O A C I S

Open **A**rchitecture **C**ontrol **I**ntegrated **S**ystem

Function Description

Version 05.00.02

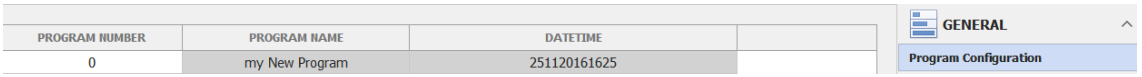
www.atainc.com
ata@atainc.com

CONTENTS

I. CONFIGURATION

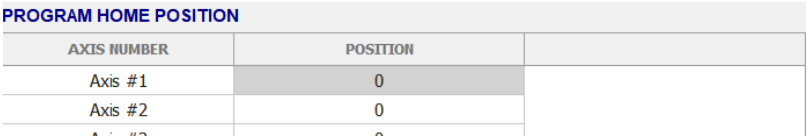
#Note: It is a separately used Program Configuration (different from “System Configuration”) per one program. And it is not a “Function”. But it is to be used in following Functions.

A. Program Information



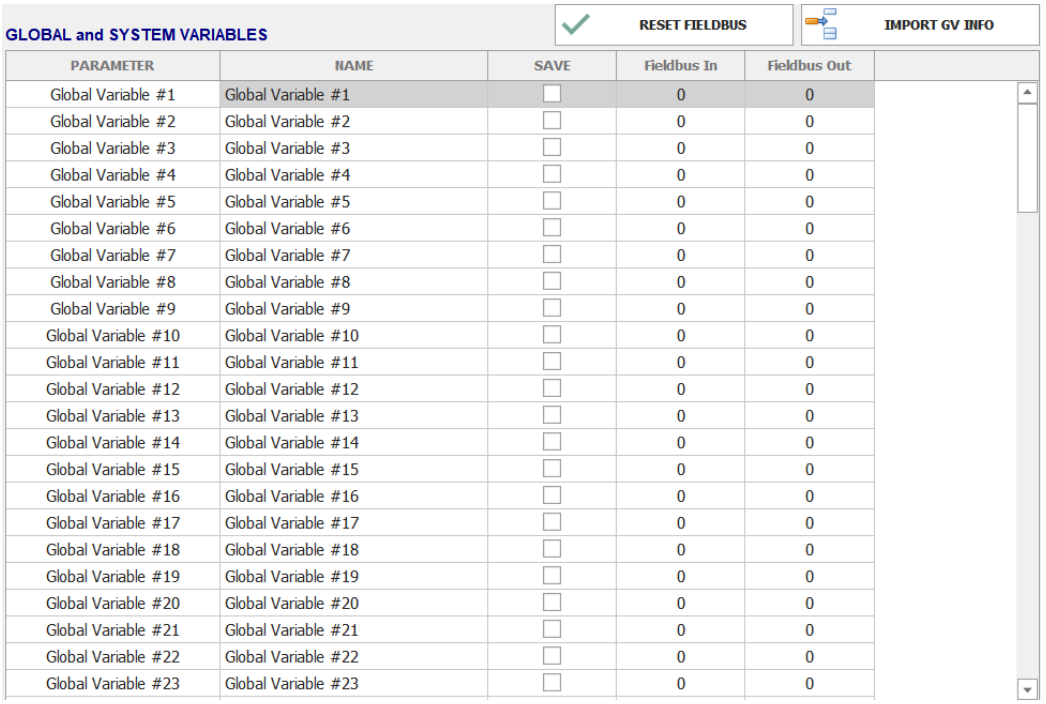
- **PROGRAM NUMBER:** It is to be one of 1 ~ 120. Only one program is allowed for the specific program number in the OACIS. But multi programs can be saved in the local PC with same program number if program name is different.
- **PROGRAM NAME:** Min string length is 1 and max string length is 32 bytes.
- **DATETIME:** Time when the program saved. It is to be created by program. You do not need to input.

B. Program Home Position



- You can assign the program home position of each axis. The program home position is to be used for “Move to Program Home” function. To run a cycle. all axes need to be at home position or program home position.

C. Global Variables and System Variables



PARAMETER	NAME	SAVE	Fieldbus In	Fieldbus Out
Global Variable #1	Global Variable #1	☐	0	0
Global Variable #2	Global Variable #2	☐	0	0
Global Variable #3	Global Variable #3	☐	0	0
Global Variable #4	Global Variable #4	☐	0	0
Global Variable #5	Global Variable #5	☐	0	0
Global Variable #6	Global Variable #6	☐	0	0
Global Variable #7	Global Variable #7	☐	0	0
Global Variable #8	Global Variable #8	☐	0	0
Global Variable #9	Global Variable #9	☐	0	0
Global Variable #10	Global Variable #10	☐	0	0
Global Variable #11	Global Variable #11	☐	0	0
Global Variable #12	Global Variable #12	☐	0	0
Global Variable #13	Global Variable #13	☐	0	0
Global Variable #14	Global Variable #14	☐	0	0
Global Variable #15	Global Variable #15	☐	0	0
Global Variable #16	Global Variable #16	☐	0	0
Global Variable #17	Global Variable #17	☐	0	0
Global Variable #18	Global Variable #18	☐	0	0
Global Variable #19	Global Variable #19	☐	0	0
Global Variable #20	Global Variable #20	☐	0	0
Global Variable #21	Global Variable #21	☐	0	0
Global Variable #22	Global Variable #22	☐	0	0
Global Variable #23	Global Variable #23	☐	0	0

- There are 100 global variables and 20 system variables. The variables are to be used at various functions like “Move to Position by Var”. Global Variables and System Variables are basically same except that the Global Variables are to be reset as zero by “Reset All Global Variables” but System Variables are not to be reset. (You can reset System Variable by using “Set Global Variable” function.)
- **Name**: You can assign specific name for each variable. The name should be unique.
- **Save**: You can set as “save” by checking the check box. Whenever OACIS complete a whole cycle, it saves all 120 variables with cycle number, program name and time information. And you can see

only “save” checked variables on the “Result” and “DATA” tab of main screen.

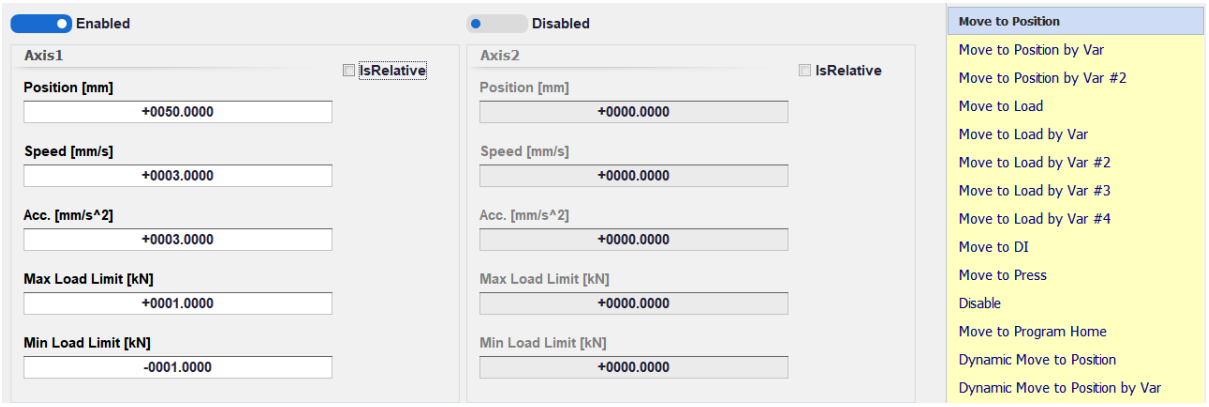
- **Fieldbus In**: OACIS can receive mapped GV values from PLC by assigning the numbers from 1 to 45. You can assign Max 45 and number 0 means no mapping. You does not need to assign the numbers in order. But it is not allowable to reuse the same number. Additionally, simultaneous allocation of fieldbus in and out for a specified GV is prohibited.
- **Fieldbus Out**: It is the same as Fieldbus In except that OACIS can send mapped GV values to PLC. Here is an example of mapped Fieldbus In and Out below.

PARAMETER	NAME	SAVE	Fieldbus In	Fieldbus Out
Global Variable #1	MinFreeLoad	☑	0	0
Global Variable #2	MaxFreeLoad	☑	0	0
Global Variable #3	MaxRunLoadPos	☑	0	0
Global Variable #4	MaxRunLoad_Internal	☑	0	0
Global Variable #5	MaxRunLoad_External	☑	0	1
Global Variable #6	MaxRunLoad_Var	☑	0	0
Global Variable #7	EndPos	☑	0	3
Global Variable #8	EndLoad	☑	0	0
Global Variable #9	ContactPos	☑	0	0
System Variable #1	Spec_Load_LL	☐	1	0
System Variable #2	Spec_Load_UL	☐	2	0
System Variable #3	System Variable #3	☐	0	0
System Variable #4	System Variable #4	☐	0	0
System Variable #5	Spec_Distance_LL	☐	5	0
System Variable #6	Spec_Distance_UL	☐	6	0

- **IMPORT GV INFO**: You can import GV information from another OACIS program.
- **RESET FIELDBUS**: You can set all the assigned values of fieldbus in & out as 0.

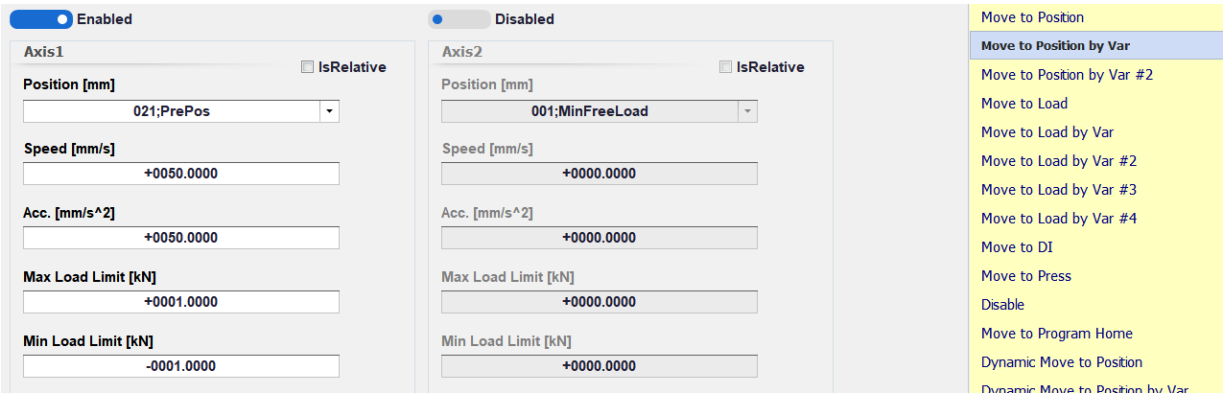
II. MOVE

A. Move to Position



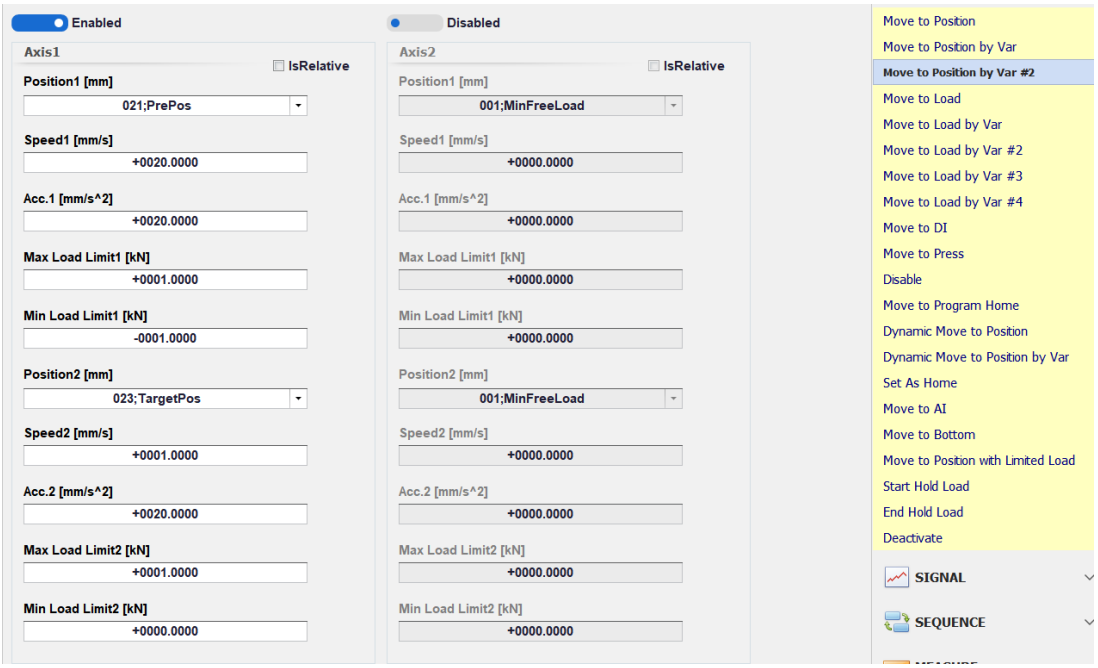
1. **Description:** Move the selected Axis to the specified position. Multi Axes can be selected.
2. **Parameters:**
   - **Position**: Target Position to move / [mm] or [deg]
   - **Speed**: [mm/s] or [deg/s]
   - **Acceleration**: [mm/s^2] or [deg/s^2]
   - **Max Load Limit / Min Load Limit**: If the load is higher than this limit, OACIS stops the operation and sends an error message / [kN] or [Nm].
   - **Absolute or Relative**: You can use “Absolute” position or “Relative” position.
   - **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

B. Move to Position by Var



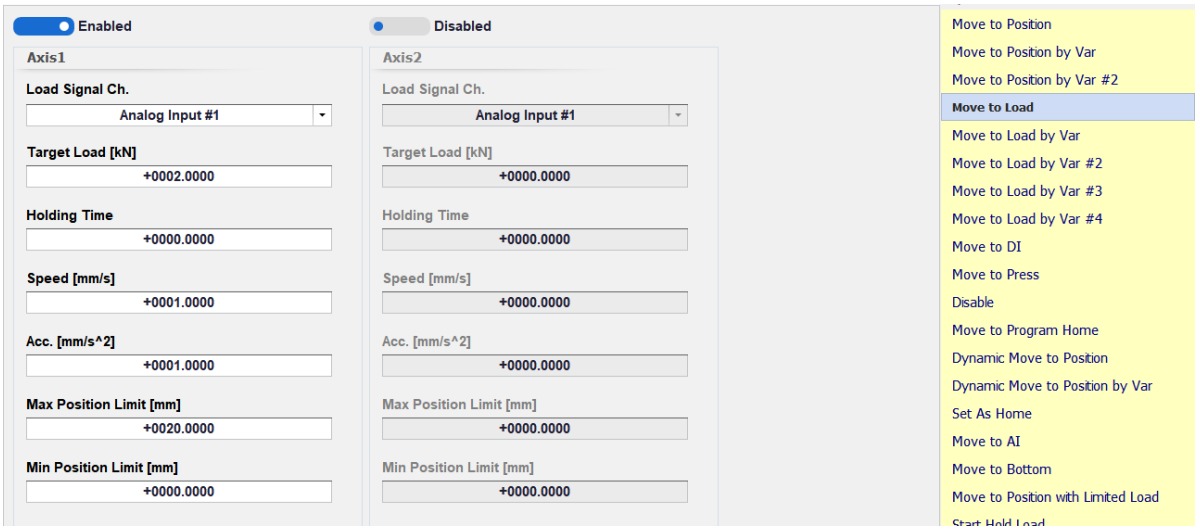
1. **Description:** It is same as “Move to Position” except that the target position is to be assigned by selected global variable. Multi Axes can be selected. This function allows you to move the axis to various positions depending on process. The variable could be captured from analog signal or to be calculated to compensate proper position.

