum and minimum rough value, with the unit of measurement, read from the field basing on the technical specification of the acquisition card;
- Converted – Max, min, unit: maximum and minimum values set for the calibration, and the unit of measurement configured in the dedicated interface (please note that it is not obligatory to have the maximum and minimum value configured in the calibration equal to the full-scale of the device);
- Label: a custom name for the channel that can be chosen by the user – not in this interface, but in the specific channel configuration interface;
- Hardware coupling: if the channel is coupled with an actuator, the actuator goes into alarm state as soon as the channel does;
- Filter: here it is summarized whether the channel has some active filters or not;
- Device type: the type of acquisition module to which this channel is associated;
- Channel configuration: depending on its device type, each channel can be configured choosing among some options, and this indicator displays which is the active configuration;
- : this button launches the configuration interface of the selected channel;
- : this button allows to save the entire channel configuration into an xml-format file;
- : this button allows to load the entire channel configuration from an xml-format file saved through the button.

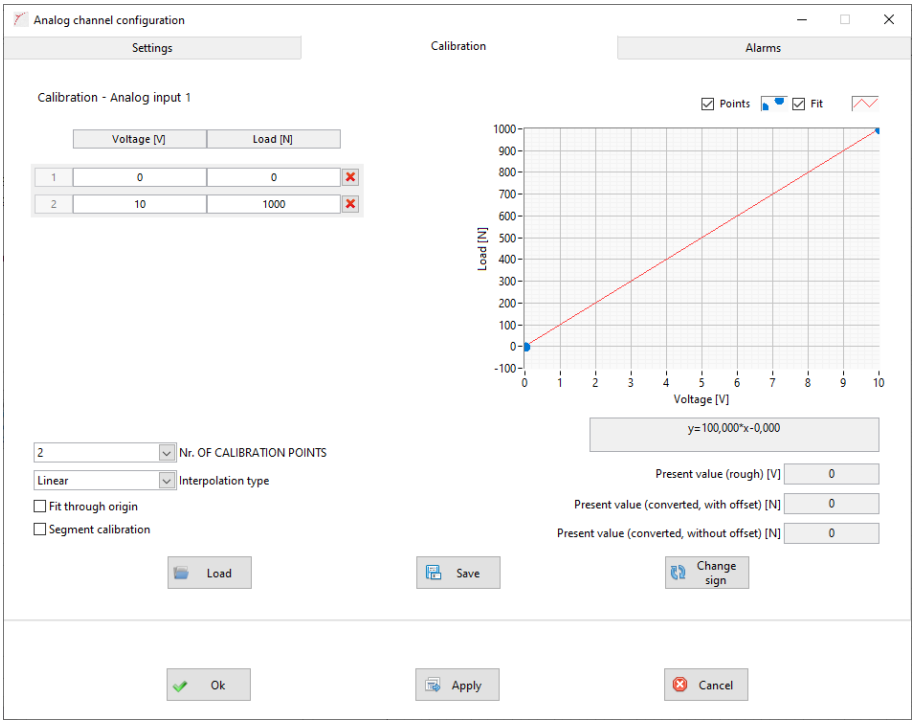
5.3.1. Analog channel configuration

If the button is pressed from the analog channels overview interface, the specific configuration interface of the selected channel is opened; it is composed of multiple pages, the first is dedicated to the channel general information and the filtering options:



Here it is possible to choose a name and a unit of measurement for the channel (in particular, the unit of measurement is important because the value written here has to be respected while introducing the calibration points; in other words, if the user writes N in this *Unit of measurement* text control then all the load values in the calibration points have to be expressed in newton). Additionally, up to three filters (that are applied sequentially by the controller) can be set up choosing among a moving average, a low-pass and a Bessel filter; depending on the choice operated for each filter, the dedicated parameters are displayed below the filter type. Starting from software version 6.8, the device type (model of the acquisition module) is also displayed and there is a drop-down menu (“Available configurations”) to change the active configuration of the analog channel, choosing among those that are available for its device type.

The second page of this window is dedicated to the calibration interface:



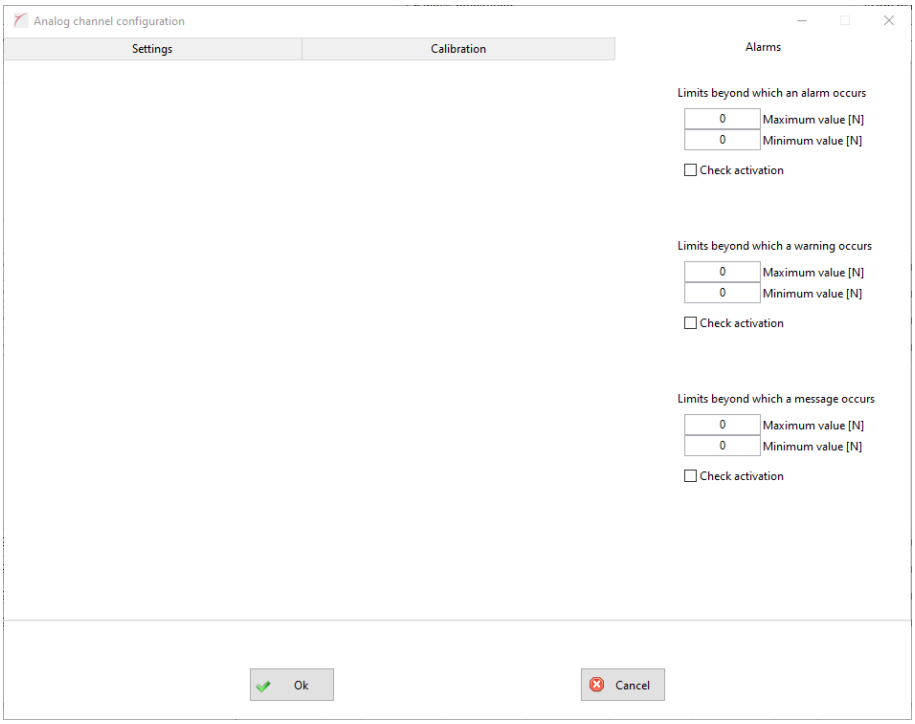
The calibration points – from 2 up to the maximum value allowed by the drop-down menu – can be inserted in the table, each of them providing the *rough value* (value that is sampled by the acquisition card in its native unit of measurement) and the *converted value* (value that is expressed in the unit of measurement chosen for the channel in the previous page); on the right of each calibration point inserted there is a button available to easily delete it. Additionally, two checkboxes are available to force the calibration interpolation through the origin and to use a segment calibration, if the selected *interpolation type* is *linear*.

On the right side of this interface, below the preview graph, the calibration formula and the following values read from the field for the channel:

- Rough: is the value of the electrical quantity, which the controller converts through the set calibration law to obtain the converted value;
- Converted without offset: is the final value of the channel in the unit of measurement chosen by the user;
- Converted with offset: is the converted channel value but without any offsets applied through reset procedures (manual or by micro-action in multiaction mode).

Finally, two buttons are available to load and save calibration files, so that it is not necessary to write the calibration tables manually every time they need to be changed, and another one named “Change sign” allows to instantly invert the sign of all calibration points if the user wants to change the rule according to which the channel values are positive or negative.

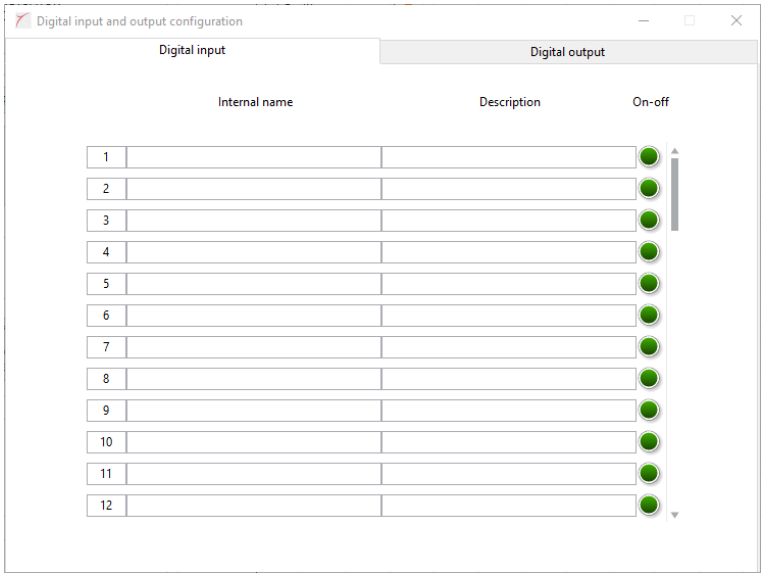
The final page of the analog channel interface is dedicated to the alarms configuration:

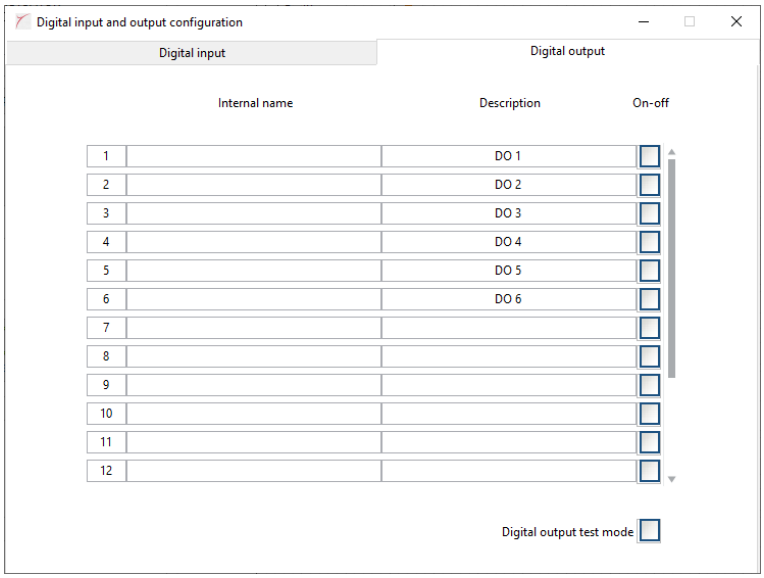


It is possible to choose a maximum and minimum value (to be expressed in the same unit of measurement that is written on the *Unit of measurement* control in the first page of this interface) that act as thresholds to trigger alarms and warnings.

Input/output – Digital channels: the digital input/output interface provides two pages, the first dedicated to the inputs and the second to the outputs:

5.3.2. Digital channels configuration



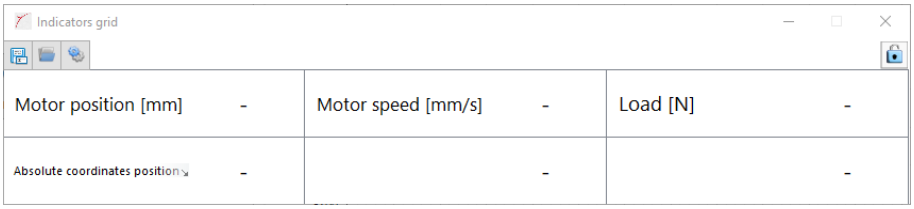


Both inputs and outputs have an internal name that can't be changed and a description that the user is free to customize; additionally, the inputs have an indicator of the current state (on/off), and the outputs have a checkbox that can also force the value to the field (typically for testing and diagnostics purposes) if the *Digital output test mode* option is active.

5.4. HMI settings *submenu*

HMI settings – Indicators (main interface): opens a window configured to select how many numeric indicators have to be displayed in the lower bar and which quantities have to be included inside them. The number of indicators to display is variable between 1 and 5.

HMI settings – Indicators (additional interface): opens a window through which some additional indicators can be displayed for the same quantities that can also be mapped in the built-in indicators at the bottom of the TestCenter software interface:



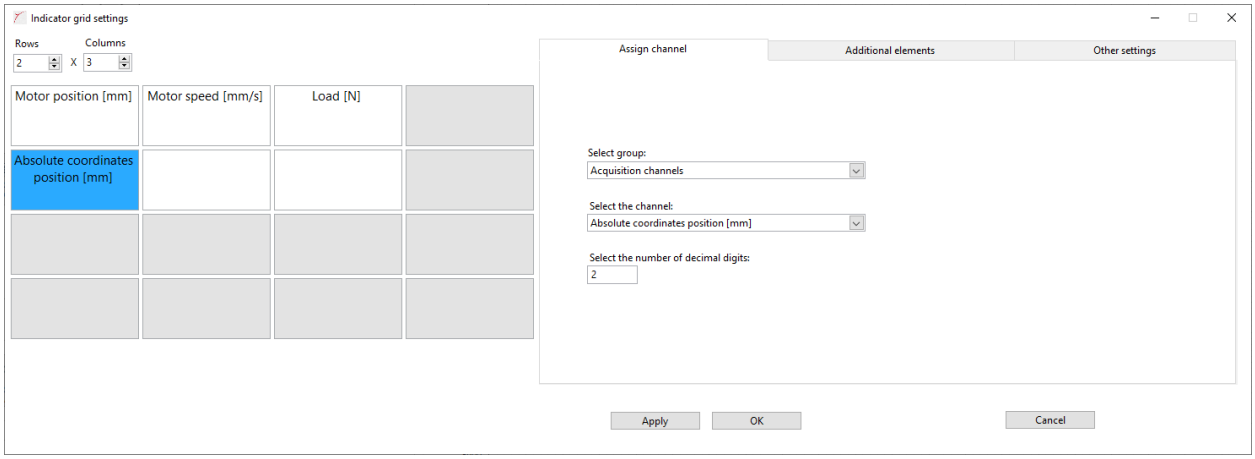
The buttons at the top have the following functions:

 If activated, it ensures that the additional indicator window always remains in the foreground;

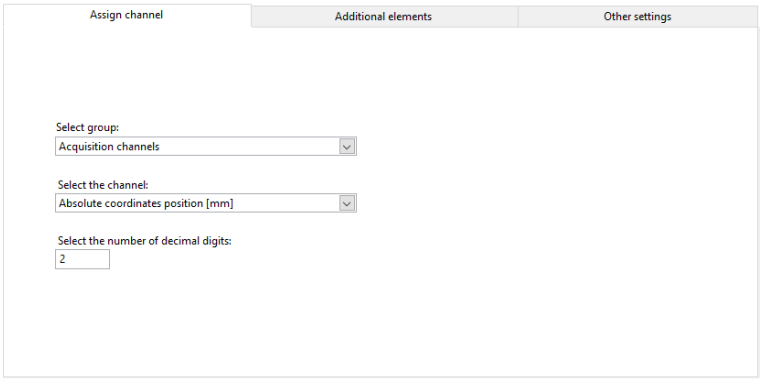
 Saves the current indicator configuration to a file, so that it can be retrieved later via the load function (note: the indicators built into the bottom of the TestCenter window are not included in the configuration saved via this command);

 Loads a previously saved indicator configuration via the save button, and applies it to the window (note: the indicators integrated at the bottom of the TestCenter window are not included in the configuration applied via this command);

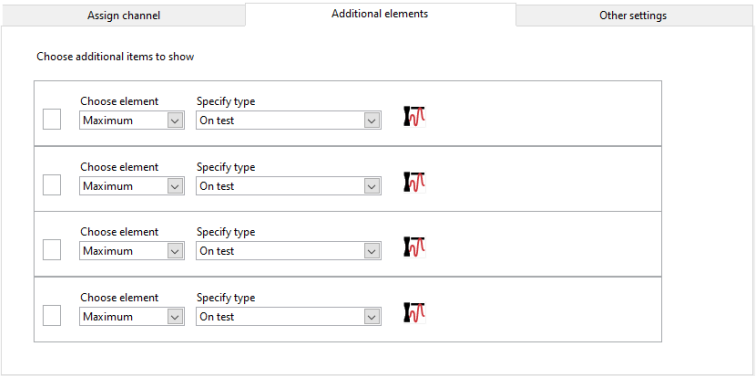
 Opens the configuration interface of the indicator window:



At the top left, it is possible to choose how many rows (from one to four) and columns (from one to four) the grid of indicators should consist of; clicking on one of them colours it blue, and on the right-hand side of the screen it can be configured:



This tab is used to choose the channel to be mapped in the indicator from all those available, and the number of decimal digits to be displayed.