C. Move to Position by Var #2



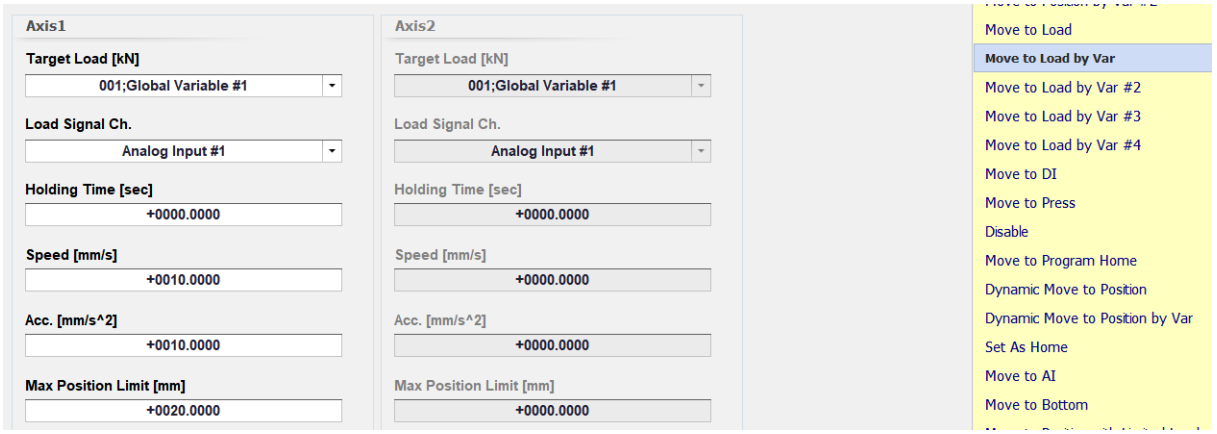
1. **Description:** It is the same as “Move to Position by Var” function is applied twice except that the accelerating profile between the two “Move to Position by Var” is different from “Move to Position by Var #2”. Multi Axes can be selected. This function allows you to move the axis to various positions without any unnecessary accelerating or decelerating period. The variables could be captured from analog signal or to be calculated to compensate proper position.
2. **Parameters:**
   - **Position 1**: Target Position to move / [mm] or [deg]
   - **Speed 1**: [mm/s] or [deg/s]
   - **Acceleration 1**: [mm/s^2] or [deg/s^2]
   - **Max Load Limit 1 / Min Load Limit 1**: If the load is higher or lower than this limits, OACIS stop the operation and send an error message / [kN] or [Nm].
   - **Absolute or Relative**: You can use “Absolute” position or “Relative” position.
   - **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.
   - The usage of No. 2 group like Position 2, Speed 2 and so on is the same as No. 1.

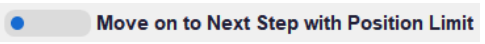
D. Move to Load



1. **Description:** Move the selected Axis to the target load. Multi Axes can be selected.
2. **Parameters:**
   - **Target Load Ch:** Analog Signal Input Channel for motion control. It is recommended to use default Analog Input Signal Channel.
   - **Target Load:** Target Load to move / [kN] or [Nm]
   - **Holding Time:** OACIS holds on the target load for the duration / [sec].
   - **Speed:** [mm/s] or [deg/s]
   - **Acceleration:** [mm/s^2] or [deg/s^2]
   - **Max Position Limit / Min Position Limit:** If the OACIS fail to reach at the target load within the specified position limit, OACIS stop the operation and send an error message / [mm] or [deg].
   - **Step Tag:** The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

E. Move to Load by Var



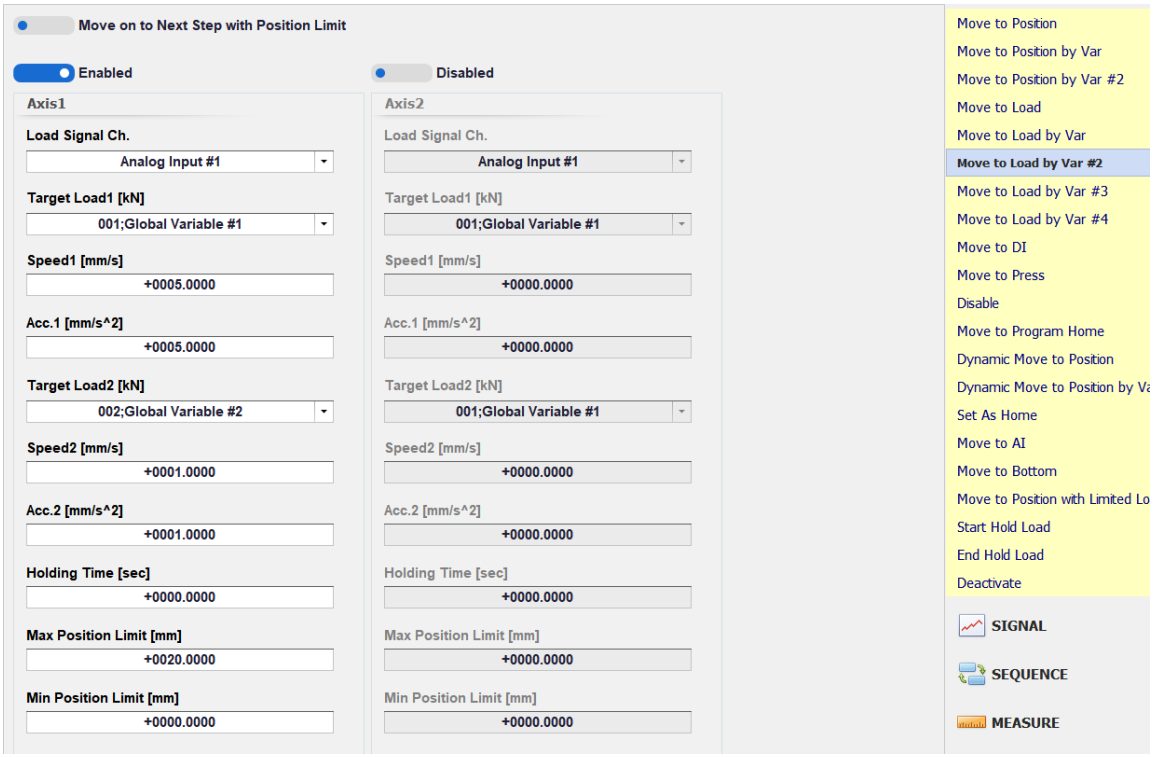
1. **Description:** It is same as “Move to Load” except for followings.
   - Target Load is to be assigned by selected global variable.
   - “Move On to Next Step with Position Limit”
     - 

     - With this option, the cycle will move on to next step even if it fails to reach at the target load. If you want to move the selected Axis to the target load **or** the position limit, this option would allow you to do that. For sure, the Target Load has priority.
   - “Error and Stop with Position Limit”

➢ **Error and Stop with Position Limit**

➢ With this option, it is working like “Move to Load”. It means, if the cycle fails to reach at the target load, it stop at the position limit with “Error” signal. (with 602, 702, 802 or 902 error code).

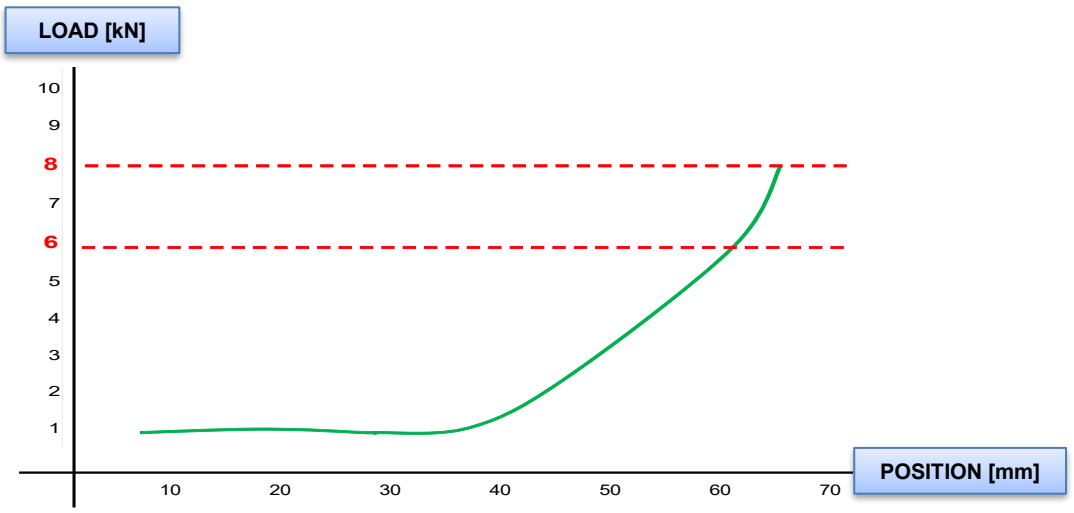
F. Move to Load by Var #2



1. **Description:** It is working like “Move to Load” or “Move to Load by Var”. But you can specify Target Load #1 and Target Load #2 respectively. Each Target Load has its own Speed and Acceleration. By specifying two Target Loads, you can get more accurate result. If you use fast speed for “Move to Load” function, it is hard to prevent “Over Shoot”. But low speed requires longer cycle time. So, you can move to Target Load #1 fast and move to Target Load #2 slowly to reach at the final Target Load without “Over Shoot”.
2. **Parameters:**
   - **Target Load Ch**: Analog Signal Input Channel for motion control. It is recommended to use default Analog Input Signal Channel.
   - **Target Load #1**: First Target Load to reach / [kN] or [Nm]
   - **Speed #1**: Speed to reach at Target Load #1 / [mm/s] or [deg/s]
   - **Acceleration #1** : Acceleration to reach at Target Load #1 / [mm/s^2] or [deg/s^2]
   - **Target Load #2**: Final Target Load to reach / [kN] or [Nm]
   - **Speed #2**: Speed to reach at Target Load #2 / [mm/s] or [deg/s]
   - **Acceleration #2** : Acceleration to reach at Target Load #2 / [mm/s^2] or [deg/s^2]
   - **Holding Time**: OACIS holds on the Target Load #2 for the duration / [sec].
   - **Max Position Limit / Min Position Limit**: If the OACIS fail to reach at the target load within the specified position limit, OACIS stop the operation / [mm] or [deg].
   - “**Move On to Next Step with Position Limit**”
     - ➢ **Move on to Next Step with Position Limit**
     - ➢ With this option, the cycle will move on to next step even if it fails to reach at the target load. If you want to move the selected Axis to the target load or the position limit, this option would allow you to do that. For sure, the Target Load has priority.
   - **“Error and Stop with Position Limit”**

- ➢ **Error and Stop with Position Limit**
- ➢ With this option, it is working like “Move to Load”. It means, if the cycle fails to reach at the target load, it stop at the position limit with “Error” signal. (with 602, 702, 802 or 902 error code).

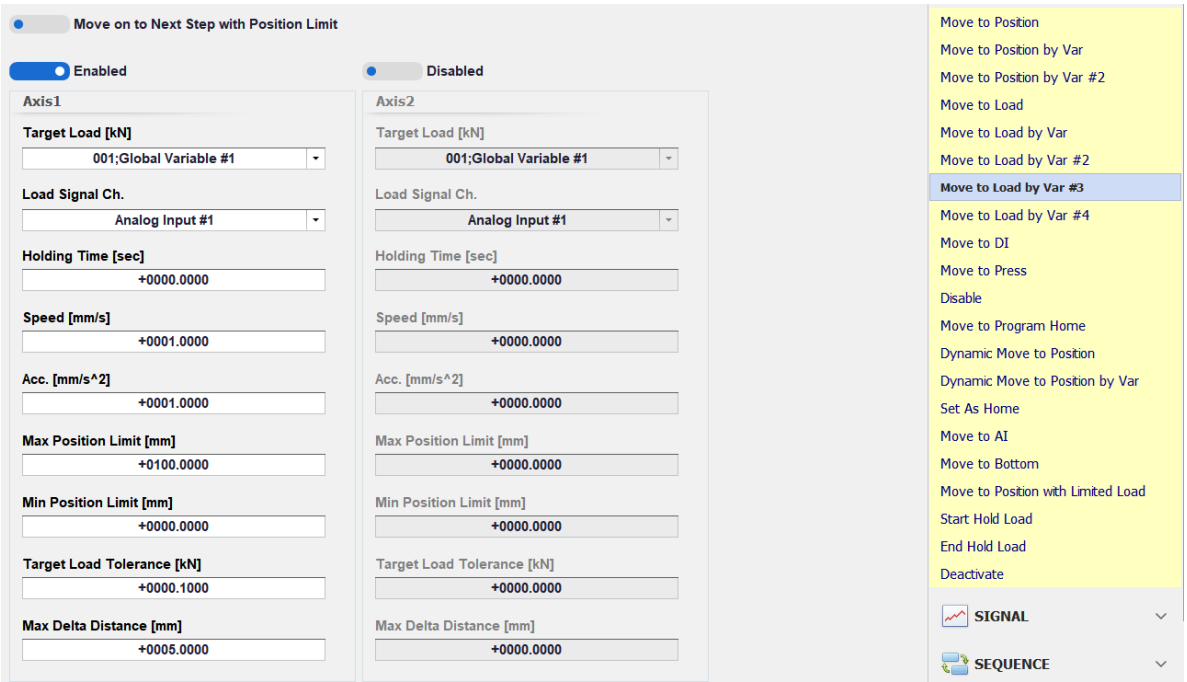
3. **Example:**



If your Target Load is 8kN, you can set the parameters as below,

- Target Load #1: 6 kN
- Speed #1: 5 mm/s
- Acc. #1: 10 mm/s^2
- Target Load #2: 8 kN
- Speed #2: 0.5 mm/s
- Acc. #2: 1 mm/s^2

G. Move to Load by Var #3

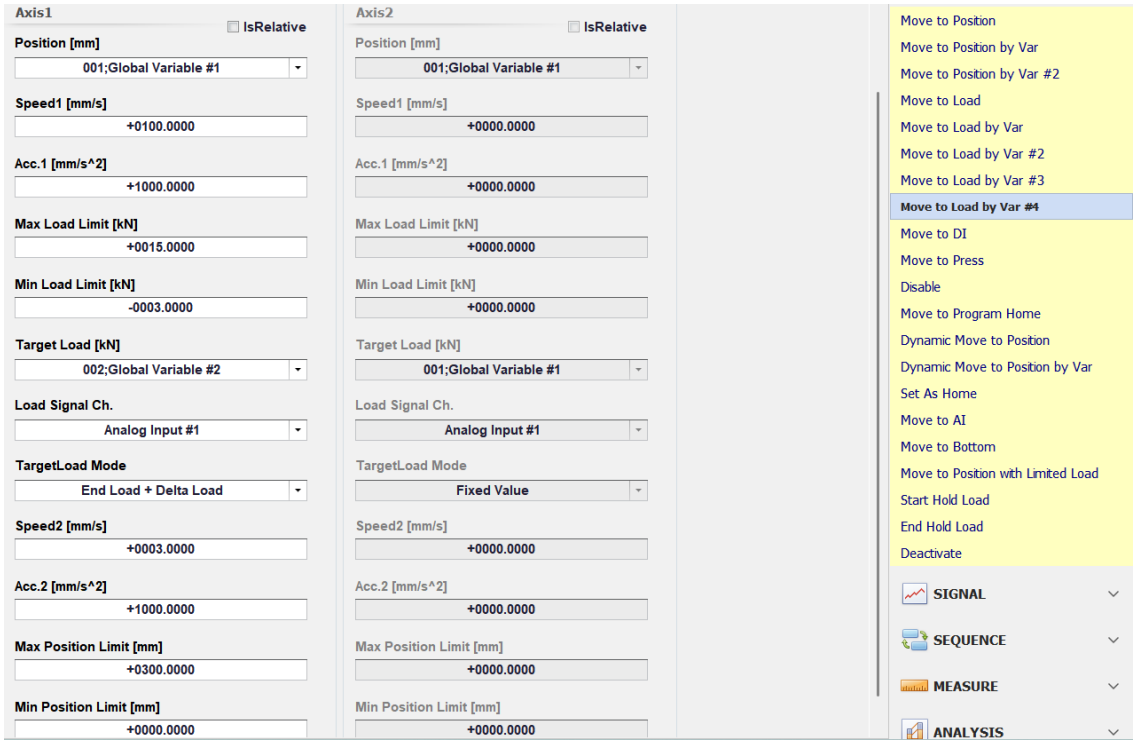


1. **Description:** It is working like “Move to Load” or “Move to Load by Var”. But you can specify the exit condition in more detail with two options which are “Target Load Tol.” and “Max Delta Dist.”. When one of

two turns out to be satisfied, OACIS will terminate this function.