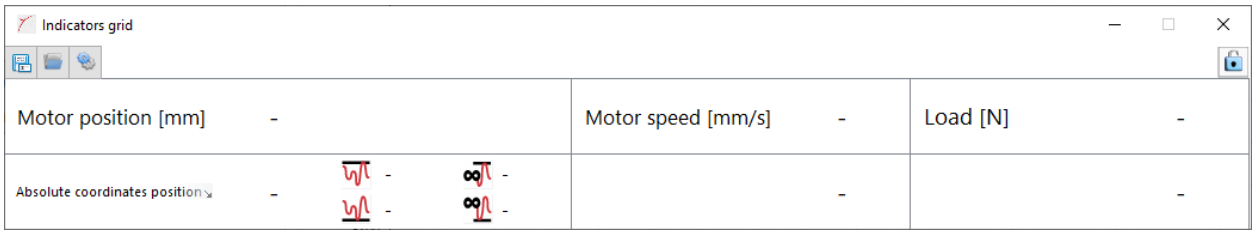


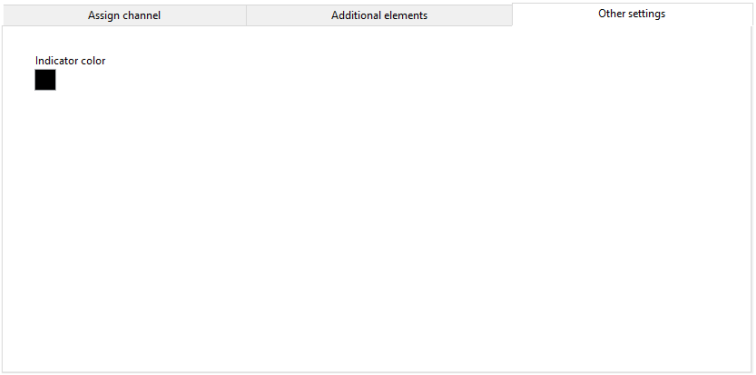
This tab, visible only if the channel configured for the indicator is among the acquisition channels (i.e. not for quantities referring to global variables or the 'other variables' section), allows you to configure

up to a maximum of 4 secondary indicators displaying either maximum, minimum, average, range or amplitude (half of the range)

- ... of the test: the value refers to the current test;
- ... of the cycle: the value refers to the individual cycle and is reset at the start of a new cycle;
- ... since program start-up: the value refers to all the time that has elapsed since the TestCenter software was started;
- ... of a certain time interval: the value refers to a user-configurable time interval.

When an indicator has additional elements configured, it appears as in the following image (bottom left indicator for position in absolute coordinates):



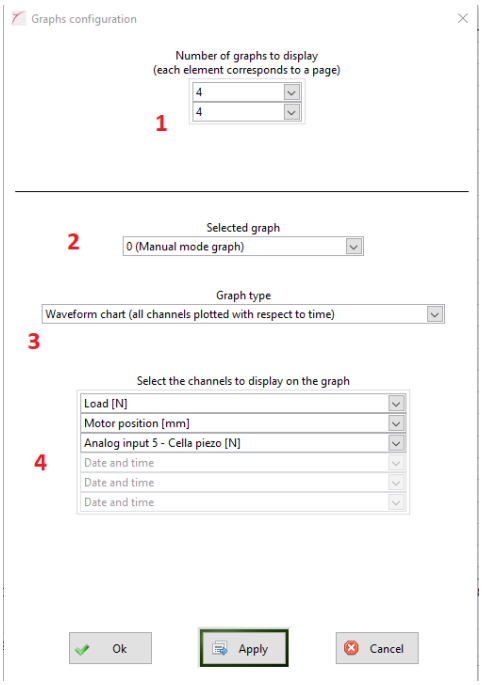


From this tab, the colour of the indicator text can be configured.

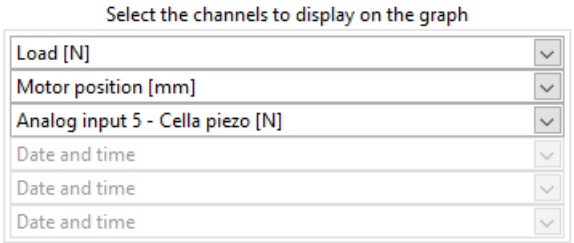
HMI settings – Language: allows the user to change the language of the program; the TestCenter software has to be restarted in order to make the language change effective.

HMI settings – Graphs: this window allows the user to configure up to 9 graphs (manual mode graph + 8 real-time test graphs) that are available for the multiaction test mode (please note that for the standard test environments – traction/compression, creep and fatigue – the real time graphs are pre-configured: in fact, the *Graphs* tab on the main tab control is not visible if one of these tests is loaded). The interface is structured as following:

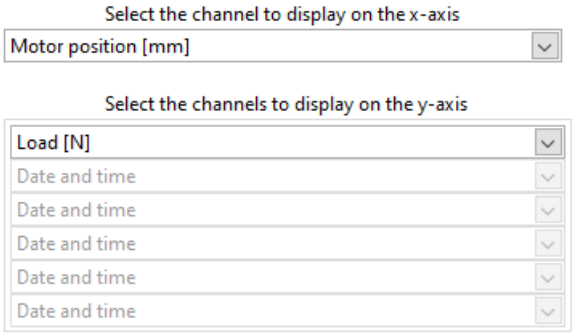
5.4.1. Graphs configuration



1. It is possible to choose how many graphs to display, from 1 to 4, on each of the two windows available inside the *Graphs* tab if a multiaction test is loaded;
2. Through this drop-down menu the user selects the graph to configure;
3. The *Graph type* is the choice between a *waveform chart*, in which one or more channels are displayed with respect to time, and an *xy graph*, where it is possible to choose a channel for the x scale and one or more others for the y scale;
4. This is the menu used to choose the channels to plot on the graph that is currently selected: for waveform charts it displays the various y-axes channels as shown in this picture:



Instead, the configuration menu for an xy graph provides also the drop-down control to choose the channel for the x scale:

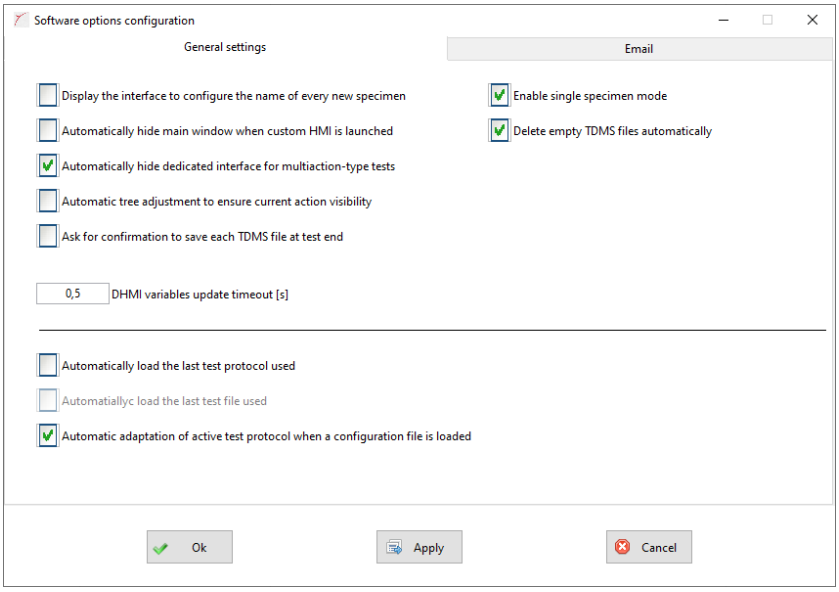


Up to 10 channels are available for each graph.

In order to delete a channel from a graph, just right-click on it and then select *Delete from array*.