2. **Parameters:**
   - "**Move On to Next Step with Position Limit**"
     - ➢ Move on to Next Step with Position Limit
     - ➢ With this option, the cycle will move on to next step with no stop even if it fails to reach the target load. If you want to move the selected Axis to the target load or the position limit, this option would be useful.
   - **"Error and Stop with Position Limit"**
     - ➢ Error and Stop with Position Limit
     - ➢ With this option, it is working like "Move to Load". It means, if the cycle fails to reach the target load, it stops at the position limit with "Error" signal.
   - **Target Load**: Target Load to reach / [kN] or [Nm]
   - **Target Load Ch**: Analog Signal Input Channel for motion control. It is recommended to use default Analog Input Signal Channel.
   - **Holding Time**: OACIS holds on the Target Load for the duration / [sec].
   - **Speed**: Speed to reach at Target Load / [mm/s] or [deg/s]
   - **Acceleration**: Acceleration to reach at Target Load / [mm/s^2] or [deg/s^2]
   - **Max Position Limit / Min Position Limit**: If the OACIS fail to reach at the target load within the specified position limit, OACIS stop the operation / [mm] or [deg].
   - **Target Load Tolerance.:** When the OACIS reaches the range within "Target Load +/- Target Load Tolerance", it terminates this function / [kN] or [Nm]. Target Load Tolerance is an absolute value.
   - **Max Delta Distance.:** If OACIS arrives at the +/- relative position of Max Delta Distance from the get-go, it exits the function / [mm] or [deg]. Max Delta Distance is an absolute value.

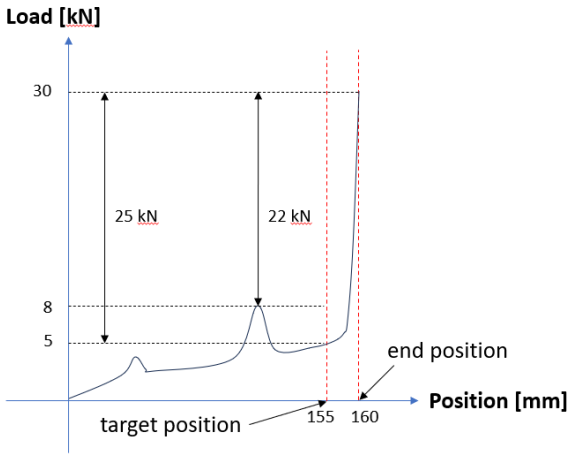
H. Move to Load by Var #4

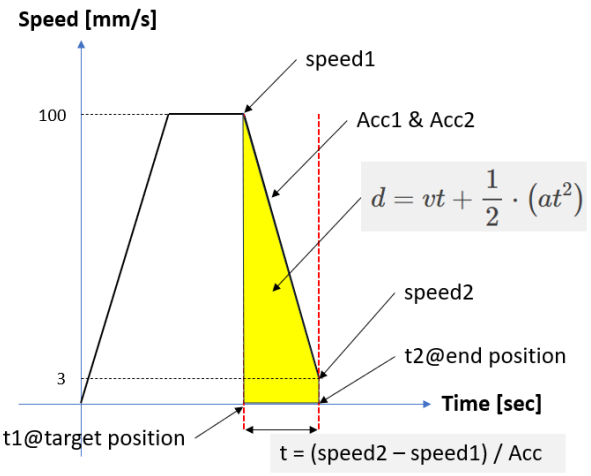


1. **Description:** It is working like "Move to Position by Var" and then "Move to Load by Var". But the biggest difference from them is that it goes directly to "Move to Load" from Position at Speed1 which means there is no downtime between these two functions. To fully satisfy all parameter's conditions, users need to

consider if it is physically feasible condition. The function is useful for bottoming applications.

2. **Parameters:**
   - "**Move On to Next Step with Position or Load Limit**"
     - ➢ Move On to Next Step w/ Position or Load Limit
     - ➢ With this option, the cycle will move on to next step with no stop even if it fails to reach the target position or target load.
   - **"Error and Stop with Position or Load Limit"**
     - ➢ Error and Stop w/ Position or Load Limit
     - ➢ With this option, it will stop and show an error when the cycle fails to reach the target position or target load.
   - **IsRelative**: If checked, OACIS moves relative to its current position.
   - **Position**: Target position to reach / [mm].
   - **Speed1**: Speed to approach to the position / [mm/s] or [deg/s]
   - **Acceleration1**: Acceleration to approach to the position / [mm/s^2] or [deg/s^2]
   - **Max Load Limit / Min Load Limit**: If the load runs out of load limits while approaching the position, OACIS stops and goes into error state or jumps into next step on the option button / [kN], [N] or [kgf].
   - **Target Load**: The meaning can be changed depending on target load mode. It means final load for Fixed Value but delta load for Max Load + Delta Load or End Load + Delta Load.
   - **Target Load Ch**: Analog Signal Input Channel for motion control. It is recommended to use default Analog Input Signal Channel.
   - **Target Load Mode**:
     - ➢ **Fixed Value:** When you select this mode, target load means final load.
     - ➢ **Max Load + Delta Load:** If you select this mode, final load is decided by calculating max load during move to target position + target load which means delta load.
     - ➢ **End Load + Delta Load:** If you select this mode, final load is decided by calculating end load at the target position + target load which means delta load.
   - **Speed2**: Speed to approach to the target load / [mm/s] or [deg/s]
   - **Acceleration2**: Acceleration to approach to the target load / [mm/s^2] or [deg/s^2]
   - **Max Position Limit / Min Position Limit**: If the position runs out of position limits while approaching the target load, OACIS stops and goes into error state or jumps into next step on the option button / [mm]
3. **Example**





- Given the above bottoming graph, you can make use of move to load by var#4 according to target load mode.

Target Load Mode:

- ➢ **Fixed Value**

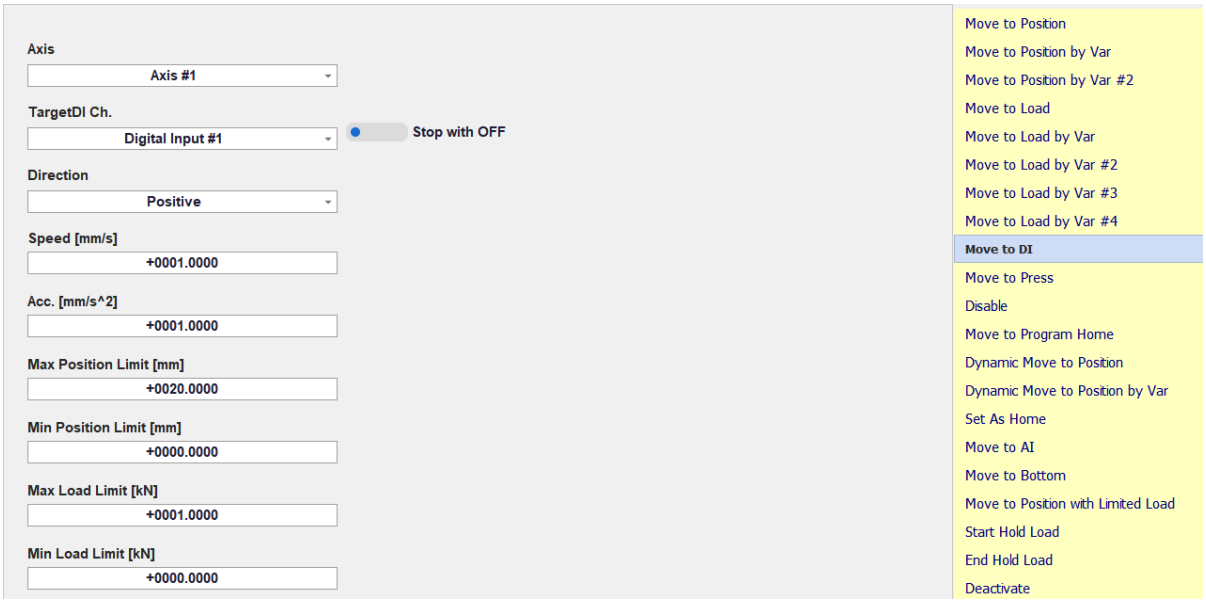
: When you want to control OACIS to 155 mm target position at 100 mm/s speed1 and then move to 30 kN target load at 3 mm/s speed2 regardless of the load at the target position, you can select this mode.

- **Max Load + Delta Load**
  : If you want to control OACIS to 155 mm target position at 100 mm/s speed1 and then move to 8 kN max load during move to target position + 22 kN target load which means delta, this mode can be good choice.
- **End Load + Delta Load**
  : If you want to control OACIS to 155 mm target position at 100 mm/s speed1 and then move to 5 kN end load at the target position + 25 kN target load which means delta load, this mode can work.

⚠ **Cautions**

When you use this function, you need to consider if it is physically feasible condition. If you want to reduce speed from 100 mm/s speed1 to 3 mm/s speed2 with 1000 mm/s^2 acceleration, then you need to put target position at over 5 mm away before end position. It is because it takes at least 0.097 sec = (3 – 100) / -1000 to arrive at speed2 from speed1. You have to assign the distance between target position and end position at least by over 4.9955 mm = 100 x 0.097 + ½ (-1000 x 0.097^2)

I. Move to DI

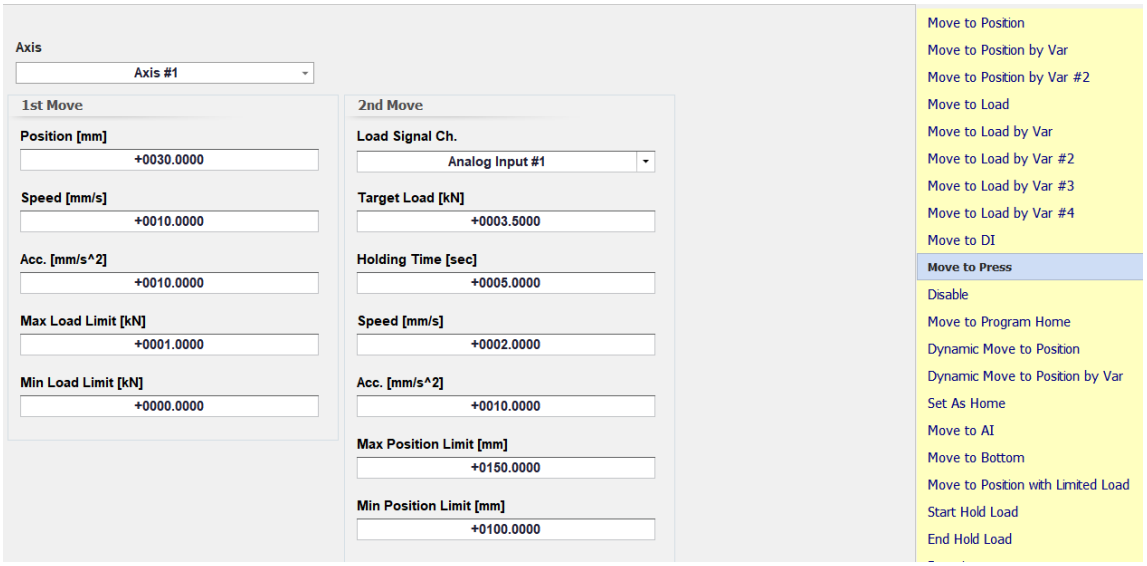


1. **Description:** Move the selected Axis until the target Digital Input signal turns on or off.
2. **Parameters:**
   - **Axis**: Axis to move
   - **Target DI**: Digital Input Channel to stop the motion
   - **Stop Condition**: If you select [ON], the axis moves until the Digital Input signal turns on. If the signal is on when the motion is about to move, OACIS skip the step / [ON] or [OFF].
   - **Direction**: The direction of motion. If you choose “Positive”, the axis moves with positive direction to find stop condition / [Positive] or [Negative].
   - **Speed**: [mm/s] or [deg/s]
   - **Acceleration**: [mm/s^2] or [deg/s^2]
   - **Max Position Limit / Min Position Limit**: If the OACIS fail to reach at the target load within the specified position limit, OACIS stop the operation and send an error message / [mm] or [deg].
   - **Max Load Limit / Min Load Limit**: If the load is higher than this limit, OACIS stop the operation

and send an error message / [kN] or [Nm].

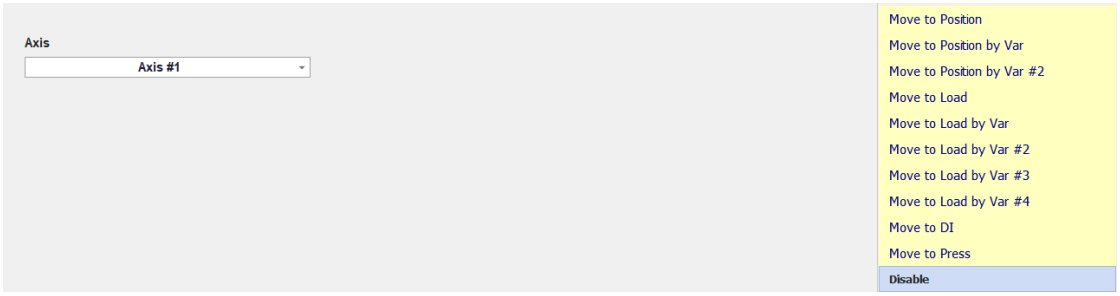
- **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

J. Move to Press



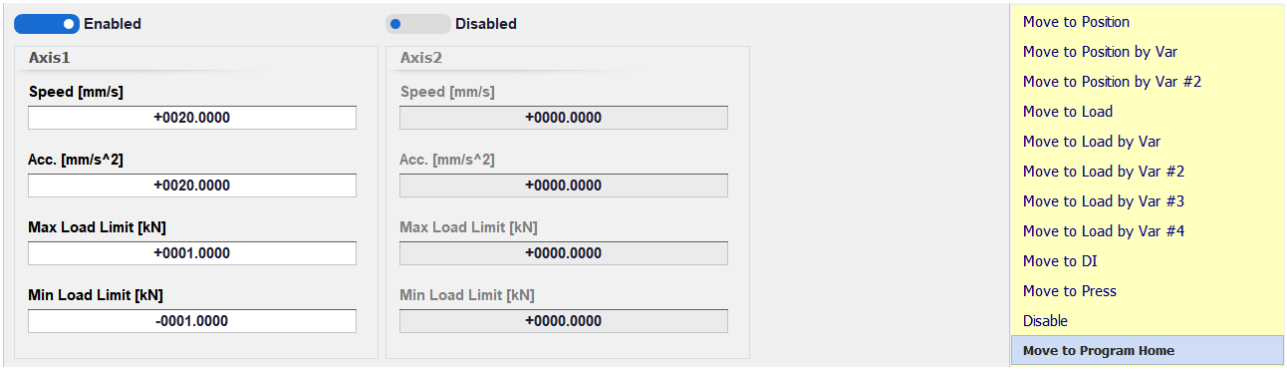
1. **Description:** It is special function for press operation combined with “Move to Position” and “Move to Load”. First Stroke is same as “Move to Position” and Second Stroke is same as “Move to Load”.

K. Disable



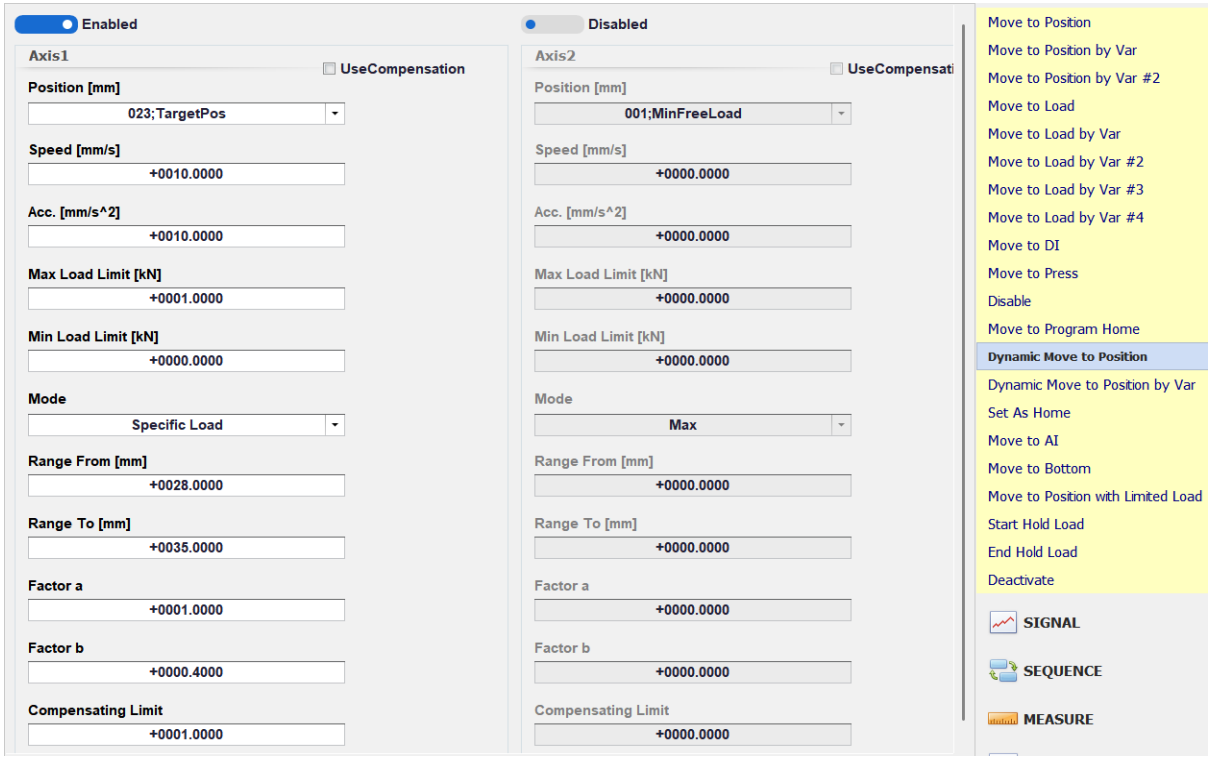
1. **Description:** Disable the selected Axis until the next moving function happens. This function is useful to protect the Axis from unexpected external impact or interference. It is different from “Power Off”. Even you disable the axis, the OACIS is still monitoring the position. So when you enable the axis, you do not need to initialize the OACIS.
2. **Parameters:**
   - **Axis**: Axis to be disabled.
   - **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

L. Move to Program Home



1. **Description:** Move the selected Axis to the Program Home that is specified in the Program Configuration. Multi Axes can be selected. This function is useful to optimize axis travel. When you start Program, the Axes should be positioned at Home Position or Program Home Position. And you can save cycle time with reasonable Program Home Position.
2. **Parameters:**
   - **Speed**: [mm/s] or [deg/s]
   - **Acceleration**: [mm/s^2] or [deg/s^2]
   - **Max Load Limit / Min Load Limit**: If the load is higher than this limit, OACIS stop the operation and send an error message / [kN] or [Nm].
   - **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

M. Dynamic Move to Position

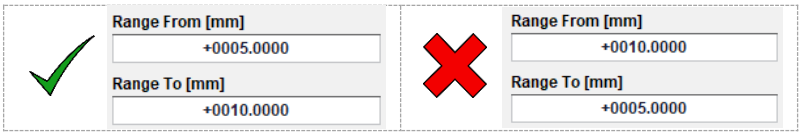


1. **Description:** It is same as “Move to Position” except that the OACIS is compensating target position depending on the running load. Multi Axis can be selected.
2. **Parameters:**
   - **UseCompensation**: If checked, OACIS applies one of several modes.

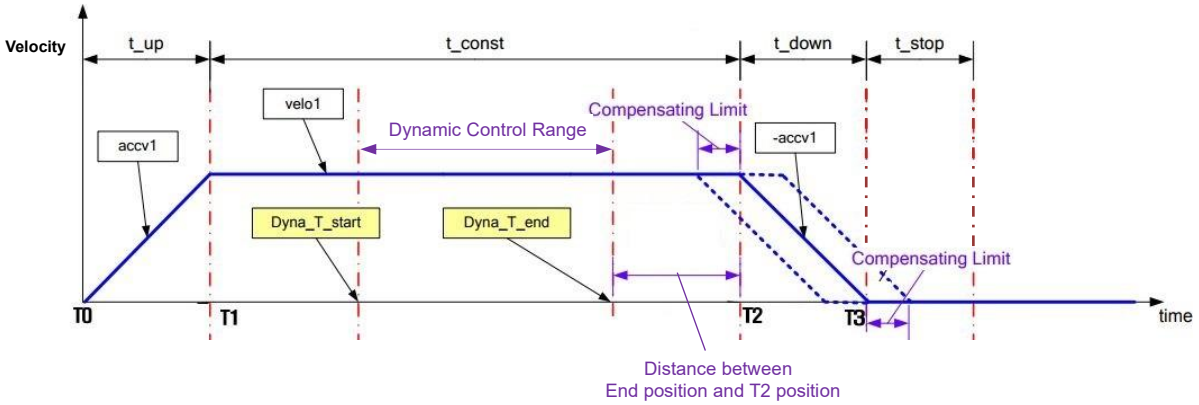
- **Position**: Target Position to move. Select a Global Variable as a Target Position / [mm] or [deg].
- **Speed**: [mm/s] or [deg/s]
- **Acceleration**: [mm/s^2] or [deg/s^2]
- **Max Load Limit / Min Load Limit**: If the load is higher than this limit, OACIS stop the operation and send an error message / [kN] or [Nm].
- **Mode**: How to find a compensating coefficient or a actual final position
  - **Max**: Capture maximum value in the range as a compensating coefficient.
  - **Min**: Capture minimum value in the range as a compensating coefficient.
  - **Ave**: Capture average value in the range as a compensating coefficient.
  - **Slope1**: Capture slope calculated by using starting point and end point of the range as a compensating coefficient.
  - **Slope2**: Capture slope calculated by using the lowest point and the highest point in the range as a compensating coefficient.
  - **Specific Load**: The actual final position = position correspoding to “a” load + “b”. Increase Acc. when b value is very small.
- **Range**: The range to get a compensating parameter.
- **“a” and “b”:** Compensating value = **“a” × “Compensating Coefficient” + “b”**
  - **So the actual final position = Position + Compensating Value**
  - The compensating coeffienct is to be specified by “Mode” selection.
- **Compensating Limit**: If the compensating value is bigger than the limit, OACIS compensate by this limit.
  - For Example: If the calculated compensating value is 1.5, but the limit is 1.0, actual compensating value would be 1.0.
- **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

⚠ Cautions

1) In the dynamic control rage setting, the end position of dynamic control must be far from the position of start.



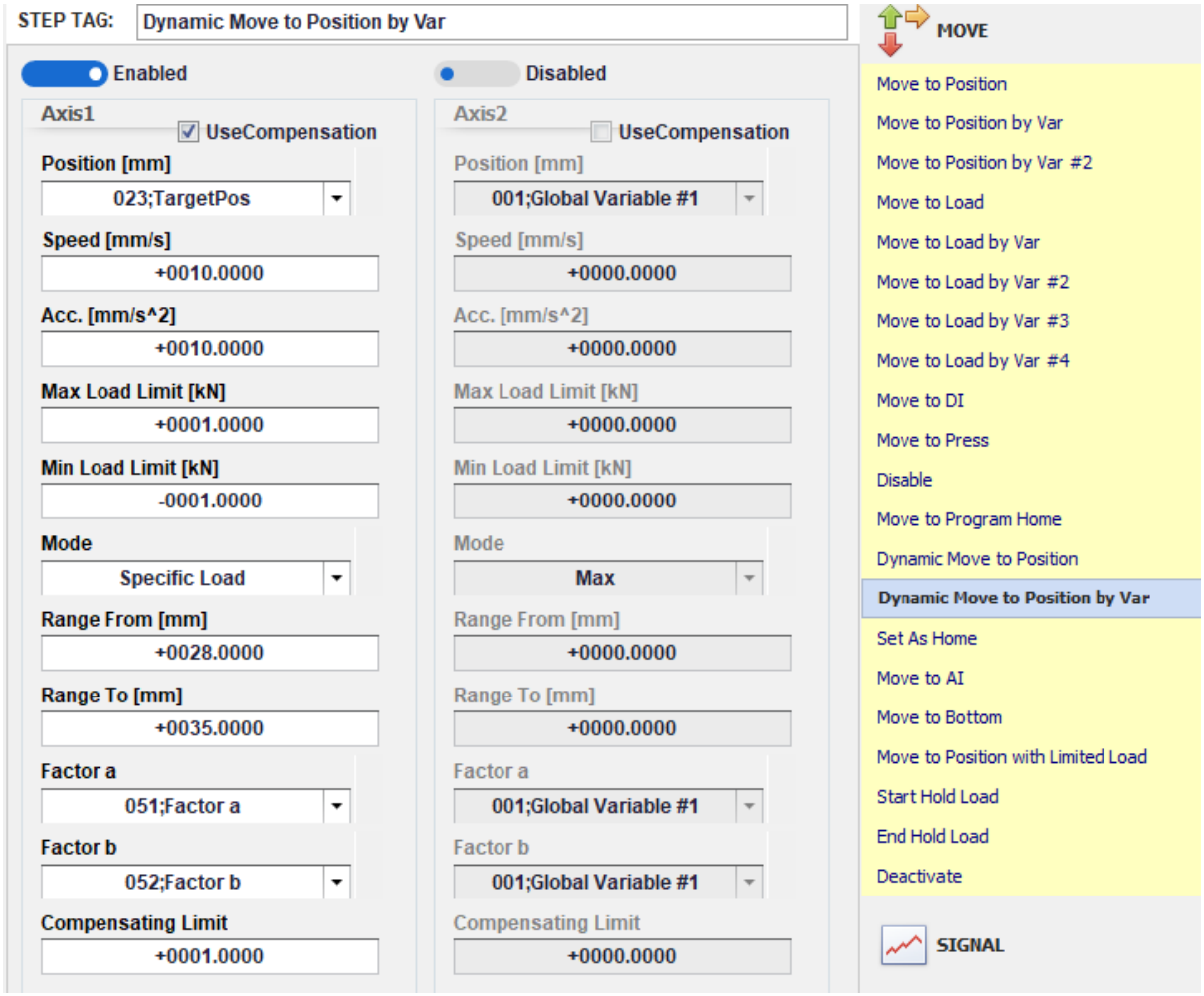
2) It is necessary to set the movement distance, speed, acceleration value of the command so that it becomes a trapezoidal trajectory with constant velocity section.



3) The dynamic control range must be set only within a constant velocity section (Between T1 and T2).
4) The compensation limit is smaller than distance between the end position of dynamic range and T2 position.

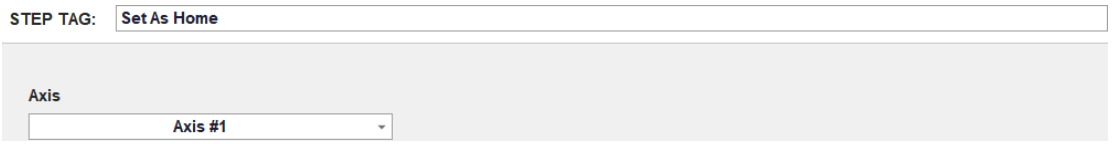
✓ If you have any problem about this function, please do not hesitate to contact us.

N. Dynamic Move to Position by Var



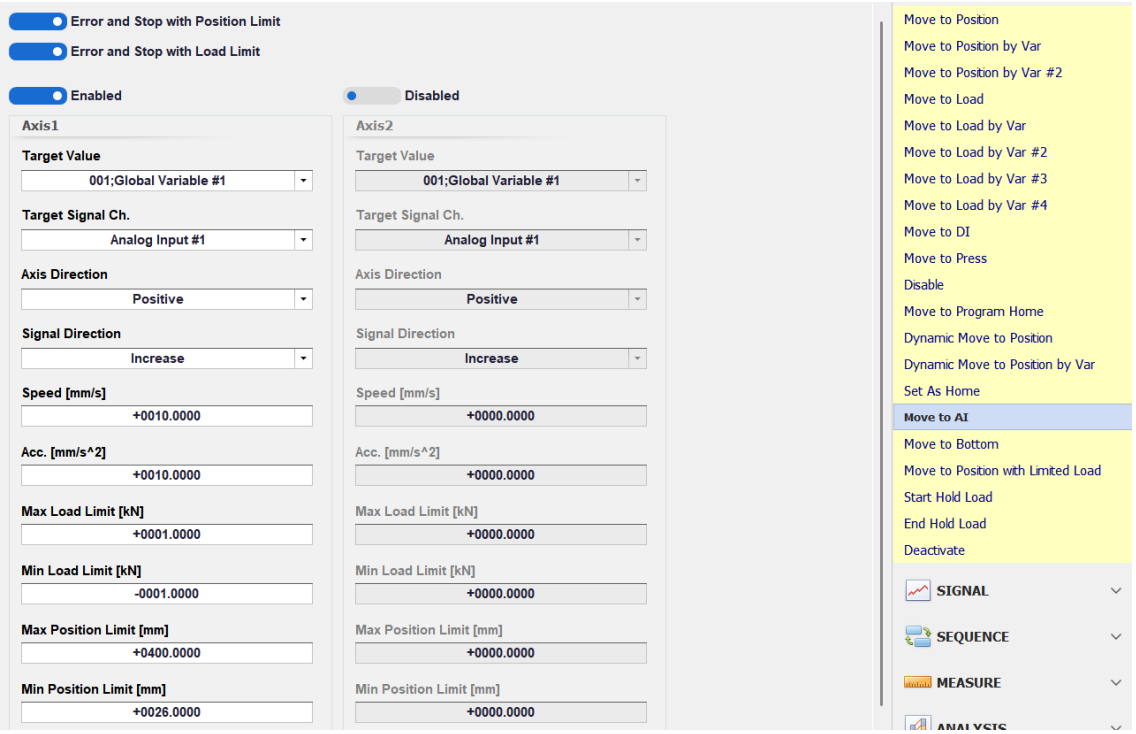
1. **Description:** It is same as “Dynamic Move to Position by Var” except that users can assign Factor a and Factor b as global variables.