HMI settings – Software options: the following interface is displayed to configure some options related to the software behaviour:

5.4.2. Software options configuration

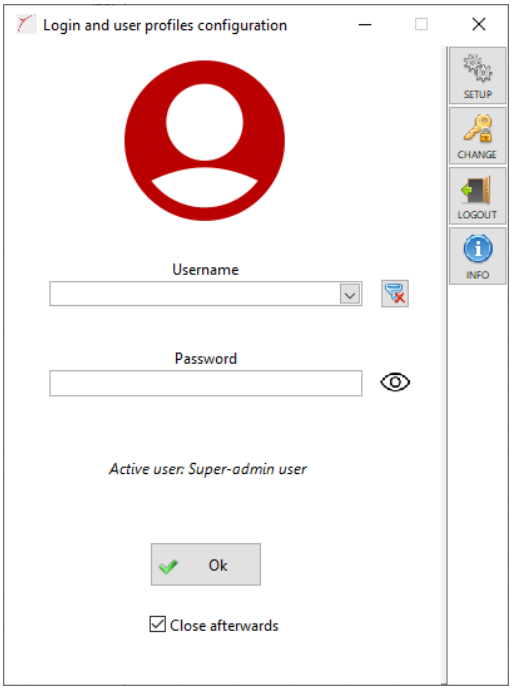


- Display the interface to configure the name of every new specimen: if this option is active, a popup interface is displayed every time a new specimen is created within a test procedure in order to set its name;

- Automatically hide main window when custom HMI is launched: if this option is active, the main TestCenter window is automatically hidden as soon as the *Dynamic HMI* is opened in a multiaction-based test;
- Automatically hide dedicated interface for multiaction-type tests: if this option is active and the current test protocol has a custom floating interface, this is hidden by default;
- Automatic tree adjustment to ensure current action visibility: if this option is active, while a multiaction test is running the software automatically adjusts the scrollbar position of the tree with the test sequence in order to make sure that the current action is always visible;
- Ask for confirmation to save each TDMS file at test end: if this option is active, when a multiaction test finishes the user is asked whether the results file has to be saved or not through a two-option popup message;
- DHMI variable update timeout [s]: this setting only acts on the responsiveness of the *Dynamic HMI*; please refer to the technical support to modify it;
- Enable single specimen mode: this option is valid only for multiaction-based tests; if it is activated, every time a new specimen is added to a test configuration file (except for the first one), the previous specimen is automatically deleted; however, its results file is preserved;
- Delete empty TDMS files automatically: if this option is checked, when a test is stopped the software verifies if there are data stored inside its TDMS results file: if not, the results file is moved into a temporary "bin" folder of the software folders hierarchy (so that it is moved away from the standard saving folder, without permanently deleting it);
- Automatically load the last test protocol used: if this option is active, the TestCenter gives an automatic "New test" command at startup in order to enter the latest test mode used (tension, compression, fatigue, multiaction…);
- Automatically load the last test file saved: this option is deactivated if *automatically load the last test protocol used* is off; if this option is active, the TestCenter automatically loads the latest test file used at startup;
- Automatic adaptation of active test protocol when a configuration file is loaded: this option allows to avoid the necessity of giving a "New test" command when the active test protocol is different from the test type of the file that the user wants to load.

HMI settings – User profile: this command opens the user profiles management window:

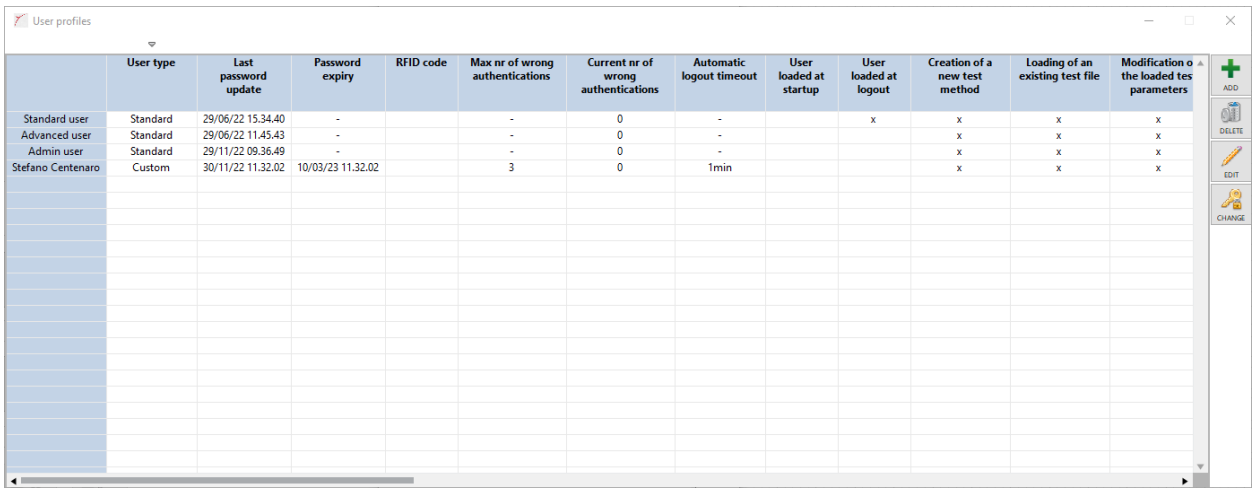
5.4.3. User profile



The "username" and "password" fields can be used to log in with the desired profile; as the username is progressively populated, the software filters the available options of the drop-down menu according to the letters written; the button  clears everything and allows to see the entire user names list. The checkbox named "close afterwards" sets whether the login window has to be closed automatically after a successful login or not.

On the right side of this interface there are some buttons:

If permitted to the active user, this button opens the user profiles setup page:

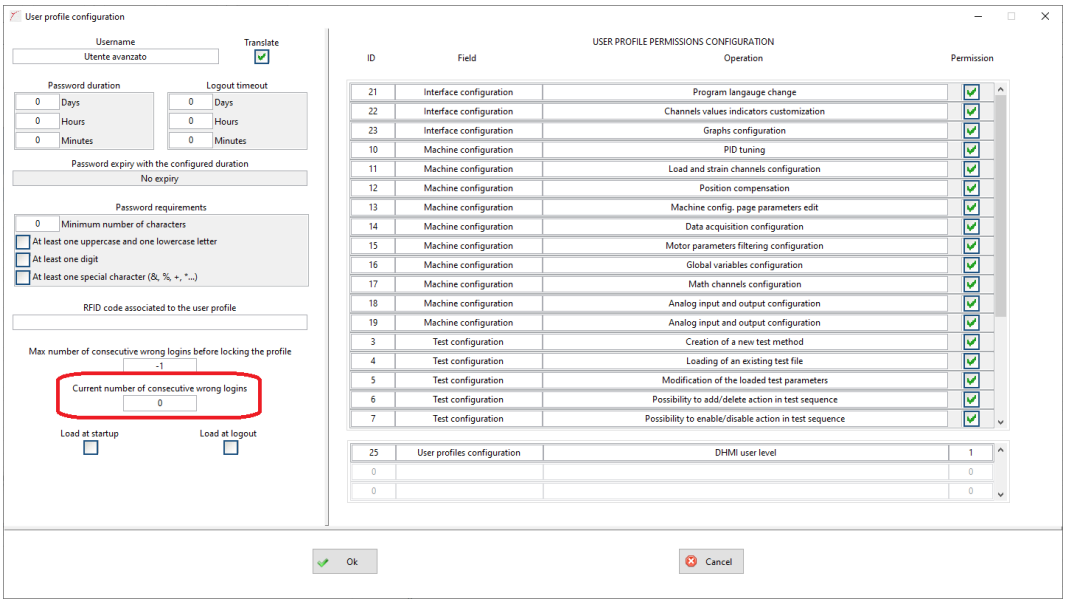


All the existing user profiles, both standard and custom (standard = provided by STEP Lab; custom = created by the users of the software), are displayed in this table, together with all their characteristics and

permissions; for the permissions, if an operation is allowed for a user profile there is a “x” in the corresponding cell; if the cell is blank, the operation is not allowed for that user.

Some significant elements about this table:

- Password expiry: if the password of a user is configured to expire after a certain moment, when the user logs in for the first time after the expiration, the software makes it compulsory to change the password immediately after the login, which is not permitted otherwise;
- RFID code: there is the possibility to assign a RFID code to each user profile in case the system is equipped to include such a device;
- Max and current nr of wrong authentications: it is possible to configure a maximum number of consecutive wrong logins, after which a user profile is automatically locked and requires the intervention of an administrator to be unlocked: in order to unlock a user profile, an administrator has to open the profile configuration page, type 0 (or any other number lower than the maximum number of wrong authentications) in the “current number of consecutive wrong logins” in the picture below and press Ok:



The counter of wrong authentications of a profile is automatically reset when that user logs in properly;