O. Set As Home



1. **Description:** Set the current position of the selected Axis as Home Position. It is allowed only for homeless type Nut Runner. You can set the current position as Zero by using “Set AI or Position” function as well. But if you use the “Set AI or Position” function, new position will be valid in the cycle. So it is working differently from “Set AI or Position” function.
2. **Parameters:**
   - **Axis**: Axis to be set as Home
   - **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

P. Move to AI

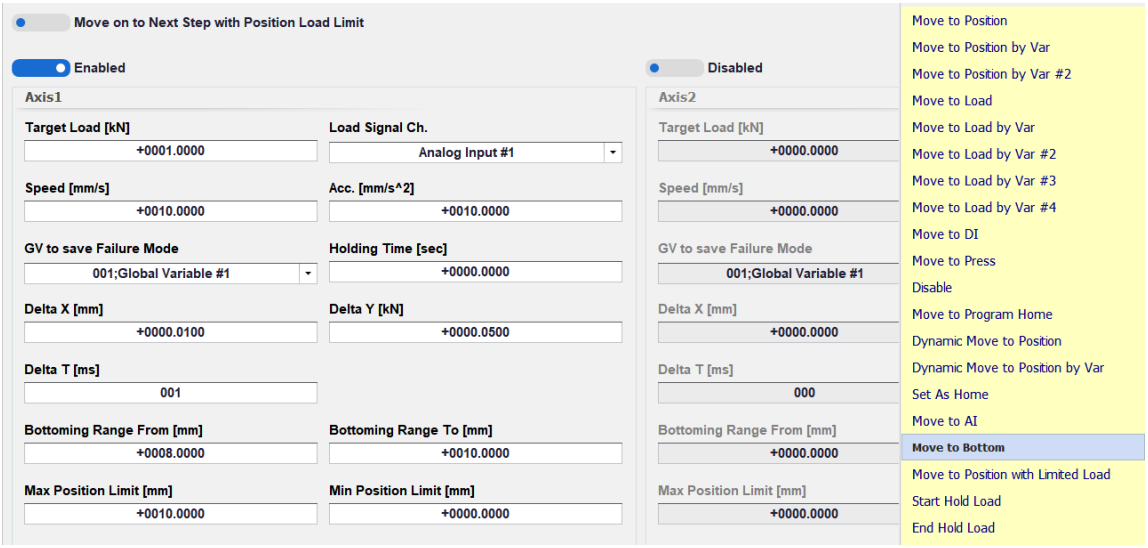


1. **Description:** Move the selected Axis to reach at the target value of the selected Analog Input Signal Channel. Multi Axes can be selected.
2. **Parameters:**
   - **"Move On to Next Step with Position Limit"**
     - ➢ Move on to Next Step with Position Limit
     - ➢ With this option, the cycle will move on to next step even if it fails to reach at the target value. If you want to move the selected channel to the target value **or** the position limit, this option would allow you to do that.
   - **"Error and Stop with Position Limit"**
     - ➢ Error and Stop with Position Limit
     - ➢ With this option, it is working like "Move to Load". It means, if the cycle fails to reach at the target value, it stops at the position limit with "Error" signal.
   - **"Move On to Next Step with Load Limit"**
     - ➢ Move on to Next Step with Load Limit
     - ➢ With this option, the cycle will move on to next step even if it fails to reach at the target value like "Move On to Next Step with Position Limit". If you want to move the selected channel to the target value **or** the load limit, this option would allow you to do that.
   - **"Error and Stop with Load Limit"**
     - ➢ Error and Stop with Load Limit
     - ➢ With this option, it is working like "Move to Position". It means, if the cycle fails to reach at the target value, it stops at the load limit with "Error" signal.
   - If you select these limit buttons in "Move to AI", they are equally applied to all the selected Axes.
   - **Target Value**: Target value to move the Axis. The target value is to be assigned by selected global variable.
   - **Target AI Ch**: Analog Input Signal Channel for motion control. It is not recommended to use default Analog Input Signal Channel. (For default channel, please use "Move to Load" function).
   - **Direction**: The direction of motion. If you choose "Positive", the axis moves with positive direction to find stop condition / [Positive] or [Negative].
   - **Signal Direction**: In case that the starting value is lower than target value, "Increase" is to be

selected. In the other case, “Decrease” is to be selected / [Increase] or [Decrease].

- **Speed**: [mm/s] or [deg/s]
- **Acceleration**: [mm/s^2] or [deg/s^2]
- **Max Load Limit / Min Load Limit**: If the load is higher than this limit, OACIS stop the operation and send an error message / [kN] or [Nm].
- **Max Position Limit / Min Position Limit**: If the OACIS fail to reach at the target load within the specified position limit, OACIS stop the operation and send an error message / [mm] or [deg].
- **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

Q. Move to Bottom

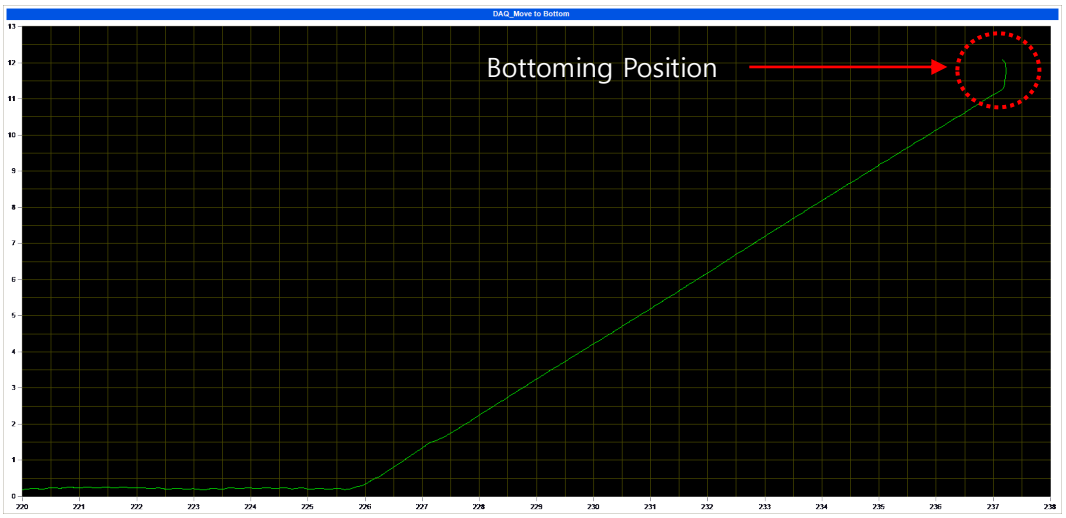


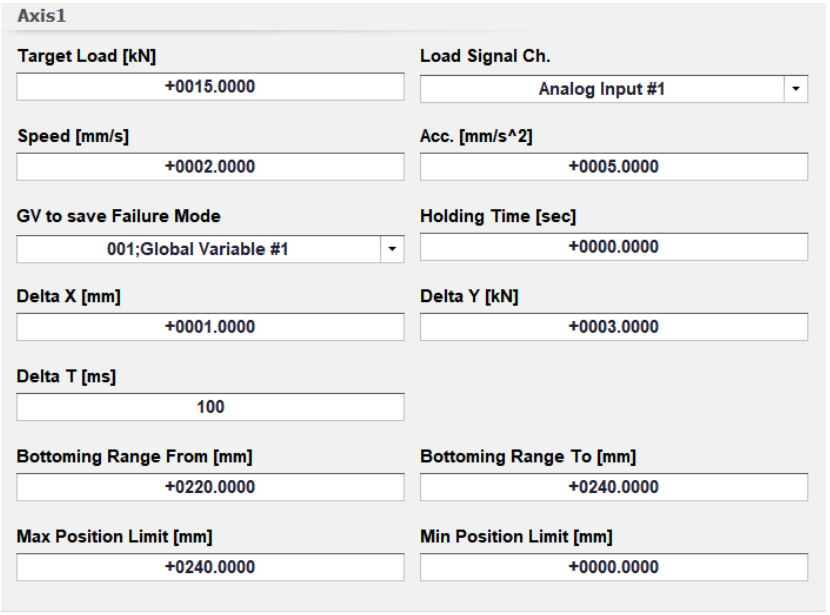
1. **Description:** It moves the selected Axis to the specified bottoming condition.
2. **Parameters:**
   - **Move On to Next Step with Position Limit Load Limit:** OACIS moves on to next step even though the cycle failed to find bottoming condition before reaching at the target load or within the position limits. Otherwise, the cycle will stop at the end of the step with “Error” signal. (Refer to the “Move to Load by Var”)
   - **Target Load**: Target Load to move. If it finds bottoming condition, it stops at the position before reaching at the Target Load / [kN] or [Nm].
   - **Target Load Ch**: Analog Signal Input Channel for motion control. It is recommended to use default Analog Input Signal Channel.
   - **Speed**: [mm/s] or [deg/s]
   - **Acceleration**: [mm/s^2] or [deg/s^2]
   - **GV to Save Failure Mode**: It saves detailed information to the selected global variable as below per the failure mode.
     - **1:** Ok. OACIS found the bottoming condition before reaching at the target load within position limits.
     - **2:** OACIS failed to find bottoming condition and reached at the target load within position limits.
     - **3:** OACIS failed not only to find bottoming condition but also reach at the target load within position limits.
   - **Holding Time**: OACIS holds on the bottoming position for the duration / [sec].
   - **Delta X [mm or deg] / Delta Y [kN or Nm] / Delta T [ms]**
     - Bottoming Condition is to be defined by three parameters, Delta X, Delta Y and Delta T. If the actual “**δy/δx**” is greater than the specified “Delta Y / Delta X” in longer time span than the specified “Delta T”, the OACIS stops the Axis with recognition of bottoming.
   - **Bottoming Range:** OACIS finds “Bottoming Condition” only in this range. It helps you not to stop

the Axis in the unexpected area with unexpected load spike / [mm] or [deg].

- **Max Position Limit / Min Position Limit**: Position limits for the step.

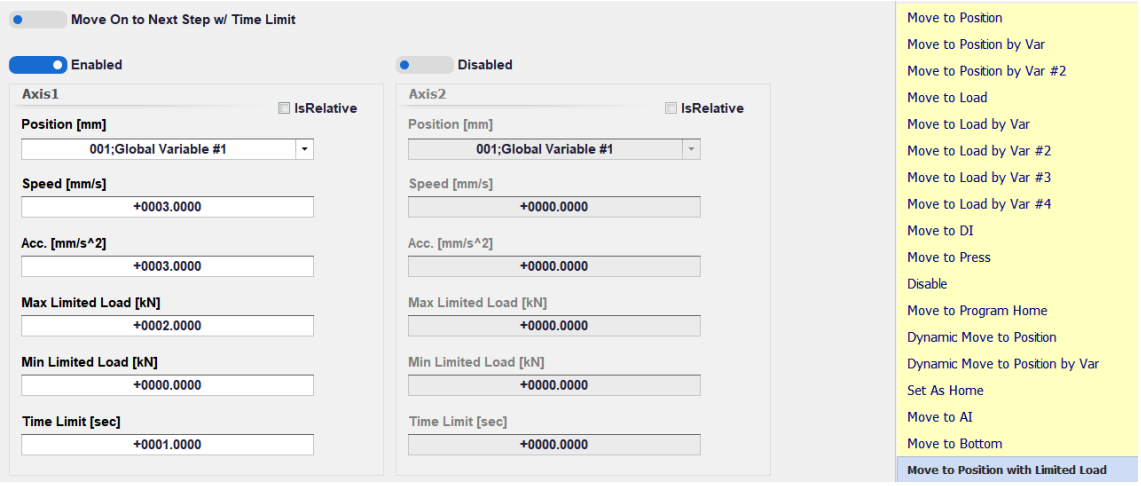
3. **Example:**





Here you set the Bottoming Condition = 3/1 (kN/mm) with longer than 100ms time span.
And you see the above curve that shows how OACIS stops the Axis at the Bottoming Condition.

R. Move to Position with Limited Load



1. **Description:** It is working like “Move to Position” or “Move to Position by Var”. But you can specify “Limited Load” and “Time Limit”. “Limited Load” is different from “Load Limit”. When the Axis meets the “Limited Load” before it reaches the target Position, OACIS is to hold the Axis at the “Limited Load”. And if the load decreases (or increases depending on the “Max Limited Load” or “Min Limited Load”), OACIS starts to move the Axis to the target Position. If the holding time at the “Limited Load” is longer than the “Time Limit”, OACIS shows Error or move on to the next step depending on “Error and Stop with Time Limit” option.

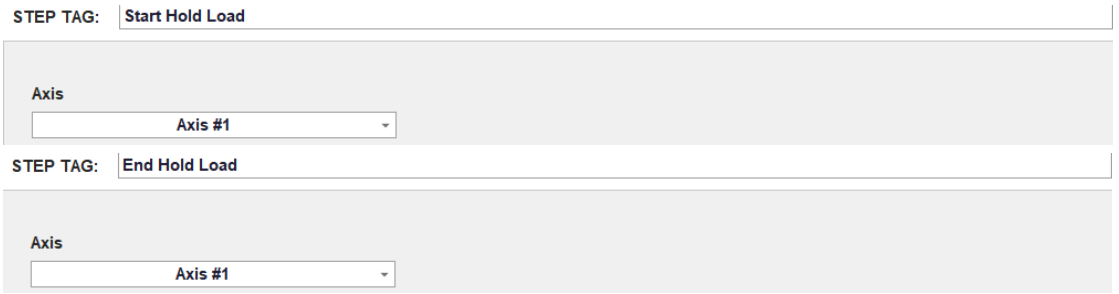
Note: The function can be used only if the moving direction is the same as the load direction. When the Axis moves forwards, the load should be increased.

2. **Parameters:**
   - “**Move On to Next Step with Time Limit**”
     - 

     - With this option, the cycle will move on to next step even if it fails to reach at the target position in the Time Limit.
   - **“Error and Stop with Position Limit”**
     - 

     - With this option, OACIS stops and shows Error at the step when it fails to reach at the target position with “Time Limit”.
   - **IsRelative**: If checked, OACIS moves relative to its current position.
   - **Position**: Target Position / [mm] or [deg].
   - **Speed**: Speed to reach at the Position / [mm/s] or [deg/s]
   - **Acceleration**: Acceleration to reach at the Position / [mm/s^2] or [deg/s^2]
   - **Max Limited Load / Min Limited Load**: The Axis moves between these Limited Loads / [kN] or [Nm]. When the Axis is moving forwards, OACIS will hold the Axis at the “Max Limited Load” during the Time Limit. On the contrary, if the Axis is moving backwards, OACIS will hold at “Min Limited Load”
   - **Time Limit**: If OACIS meets the “Limited Load” before the Position, OACIS is keeping the load until the Load decrease (or increase) during the “Time Limit” / [sec].

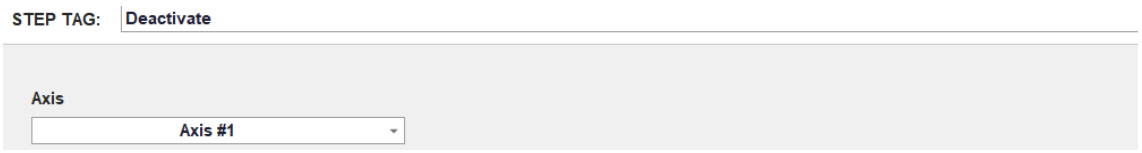
S. Start Hold Load / End Hold Load



1. **Description:** OACIS is normally controlling position except for “load control function” like “Move to Load”. But if you want to keep specific load while OACIS is doing non-motional function such as “Analysis” or “Sequence”, you can use these functions.
2. **Hold Load Start Condition:** All move functions keep basically “Hold Position” mode after finishing the functions. But “Hold Load” mode starts exceptionally under the condition below.
   - When a “Start Hold Load” function is executed and then one of load-related functions gets normally terminated without any errors, OACIS is controlling “load” instead of “position” according to parameter conditions of the load function such as target load, speed, Acc. and error limits.
   - Load-related functions include Move to Load, Move to Press, Move to Load by Var, Move to Bottom and Move to Load by Var #2.
3. **Hold Load End Conditions:** One of followings will end “Hold Load” condition. (It means it will be changed to position control mode).
   - End Hold Load
   - Position control functions (Move to Position, Move to Position by Var, …)
   - Move to Bottom: If OACIS complete it successfully (find bottoming condition), it will keep the hold load, otherwise, it will change to position control mode.
   - Abnormal Condition: OACIS also enters into “Hold Position” mode when it returns to normal from abnormal condition such as E-stop, Stop-Resume, Reset and Errors
4. **Precautions:**

   The “Hold Load” mode is not a normal condition and can cause severe problems or a dangerous situation. So you should pay special attention and need to be careful when it comes to the “Hold Load” mode

T. Deactivate



1. **Description:** Deactivate the selected Axis until the next moving function happens. This function is useful both to protect the Axis from unexpected external interference and to reduce the noise from Servo Motors. The difference between “Disable” and “Deactivate” is to get the servo enable signal On or Off. Disable function sets the output as Zero so that the servo is still enabled. However, Deactivate function sets the servo enable as off. It is also different from “Power Off”. Even you deactivate the axis, the OACIS is still monitoring the position. So when you enable the axis, you do not need to initialize the OACIS.

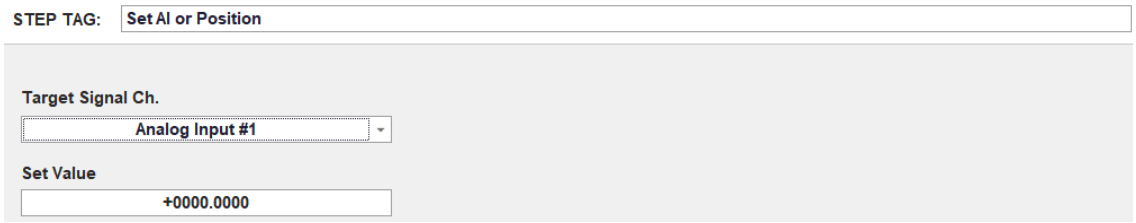
   Note: Deactivate function sets the servo enable as Off and the brake as deactivated (brake LED On). Therefore, you should be cautious about using this function especially when the tooling weighs too high.

2. **Parameters:**

- **Axis**: Axis to be deactivated.
- **Step Tag**: The step tag to be used for DAQ or Sequence control. The Tag Name should be unique.

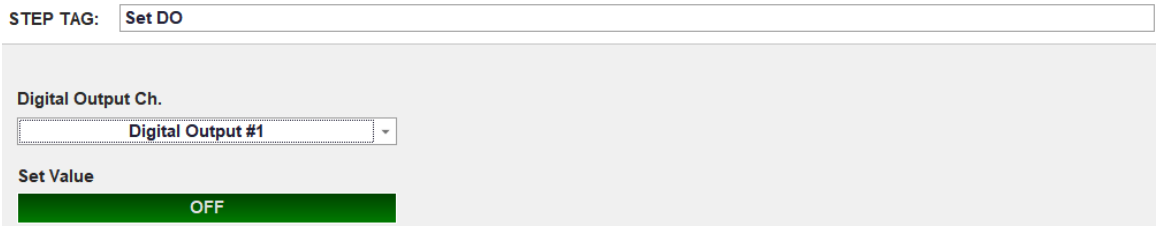
III. SIGNAL

A. Set AI or Position



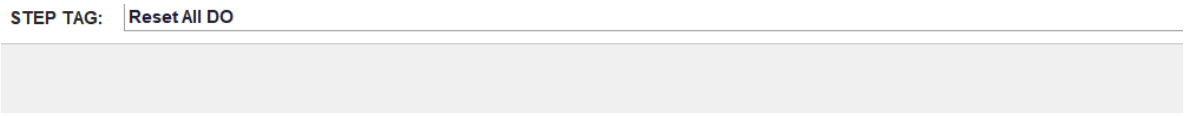
1. **Description:** Set the selected analog signal channel as the specific value. New setting value is valid till the cycle ends. After the cycle ends, OACIS sets the channel as absolute value.
   - **Note:** “Move to Program Home” function does not use offset value.
2. **Parameters:**
   - **Analog Input Ch.**: Analog Input Channel to be set as new value. One of Analog Input #1 ~ #12, RS422 Ch #1 ~ #4, Axis Position #1 ~ #4 or Encoder Input Ch #1 ~ #8.
   - **Set Value**: New value to be used for the selected Analog Input Channel.
   - **Step Tag**:

B. Set DO



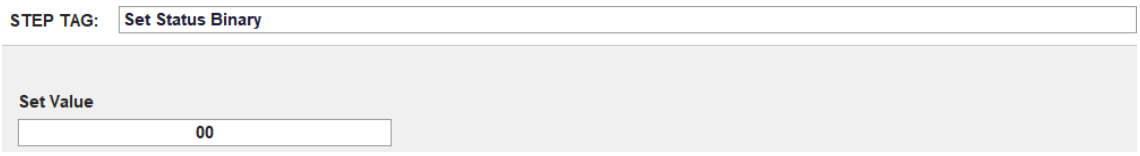
1. **Description:** Set the selected Digital Output channel as On or Off.
2. **Parameters:**
   - **Digital Output Ch.**: Digital Output Channel to be set. Programmable DO #1 ~ #14.
   - **Set Value**: [On] or [Off].
   - **Step Tag**:

C. Reset All DO



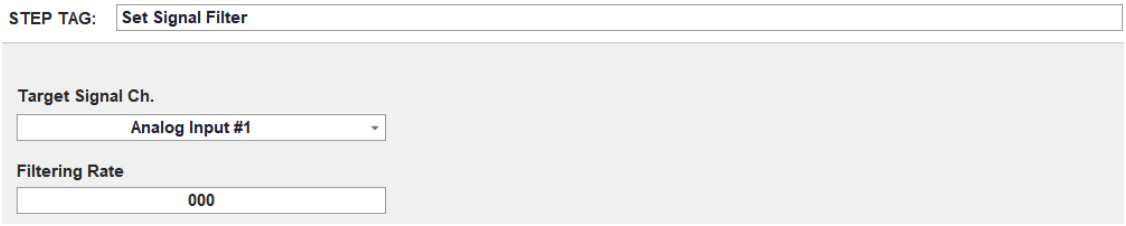
1. **Description:** Set the all Programmable Digital Output channels as Off. Normally it is used at the beginning of program to reset signal.
2. **Parameters:**
   - **Step Tag:**

D. Set Status Binary



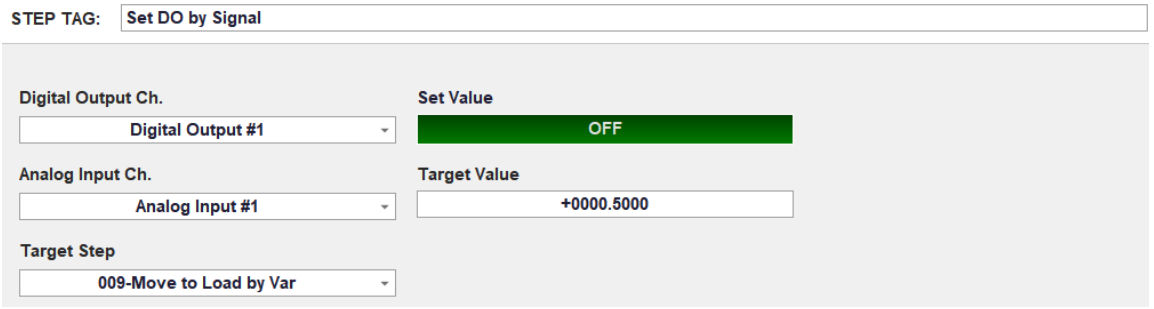
1. **Description:** Set the Status Binary as specific value. It would be used to show part status (good or bad) or failure mode to external device like PLC. If you set the value as 5, Status Bin #1 and #3 would be [On] Bin #2 and #4 would be [Off].
   - Value = Bin#1 x 1 + Bin#2 x 2 + Bin#3 x 4 + Bin#4 x 8
2. **Parameters:**
   - **Set Value**: 0 ~ 15.
     - 0: Set all [Off] / 15: Set all [On].
   - **Step Tag:**

E. Set Signal Filter



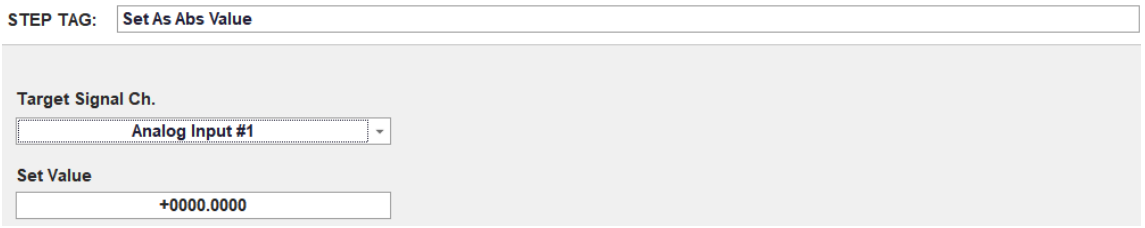
1. **Description:** Set the selected Analog Input Channel digital filter. New filtering setting would be valid until cycle ends. (After cycle ends, OACIS sets the filter as System Configuration value). It is working as digital low pass filter. The higher value would make the more stable value and the lower value would make the more sensitive value.
2. **Parameters:**
   - **Filtering Rate:** 0 ~ .
   - **Step Tag:**

F. Set DO by Signal



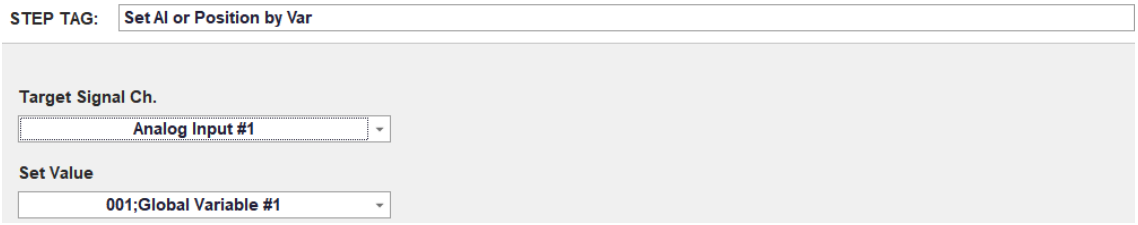
1. **Description:** Set the selected Digital Output channel as On or Off when the selected Analog Input Channel reach at the Target Value while the selected Target Step is running.
2. **Parameters:**
   - **Digital Output Ch.**: Digital Output Channel to be set. Programmable DO #1 ~ #14.
   - **Set Value**: [On] or [Off].
   - **Analog Input Ch.**: Analog Input Channel to be used as setting condition.
   - **Target Value:** Target Value to set the selected Analog Input channel.
   - **Target step:** Target Step to set the Selected DO channel as On or Off. It should be located before the selected Target Step.
   - **Step Tag**:

G. Set As Abs Value



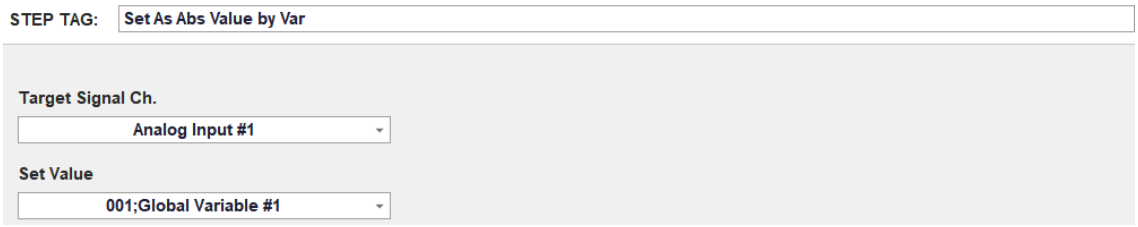
1. **Description:** Set the selected Signal Input channel as the specified set value. The Set Value will be a new absolute value. It differs from “Set AI or Position” function. “Set AI or Position” is valid only in the cycle. But it changes system configuration. OACIS will have new absolute value.
   - **Note:** You should be careful to use this function especially for the Signal Input Channels with current or voltage type signal. In some cases, the system will have accumulative errors.
2. **Parameters:**
   - **Signal Input Ch.**: Signal Input Channel to be set with new absolute value
   - **Set Value**: New absolute value to be used for the selected Signal Input Channel.
   - **Step Tag**:

H. Set AI or Position by Var



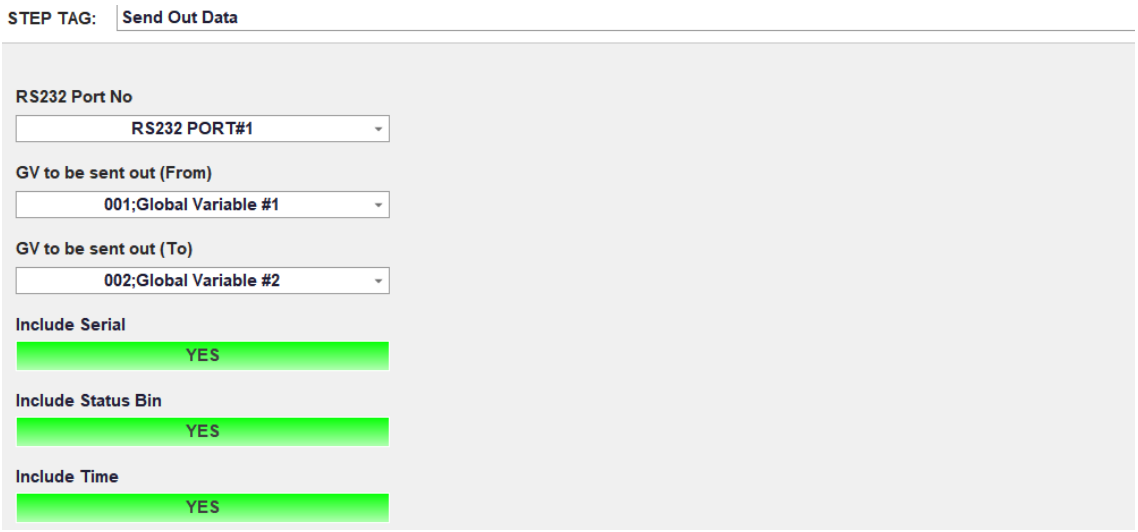
1. **Description:** This function is same as “Set AI or Position” except for that you can specify the setting value by selecting a global variable.

I. Set As Abs Value by Var



1. **Description:** This function is same as “Set As Abs Value” except for that you can specify the setting value by selecting a global variable.