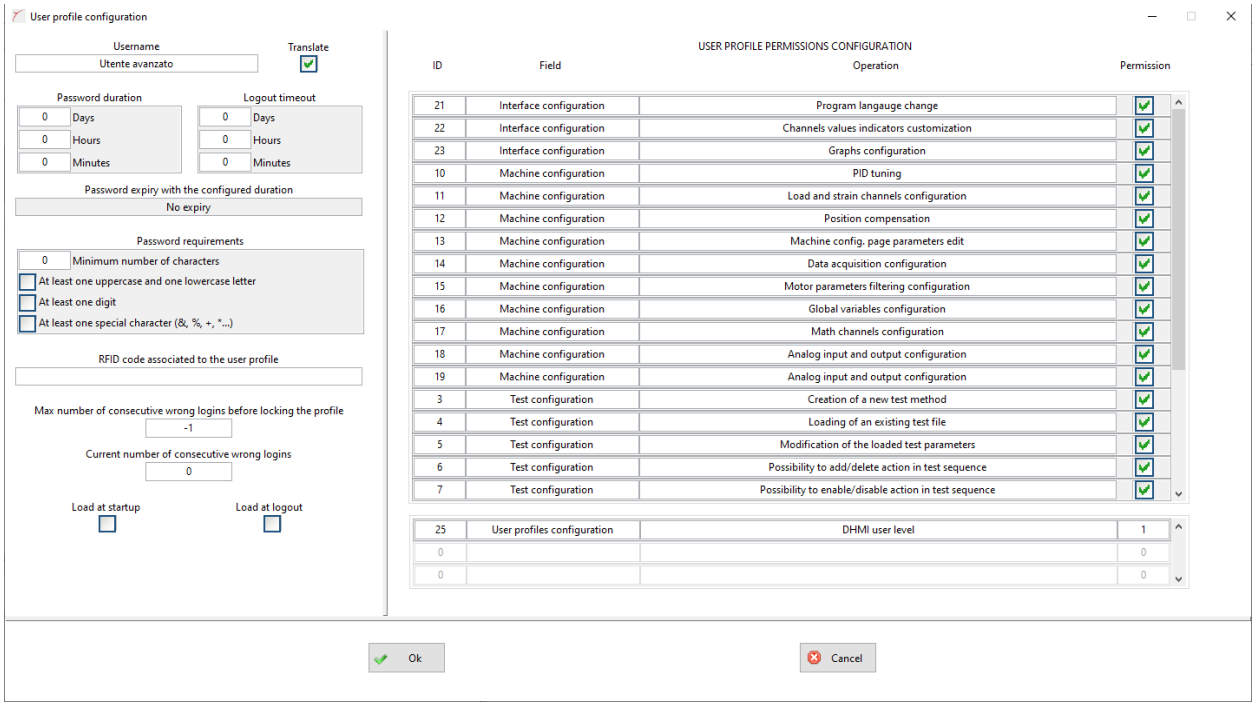
- Automatic logout timeout: if a user profile has this setting configured, it is automatically disconnected in case the specified period elapses without any activity recorded by the software from the mouse or the keyboard.

This button creates a new user profile duplicating the one that is selected on the table (it is required to select a user profile on the table to use this command); when a new user profile is created, it is

given a temporary password that needs to be remembered and then changed by the operator; the user name is also set automatically and can be changed through the user profile configuration window afterwards.

This button deletes the user profile that is currently selected on the table; user profiles marked as standard can't be deleted.

This button opens the user profile configuration window, where system administrators can setup all the configurations and permissions associated to each user profile; an alternative option to get to the same window is double-clicking the row of the desired user profile on the table.



This button opens the window to change the password of the user profile that is selected on the table (see a more detailed description here below).

This button opens the window to change the password of a user profile; if the password has to match some particular requirements (minimum length, special characters, uppercase and lowercase letters…) they are listed together with a ✔ or ✖ indicator which states if the new password selected respects

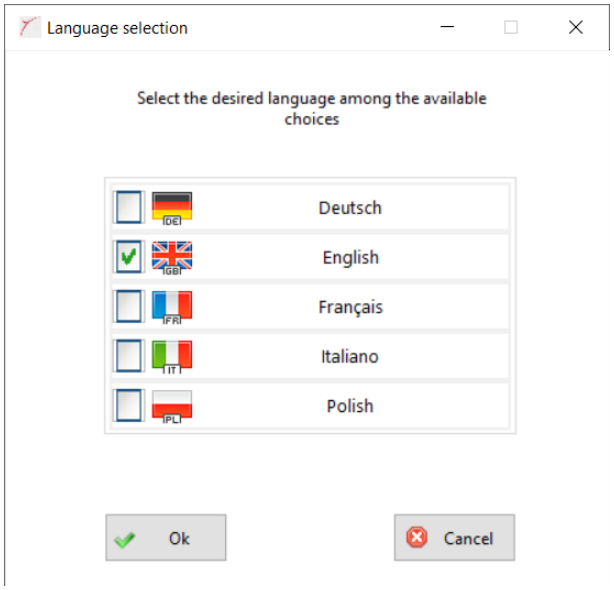
the requirement or not. In general, when changing a password, the new one has to be different from the old one.

Disconnects the active user profile and loads a profile which can be chosen through the functionalities of the "Setup" button described before.

Opens a read-only window which shows all the characteristics and permissions of the active user profile.

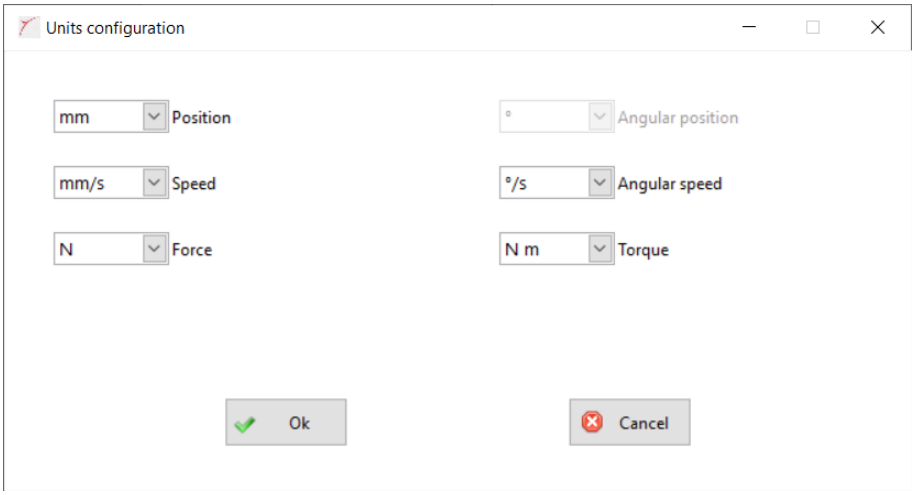
HMI settings – Audit trail mode: the audit trail function consists of recording all significant events for the software (configuration and execution of a test, configuration of the software, PID and input/output channels, etc.) in a local SQL database that resides on the same computer running the TestCenter; enabling or disabling this functionality is only allowed to STEP Lab personnel.

HMI settings – Language: this option opens the programme language selection window:



In order for a language change command to take effect, it is mandatory to restart the TestCenter.

HMI settings – Units of measurement: this option opens the window for selecting the units used by the program:



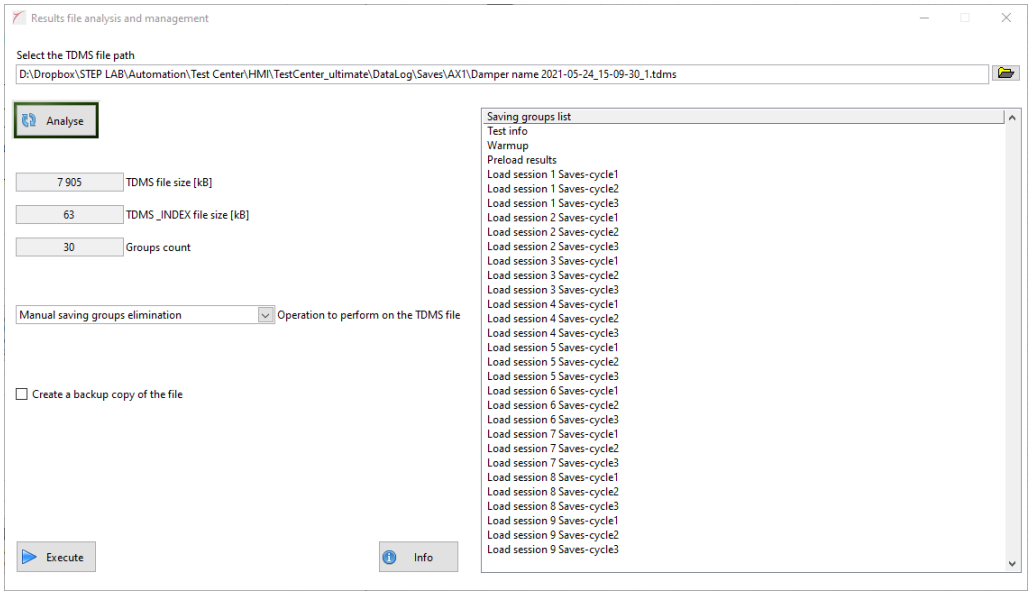
The quantities for which the default units can be changed are position, velocity, force, angular velocity and torque.