J. Send Out Data



1. **Description:** You can send out the selected Global Variable data through the selected RS232 communication port.
   - **Note:** The selected RS232 Com Port needs to be configured as “Data Out” mode not “Scan In” mode. See the manual of “How to configure” for more details.
2. **Parameters:**
   - **RS232 Com Port to Send Out Data**: The Com Port is to be used to send out data
   - **GLOBAL VARIABLE to be Sent Out (From) and (To)**: If you select Global Variable #1 as “From” and Global Variable #3 as “To”, OACIS will send out Global Variable #1 through Global Variable #3.
   - **INCLUDE SERIAL No / STATUS BIN / TIME**: When you send out data, you can attached a few optional information like “Serial No”, “Status Bin” and “Time”.

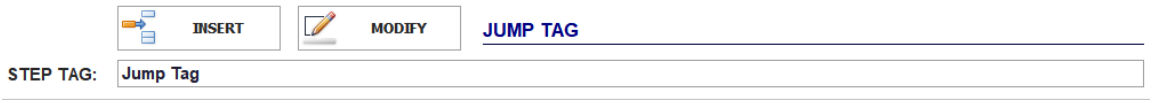
- **Note:** All the information to be sent will be the value at that step. So, you may need to consider where this step needs to be inserted to send out proper data.

- **Step Tag:**

3. **Send Out Data Format: "GV;FA;GV1;GV2;GV3…;SerialNo;StatusBin;Time;" + CR**
   - GV;FA -> Header
   - CR -> End Terminator
4. **Example:**
   - **Assumptions:**
     - GLOBAL VARIABLE to be Sent Out (From): Global Variable #1
     - GLOBAL VARIABLE to be Sent Out (To): Global Variable #3
     - All Options selected as "YES"
   - **OACIS sends out data as below:**
     - "GV;FA;+0000.0001;+0000.0002;+0000.0003;123456789;15;130328010101;" + CR
   - **If No Option selected:**
     - "GV;FA;+0000.0001;+0000.0002;+0000.0003;" + CR

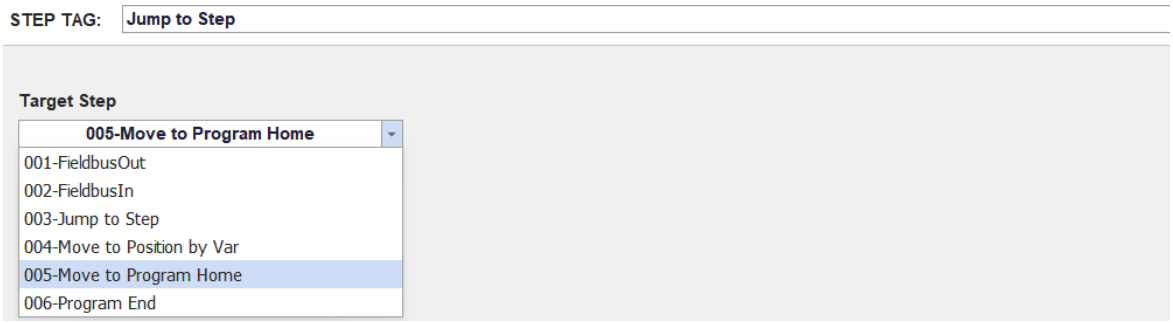
IV. SEQUENCE

A. Jump Tag



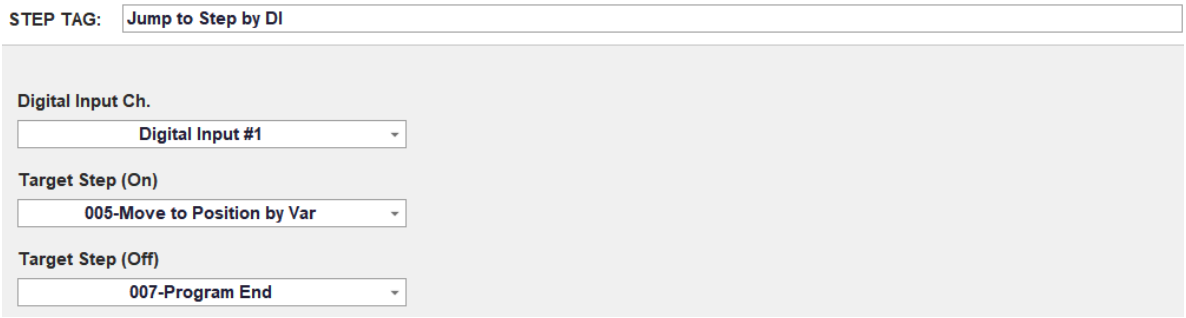
1. **Description:** The step has only tag without any function. It is to be used for target step to jump.
2. **Parameters:**
   - **Step Tag:**

B. Jump to Step

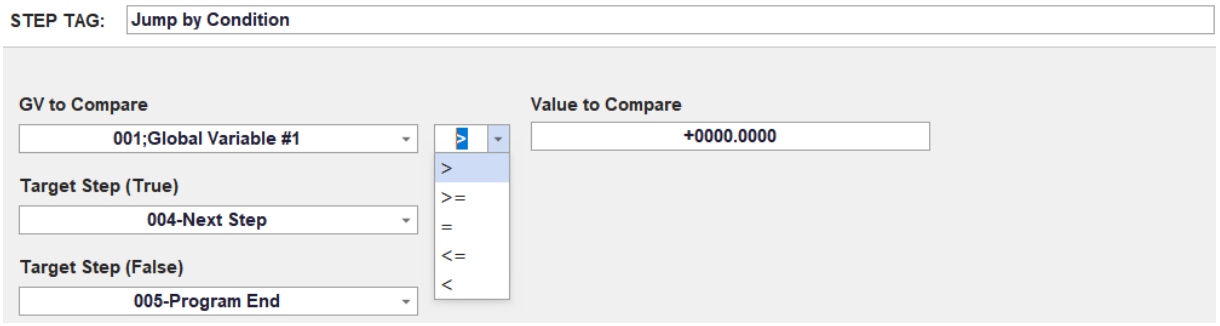


1. **Description:** Jump to the selected step
   - **Note:** If this function is used among “Loop” – “Loop End” function, a step to jump should surely be included among “Loop” function.
2. **Parameters:**
   - **Target Step:** Step to jump.
   - **Step Tag**:

C. Jump to Step by DI

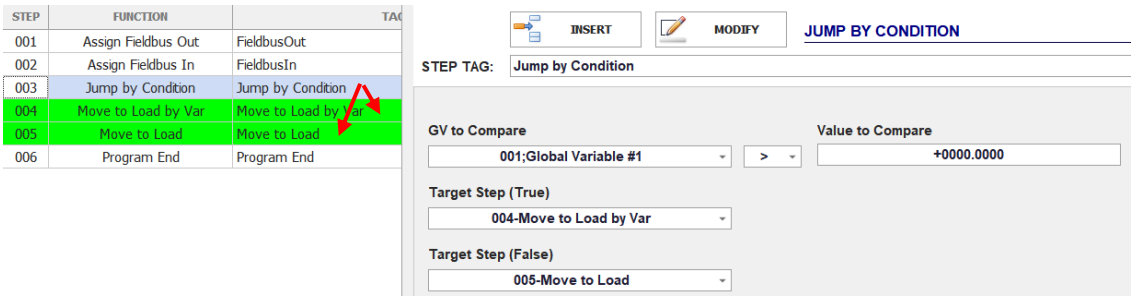


1. **Description:** Jump to the selected step by the condition of the selected Digital Input signal.
   - **Note:** If this function is used among “Loop” – “Loop End” function, a step to jump should surely be included among “Loop” function.
2. **Parameters:**
   - **Digital Input Ch.**: Digital Input Signal Channel
   - **Step to Jump (Case On):** The target step to jump in the case of the selected DI On.
   - **Step to Jump (Case Off):** The target step to jump in the case of the selected DI Off.
   - **Step Tag**:

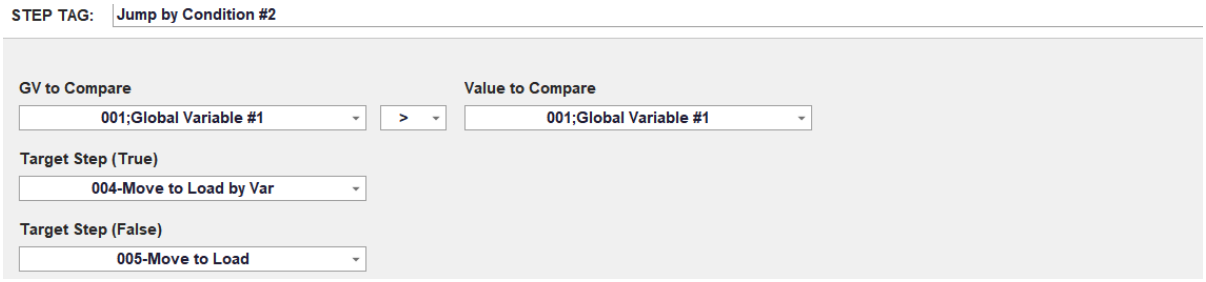


D. Jump by Condition

1. **Description:** Jump to the selected step by the condition of comparison operators.
2. **Parameters:**
   - **Global Variable to Compare:** Object to compare.
   - **Comparison Operators:** You can select one of them (>, >=, =, <=, <).
   - **Value to Compare:** Real number to compare.
   - **Step to Jump (Case True):** The target step to jump if the condition is true. If you select “Next”, it will go to the step next to “Jump by Condition”.
   - **Step to Jump (Case False):** The target step to jump if the condition is false.
   - **Step Tag:**
3. **Example**
   - If GV #1 is greater than 0 at step 3, it moves to **step 4** (Move to Position by Var). But if GV #1 is less than or equals to 0 at Step 3, it jumps to **step 5** (Move to Load).

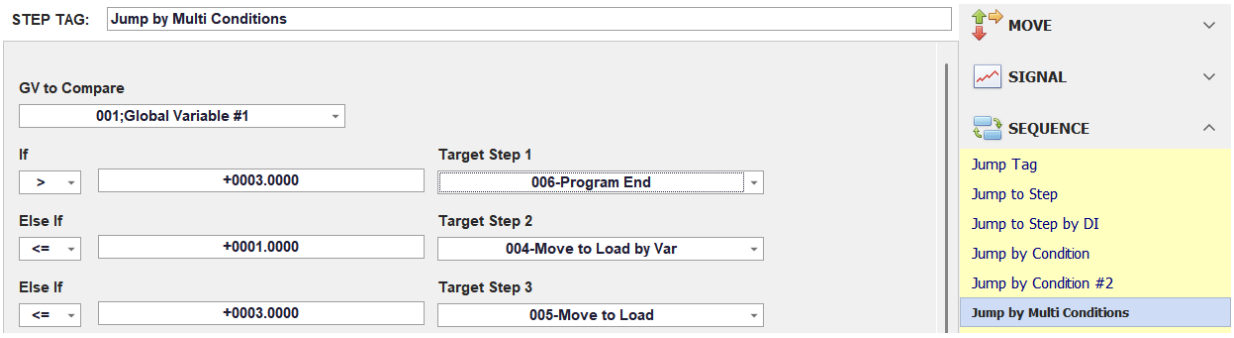


E. Jump by Condition #2

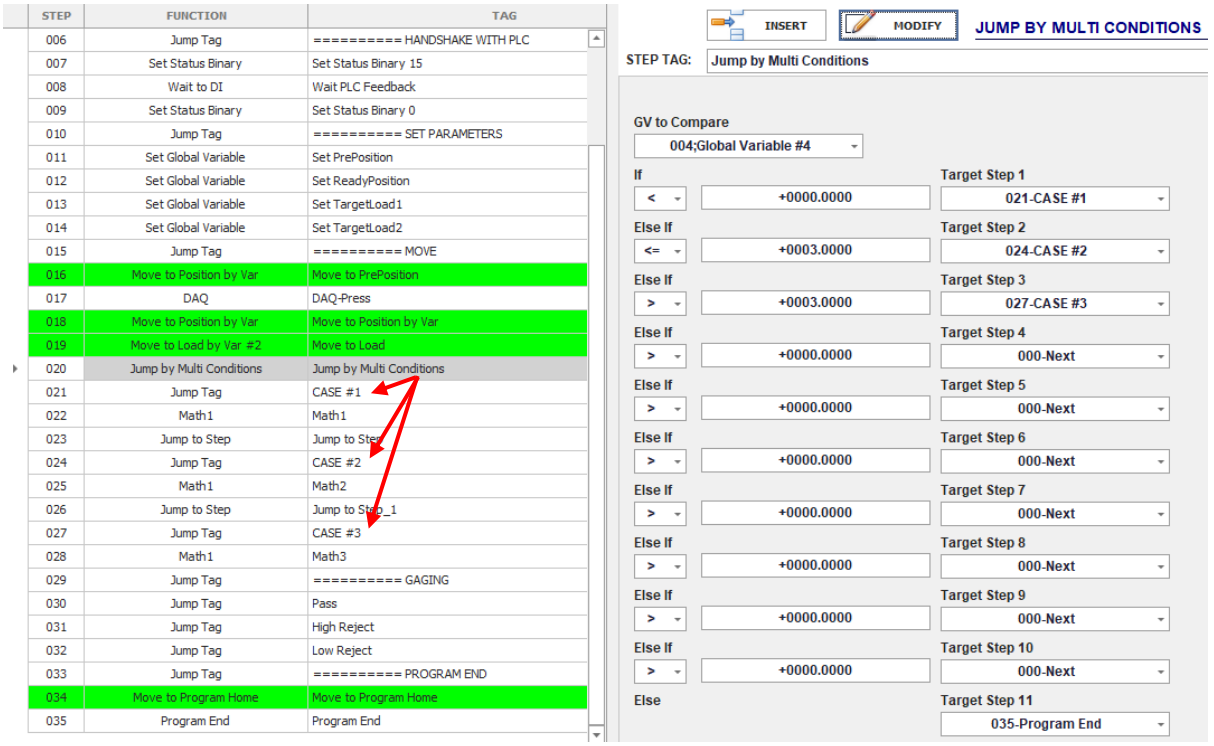


1. **Description:** It is same as “Jump by Condition” except that objects to compare are all Global Variables.
2. **Parameters:**
   - **GV #1 To Compare**: Object to compare.
   - **Comparison Operators:** You can select one of them (>, >=, =, <=, <).
   - **GV #2 To Compare:** Another object to be compared.
   - **Step to Jump (Case True):** The target step to jump if the condition is true. If you select “Next”, it will go to the step next to “Jump by Condition”.
   - **Step to Jump (Case False):** The target step to jump if the condition is false.
   - **Step Tag**:

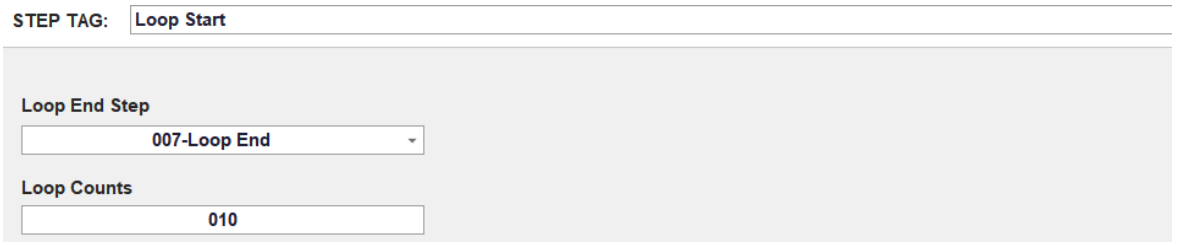
F. Jump by Multi Conditions



1. **Description:** Jump to the selected step of the condition that meets amongst multi conditions. This function moves downward from the “IF…” condition and terminates at the satisfied step.
2. **Parameters:**
   - Global Variable to Compare: Object to compare.
   - Comparison Operators: You can select one of them (>, >=, =, <=, <).
   - Value to Compare: Real number to compare.
   - Jump to…: The target step to jump if the condition is true. If you select “Next”, it will go to the step next to “Jump by Multi Conditions”. The maximum number of jump to…steps that you can assign is 11.
   - **Step Tag:**
3. **Example: See the next page.**
   - When you want to make a program including over three conditional statements as below, this function is very useful. You can assign the condition up to 11.
     - ➢ If GV #4 is less than 0 at Step 20, it jumps to step 21 (Case #1).
     - ➢ If GV #4 is less than or equals to 3 at Step 20, it jumps to step 24 (Case #2).
     - ➢ If GV #4 is greater than 3 at Step 20, it jumps to step 27 (Case #3).