5.5. Tools *submenu*

Tools – Monitoring: this command displays a window dedicated to the monitoring of some machine and – only for *Advanced users* or higher profiles – software parameters: in particular, the indicator called *I2t* is significant to detect the machine stress state.

Tools – Semiauto: the usage of this option is reserved to *Admin users* since it is dedicated to debugging and diagnostics.

Tools – TDMS file management: this command displays an interface that can be used to reduce the size of a TDMS file that grew too big:



Once the desired TDMS file is selected, it is necessary to click the *Analyse* button to run the file analysis procedure that populates the indicators (size of the TDMS file and associated TDMS_INDEX file, number of groups in the TDMS file) and the group list on the right. At this point it is possible to choose among the following operations through the drop-down menu:

- Manual saving groups elimination: the user has to select manually the groups to be eliminated from the files using the list on the right; multiple selection is supported by holding the CTRL button;
- Saving groups decimation: the software automatically decimates the groups according to a certain percentage specified by the user; please pay attention to the fact that this procedure is studied for

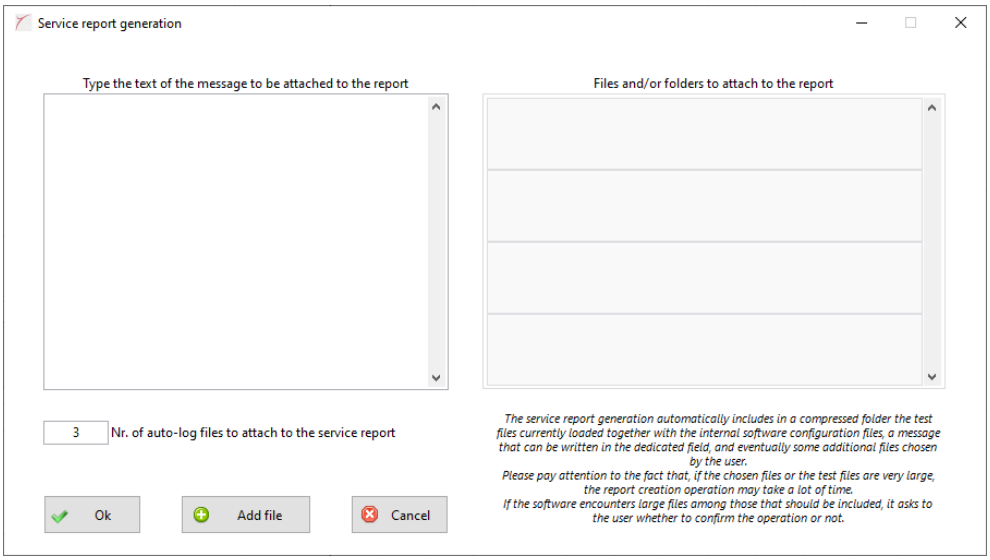
fatigue test files (that have hundreds of groups of the same type for the various cycles saved) and should not be used for multiaction-based tests with lots of groups related to different actions;

- File defragmentation: runs an automatic file optimization procedure that may reduce the time needed to process the TDMS file if it has become cluttered;
- File regeneration: reads all the data in the file, clears the original file and replaces it with another one where all the data are written at the same time; in this way the access to the file will be as quick as its size allows to.

The *Create a backup copy of the file* checkbox is a safety option that is recommendable to prevent damages due to user mistakes (such as an incorrect selection of the groups to be eliminated) or unpredictable accidents (a power supply blackout while the procedure is running may corrupt the original file).

Once everything is ready, the *Execute* button runs the procedure.

Tools – Report an issue: this command can be used to generate zip file archives that include information to be provided to the technical support in case an abnormal behaviour occurs in the testing system. Please pay attention to the fact that the software only generates the zip file and does not send it to the technical support automatically.

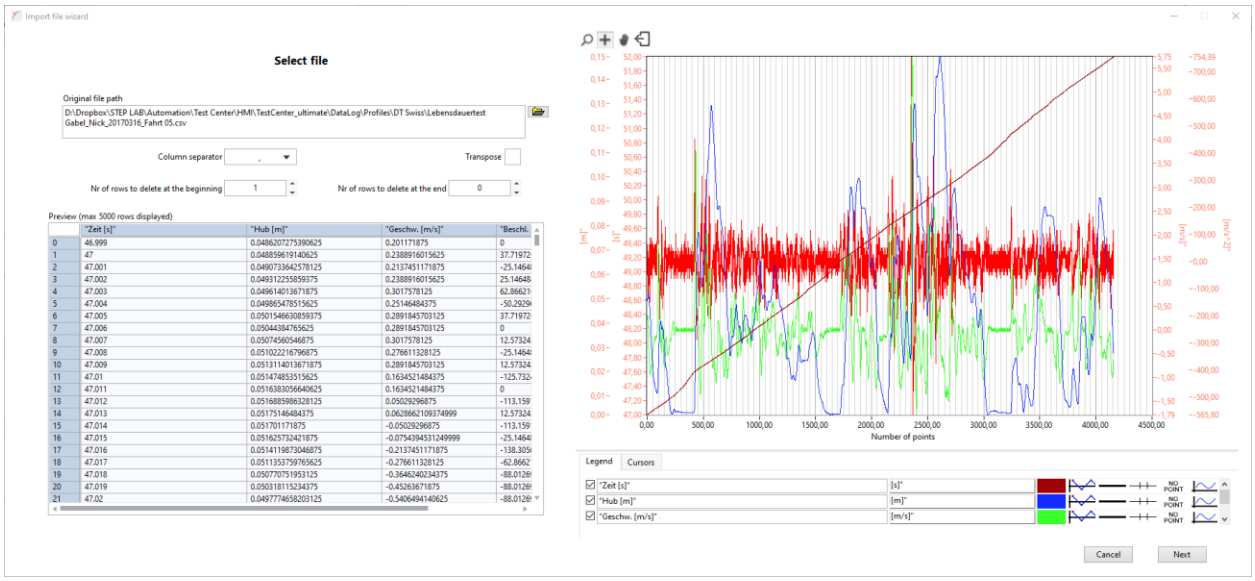


The text control on the left provides the possibility to type a description of the problem that will be attached to the service report; on the right side there is a list of eventual additional files (pictures of the machine, screenshots…) that can be also manually included through the *Add file* button; finally, the *Nr. of auto-log files to attach to the service report* is the number of the latest files to be included among the TDMS files that are automatically generated by the program for diagnostics purposes at every test end.

Once all the settings and additional files are ready, by pressing the *Ok* button the software starts generating the service zip archive.

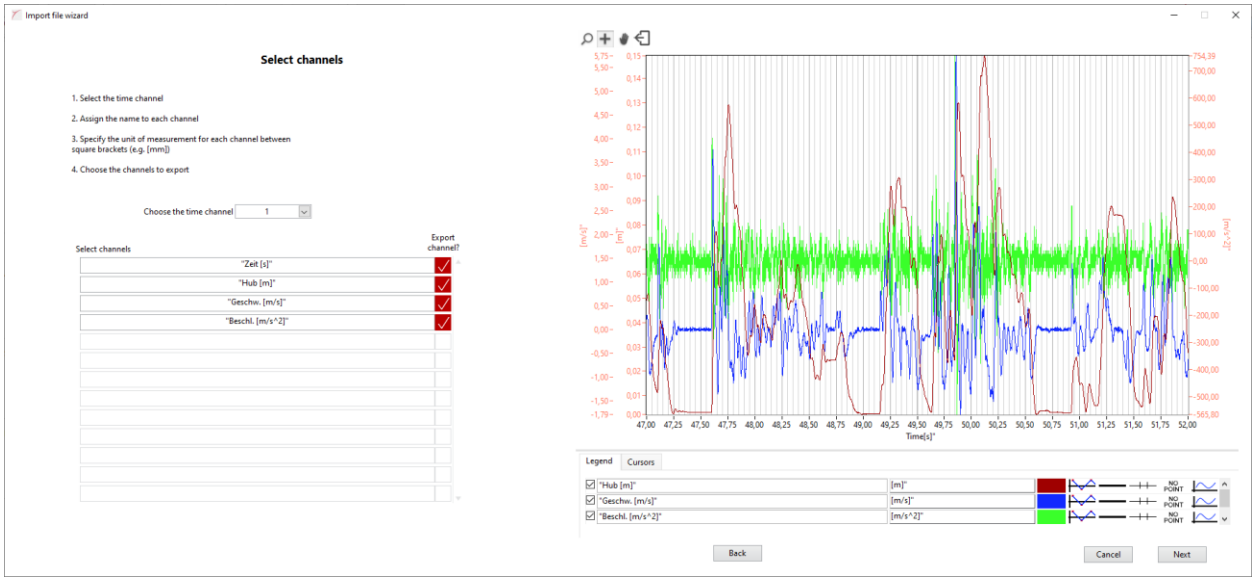
Tools – Debug mode: the usage of this option is reserved to *Admin users* since it is dedicated to debugging and diagnostics.

Tools – Profile processing: this is a software accessory that can be used to format files with random profiles into other files that can be imported into the *ProfileTracking* actions of multiaction sequences:

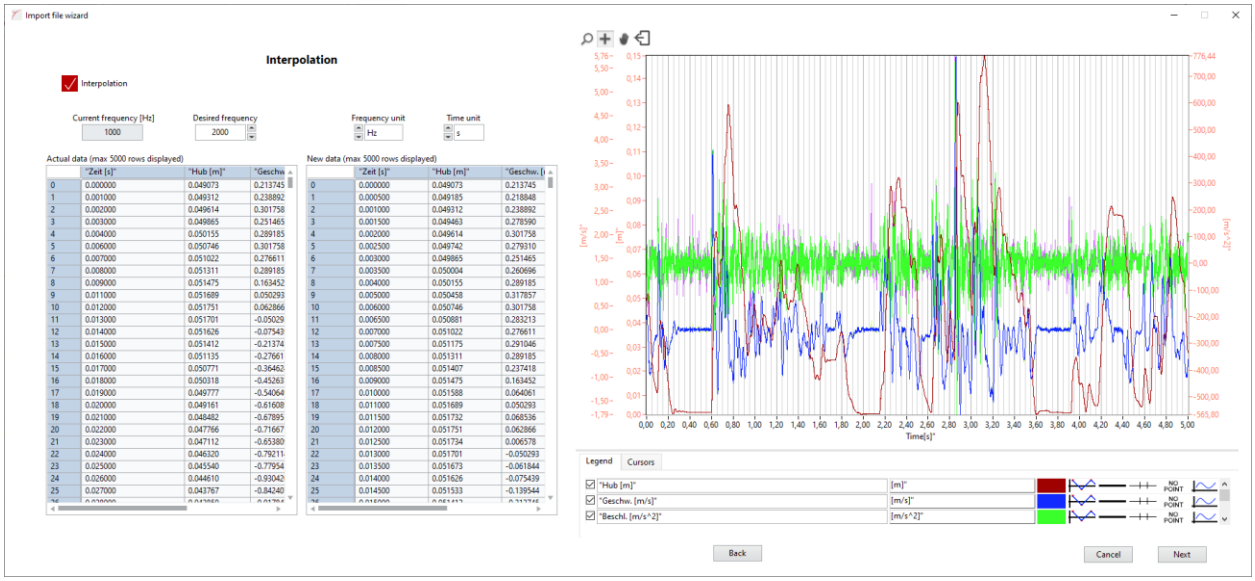


On the first page the user selects the file to import and the column separator used in it; it is also possible to delete some rows at the beginning and/or at the end of the files in order to achieve the layout exemplified in the table of the picture above: the first row (table header) has to include the channel names with their units of measurements, and data start from the second row, with one column for each channel.

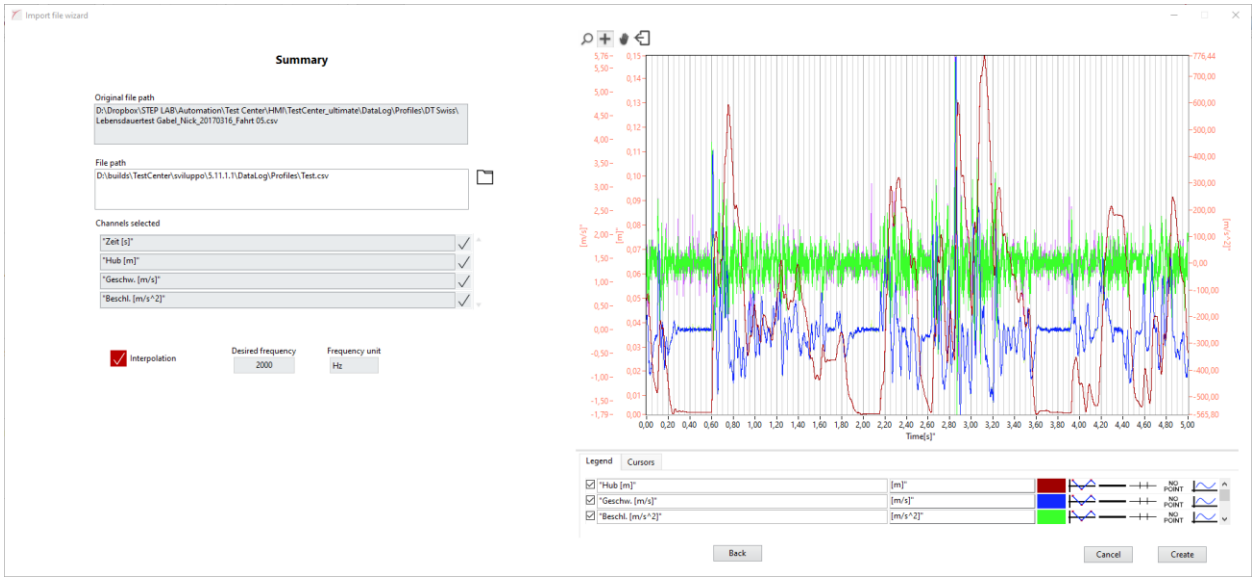
On the right of the interface a preview graph shows all the data sets of the file.



The second page of the procedure is designed to choose the channels to include in the new file and, above all, the channel that represents time, that is going to be placed in the first column.



The third page is dedicated to interpolation: it is not obligatory (unless the native sampling frequency is not compatible with the acceptable range that goes from the controller clock frequency down to 1/20 of it), but it is possible to interpolate the signals in order to vary the sampling frequency as exemplified in the picture above. If the checkbox is not selected, the native frequency is preserved.



In the final page it is possible to choose the path and name of the output file. Once the setup is completed, the software generates the new file as soon as the *Create* button on the bottom-right corner of the interface is pressed.

Tools – Recovery archive generation: recovery archives are zip file that can be used by the program to restore the configuration files in case the computer is powered off violently, without closing the software in the appropriate manner. This command manually generates a recovery archive, whereas other automatic commands are executed by the program periodically.

Tools – Action change report: this command generates a txt file with a report related to the test sequence, that is only useful for debugging and diagnostics.

5.6. Help *submenu*

Help – Software info: an informative window with the software characteristics (HMI version, controller software version, etc.) is displayed.

Help – Machine info: an informative window with the machine characteristics (machine code, safety limits, etc.) is displayed.

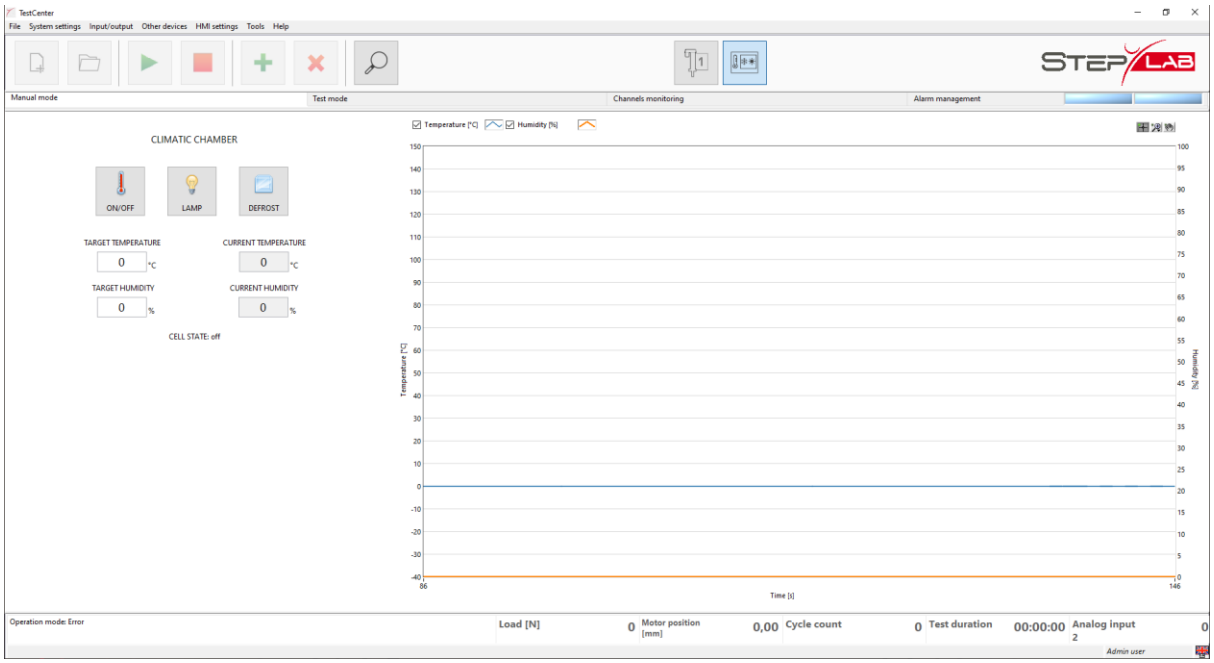
Help – Web: if the computer is online, the web browser is opened at the STEP Lab website address.

Help – Software user guide: automatically opens the TestCenter manual in the default PDF display environment; the manual is loaded in the current language of the software, or in English if that language is not available.

Help – Machine user guide: automatically opens the machine operating and maintenance manual in the default PDF display environment; the manual is loaded in the current language of the software, or in English if that language is not available.

6. Climatic chamber

The TestCenter software incorporates a module to control a climatic chamber, if this system is included as part of the STEP Lab machine. The following image displays the layout of the interface provided to control the chamber, that can be accessed by selecting the button with the climatic chamber icon on the upper bar, next to the axes selectors.



On the left side of the page there are three buttons that command the following functionalities:

ON/OFF: switches the climatic chamber on or off; please note that when the chamber is turned off it takes a couple of minutes to complete the shutdown procedure, as a dedicated popup window explains to the user;

LIGHT: switches the lamp inside the chamber on or off; if this button is disabled and grey, it means that the machine has not a lamp inside the chamber or that the lamp is commanded through a switch on the electric cabinet;

DEFROST: switches the defrost system on and off; the defrost functionality is useful when the chamber door sticks to the structure, but after some minutes it is automatically deactivated by the controller to preserve the electric resistances; if this button is disabled and grey, it means that the climatic chamber is not equipped with a defrost system.

Below the buttons there are a numeric control to command the target temperature and an indicator that displays its current value. Also, a control and an indicator of the same type are available for humidity in case the chamber allows humidity control.

Finally, a short text summarises the chamber current state.

On the right of the interface, a graph profiles the temperature in real time, with a memory of one minute.

7. Post-processing environment (*TDMS Viewer)*

The post-processing environment is distributed as a stand-alone application and has its own user guide, which is provided together with this manual as part of the TestCenter software documentation.

8. General elements

8.1. User profile

Starting from software version 3.6, some different user profiles have existed within the program, each of them having different capability of acceding and editing configuration parameters and environments:

Basic user: this is the lowest user profile: a *basic user* can't edit the settings of the tests that are related to the operation of the machine – only archive fields such as annotations and specimen names are possible to modify for this user level; also, a *basic user* has limited access to the configuration environment;

Standard user: this is the default usage level; a standard user can perform any manual operation and any test included in the program, configure and edit test templates and edit the basic configuration settings;

Advanced user: other than everything is allowed to standard users, advanced ones can enter and edit some more areas of the configuration menu dedicated to configure the machine and the software, among which the position compensation parameters and the acquisition; a password is required to use this user profile;

Admin user: this user profile is reserved to STEP Lab personnel to perform maintenance, advanced configuration and software diagnostics.

8.2. Operation mode

Paragraph 1 describes the lower bar that displays, among other information, the operation mode of the machine on its left side; four different modes exist:

Standby: the machine is still and ready to receive commands;

Manual: the machine is executing a manual operation (actuator motion through the buttons on the electric cabinet or their equivalent on the interface);

Auto: the machine is performing an automatic procedure, which is any procedure different than manual motion that involves a movement of the axis or a sequence of actions: displacement to a given position, null load maintenance, a test, etc.;

Error: the machine is in alarm state.

Depending on the operation mode, the possibility to command some operation is allowed or denied; in particular:

- When the machine is in manual or automatic mode it is not possible to start other manual or automatic procedures; if the user tries to do it, a warning message is displayed;
- When there are active alarms it is not possible to start any automatic operation: if the user tries to do it, a warning message is displayed;

- When the door alarm is disabled through the dedicated command on the electric cabinet, it is not possible to start any automatic procedure except for the null load maintenance, and the manual movement arrow buttons on the interface do not work;
- When the machine is in manual or automatic mode it is not possible to apply new configuration settings through the dedicated window described in paragraph 5.2.1.

8.3. Software shutdown and restart

For safety reasons, when the program is closed by the user, any automatic procedure eventually running is automatically stopped. Instead, if the computer is shut down abnormally (for example, by accidentally disconnecting the power supply) while the machine is running, when the TestCenter is restarted it automatically detects and stops the procedure, informing the user about this fact with a popup message.

Anyway, if the computer gets lost during a test, it is strongly recommended to stop the machine through the hardware commands, in particular by pressing the two manual movement arrows on the electric cabinet at the same time; the emergency button must be avoided for this usage if a high load or high speed test is running.

9. Installation instructions

The software is already installed and fully functional on the PC provided with the testing machine: it is located in the following path: *C:\TestCenter*; shortcuts on desktop and on the application bar are also available.

10. General instructions for use

While operating with the program the following recommendations must be followed (some of them have already been mentioned in this guide):

- When the crossbeam manual movement is used it is extremely important to unscrew all the screws that lock it, otherwise the motor dedicated to move the beam gets damaged or broken; since there are no sensors that can verify the state of the screws, it is entirely up to the user to take care of this element and communicate it to the software through the dedicated checkbox placed in the manual handling page; it is likewise important to uncheck the box when the screws are locked again after completing the crossbeam movement;
- When performing a fatigue test, the frequency must be calibrated in order to keep the *I2t* parameter less or equal to the alarm threshold (the value of this parameter can be monitored on the *Tools – Monitoring* window), even if a protection algorithm is provided to automatically decrease the test frequency in case a critical threshold is exceeded for this parameter;
- The software must not be closed while a test is running, without giving the stop command to the machine through the dedicated button on the upper bar;
- If the computer is suddenly shut down without stopping the machine and closing the interface when an automatic test is running, this procedure should be interrupted by pressing the two manual movement arrow buttons on the electric cabinet at the same time;
- Any time the electric cabinet needs to be restarted, at least 30 seconds must pass between the shutdown and the restart;
- When a fatigue test that should last tens or hundreds of thousands of cycles is being configured, the percentage of cycles to save has a major impact on the size of the log file, which might grow too big to be handled if this parameter is not set to a proper value; here are some guidelines:

Expected test duration (cycles)	Recommended % of cycles to be saved
<1000	any
1000..5000	up to 25%
5000..20000	up to 10%
20000..100000	up to 2%
100000..1000000	up to 0,2%
>1000000	less than 0,1%

Please note that these indications are really general and depend also on other parameters (test frequency, PLC scan time); they may sometimes be exceeded in case lots of data are needed for a particular test, and in conclusion experience is the best practice to properly set this value;

- In order to prevent performances from decreasing, while executing static tests (tension, compression, creep…) it is recommended not to exceed 20-25 specimens in the same configuration files and 50-60 MB for the results file: if lots of specimens are needed within the same test campaign or each test lasts for a long time, the entire set of tests should be divided into more than one configuration file, eventually using the *rename test* functionality in order to group them in a dedicated directory.

STEP LAB

Via Castellana 199, 31023 Resana - Treviso – ITALY
Tel.: +39 0423 1999391
info@step-lab.com
www.step-lab.com