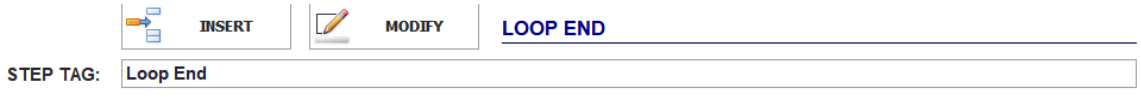


G. Loop Start



1. **Description:** It makes repeated loop by a pair with “Loop End” step. OACIS repeats the steps wrapped with “Loop Start” and “Loop End” by the specified loop cycles.
   - **Note #1:** It should be located before the paired “Loop End” step. It can’t be located after “Loop End” step.
   - **Note #2:** If there are multi loop cycles, the “Loop Start” step should be paired with the closest “Loop End” step. For example, if the first “Loop Start” is step number 3 and the second “Loop Start” is step number 4, and there are “Loop End”s at step number 10 and 11, the first “Loop Start” should be paired with step number 10 “Loop End” step.
2. **Parameters:**
   - **Loop End Tag**: Target “Loop End” step to be paired.
   - **Loop Cycles:** The repeated cycles.
     - **Note:** If you input “0”, the OACIS runs infinity loop.
   - **Step Tag:**

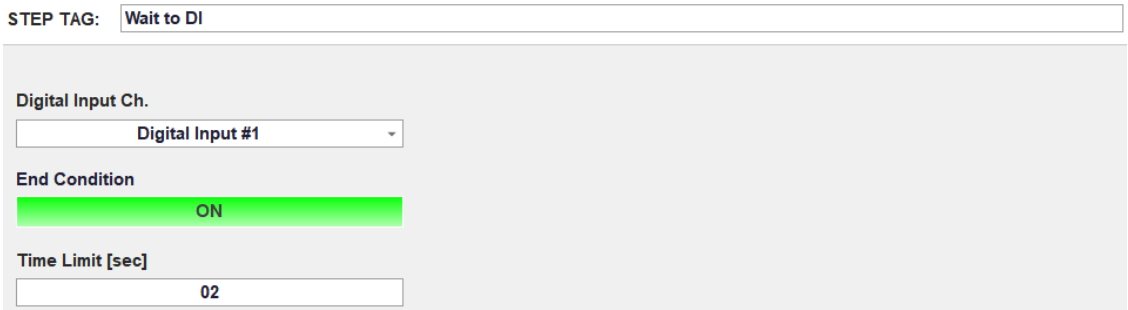
H. Loop End



1. **Description:** It makes repeated loop by a pair with “Loop Start” step. See the “Loop Start” for more details.

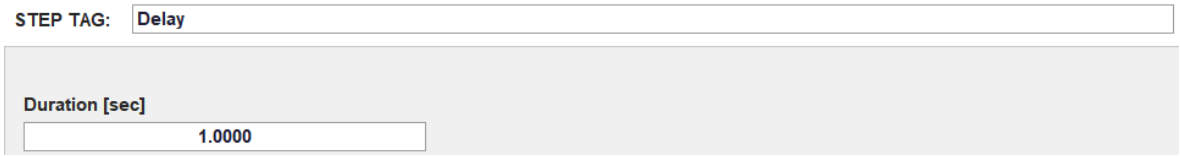
2. **Parameters:**
   - Step Tag:

I. Wait to DI



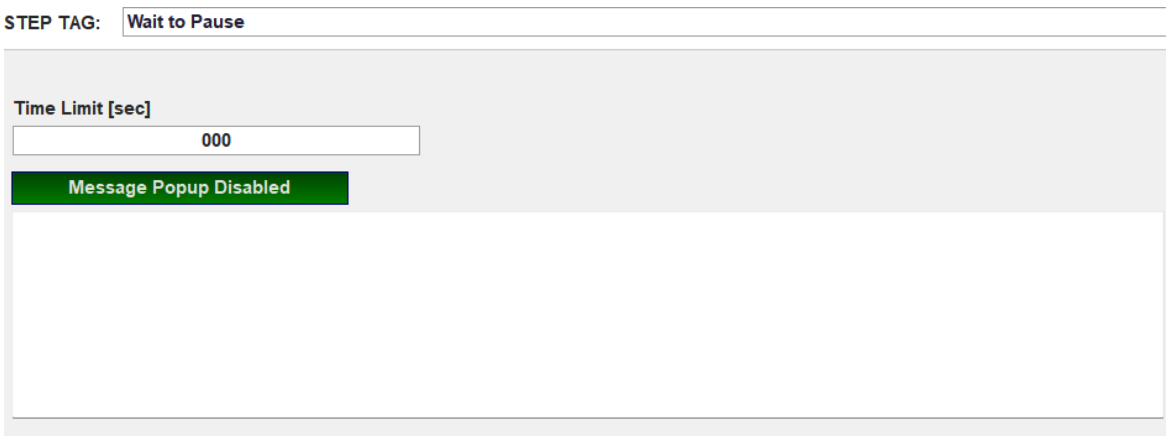
1. **Description:** Waiting for the selected Digital Input Signal. That is until a selected digital input channel have same condition, the step will wait for the condition in the present step.
2. **Parameters:**
   - **Digital Input Ch.**: Digital Input Signal Channel
   - **End Condition:** End condition to move on to next step. When the OACIS come in to this step, if the signal is already “End Condition”, then move on to the next step without waiting. [On] or [Off].
   - **Time Limit:** If there is no end condition signal within the specified “Time Limit”, the OACIS send a error message and ends program. [sec] (0 ~ 99).
   - **Step Tag**:

J. Delay



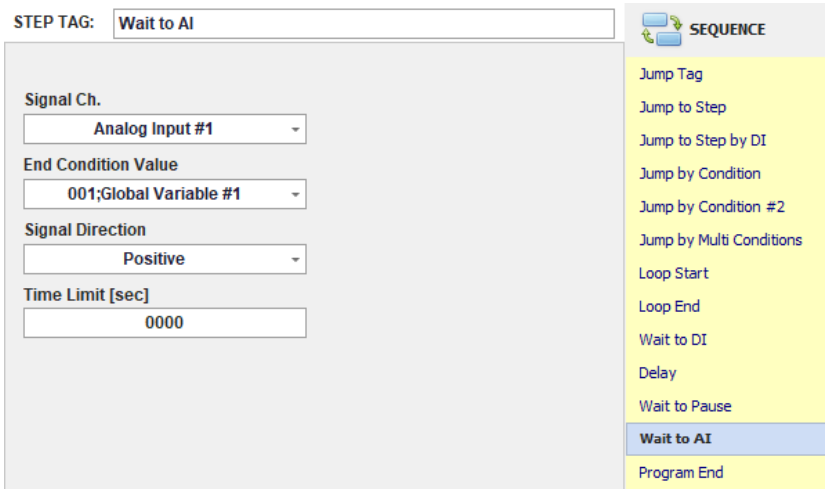
1. **Description:** It makes like idle condition of OACIS. OACIS is doing nothing within the specified duration.
2. **Parameters:**
   - **Duration**: Time period while the OACIS is to be idle. [sec] (0 ~ 9999)
   - **Step Tag**:

K. Wait to Pause



1. **Description:** The step will wait for the Program Stop On signal during the Time Limit. You need to preset Program Start Mode as 1 in User Configuration before you start the program. To terminate this step and go to the next, Program Stop should be turned Off and then Program Start signal has to be pulsed.
2. **Parameters:**
   - **TIME LIMIT (Sec)**: If there is no Program Stop signal within the specified "Time Limit", the OACIS send an error message and ends program. [sec] (0 ~ 99).
   - **Message Popup Disabled:** If OACIS meets this step, a message window pops up on the screen. The popup message shows up at the step and disappears going to the next automatically. [byte] (0 ~ 99).
   - **Step Tag**:

L. Wait to AI



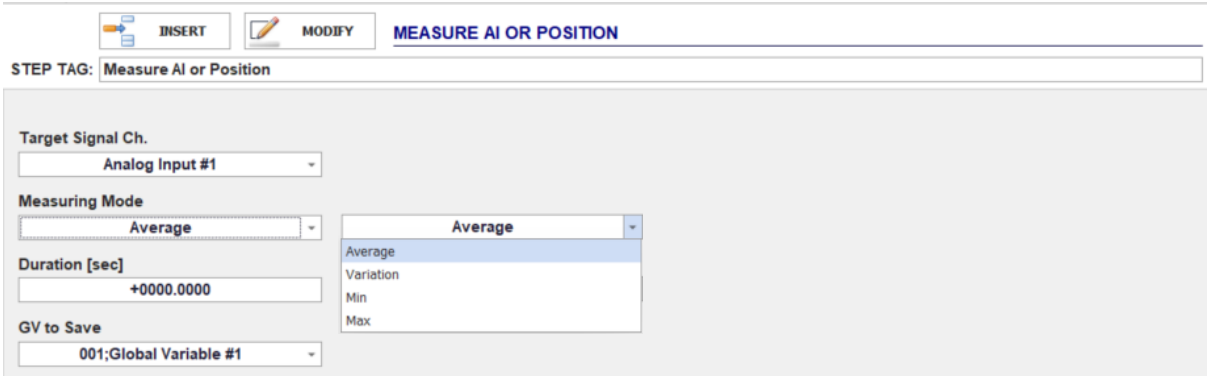
1. **Description:** When the selected signal channel reaches the end condition value in the specified signal direction (Positive or Negative), it proceeds to the next step.
2. **Parameters:**
   - **Signal Ch.:** Signal input channel.
   - **End Condition Value:** The global variable to be met by the selected signal channel.
   - **Signal Direction:** The signal channel must satisfy the completion condition by being greater than or equal to (Positive) or less than or equal to (Negative) the end condition value, depending on the signal direction.
   - **Time Limit:** All conditions must be satisfied within the time limit [sec].
   - **Step Tag:**

M. Program End

3. **Description:** All programs should end with this function. You can't add or delete this function. And one program has only one program end step.
4. **Parameters:**
   - **Step Tag:**

V. MEASURE

A. Measure AI or Position



1. **Description:** Measure the current value of the selected Analog Input Channel or Axis Position then save it into the selected global variable.
2. **Parameters:**
   - **Analog Input Ch.:** Target Analog Input Channel to measure.
   - **Measuring Mode:** You can select one of “Average”, “Variation”, “Min” and “Max”.
     - **Average:** Save the average of the measured values for the specified duration.
     - **Variation:** Save the variation (= Max – Min).
     - **Min:** Save the minimum value.
     - **Max:** Save the maximum value.
   - **Duration:** Time period to measure. If it is 0.002, OACIS captures only one value and save regardless of measuring mode. [sec] (0.002 ~)
   - **GV to Save:** The global variable to save the measured value.
   - **Step Tag**:

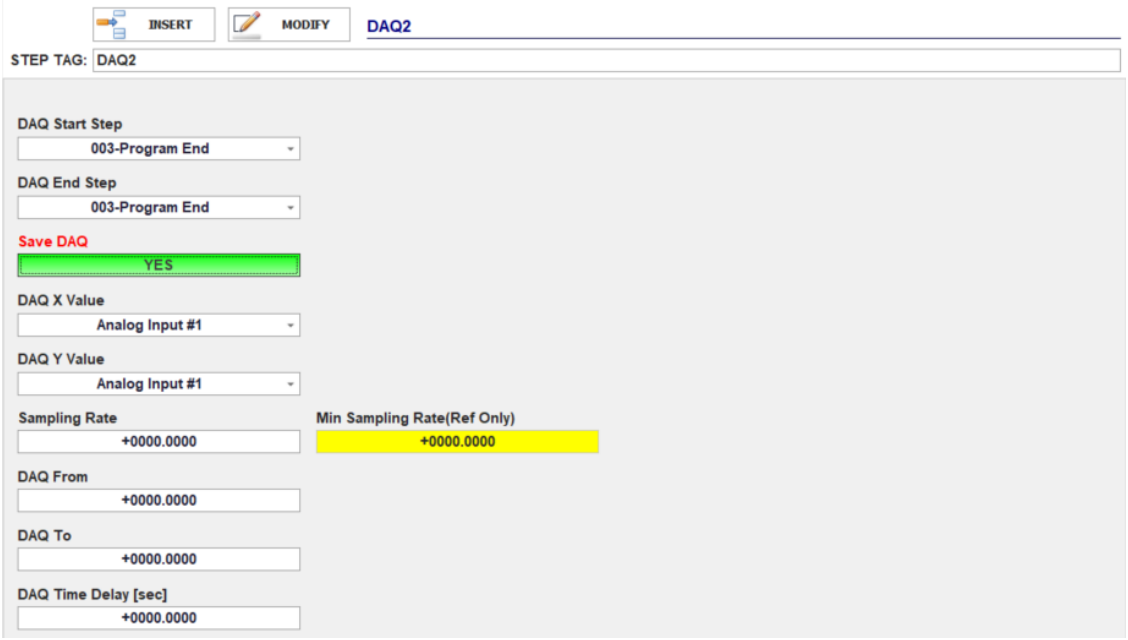
B. DAQ

1. **Description:** Acquires the selected x and y analog signal data and save with the tag name while the selected target step is running. It is to be used for Analysis functions for graph view. You will be able to see the raw data of maximum 4,000 of (x, y) points.

- **Note:** DAQ step should be located before the target step.
- **Note:** The number of all DAQs (DAQ, DAQ2, DAQD and DAQA) per one program is limited to **10**.

2. **Parameters:**
   - **Target Step:** Target Step to acquire data.
   - **Save or No Save:**
     - **Save:** DAQ and show the graph on the main screen.
     - **No Save:** Only DAQ without showing. You can still use it for Analysis functions.
   - **DAQ X Value:** You can select one of Analog Input Channel or Time. If you select “Time”, you will see the time frame curve.
   - **DAQ Y Value:** You can select one of Analog Input Channel or Time.
   - **DAQ Sampling Rate:** The rate to gather the data. It is for X values. If you select “Time” for X value and input 0.02 as Sampling Rate. The OACIS will gather the data every 0.02 seconds. If you select “Position” for X value and input 0.02 as sampling rate, OACIS will gather the data every 0.02mm. If you input “0” as sampling rate, OACIS will gather the data as many as possible.
     - **Note:** Maximum data size is 4,000. So you need to consider the rate, and DAQ range to see the reasonable result. For example, if the range is 1000 and the rate is 0.1, you will see the only first 400 range because of the limited size.
   - **Min. Sampling Rate:** This can help you calculate Min sampling rate in the range of DAQ automatically. But it is only for reference. Users should put a proper value in DAQ sampling rate according to the real range of DAQ.
   - **DAQ from:** The X range starting point to gather the data.
   - **DAQ to:** The X range end point to gather the data.
   - **Step Tag**:
3. **Example:**
   - Target Step: Move to Position of Axis #1. It moves from 0mm to 100mm. (it is servo press)
   - DAQ X value: Axis Position (mm).
   - DAQ Y value: Axis Load (kN).
   - DAQ from: 10
   - DAQ to: 90
   - Sampling Rate: 0.2
   - -> You will see around 400 points like (10, Load), (10.2, Load), …… (90, Load)

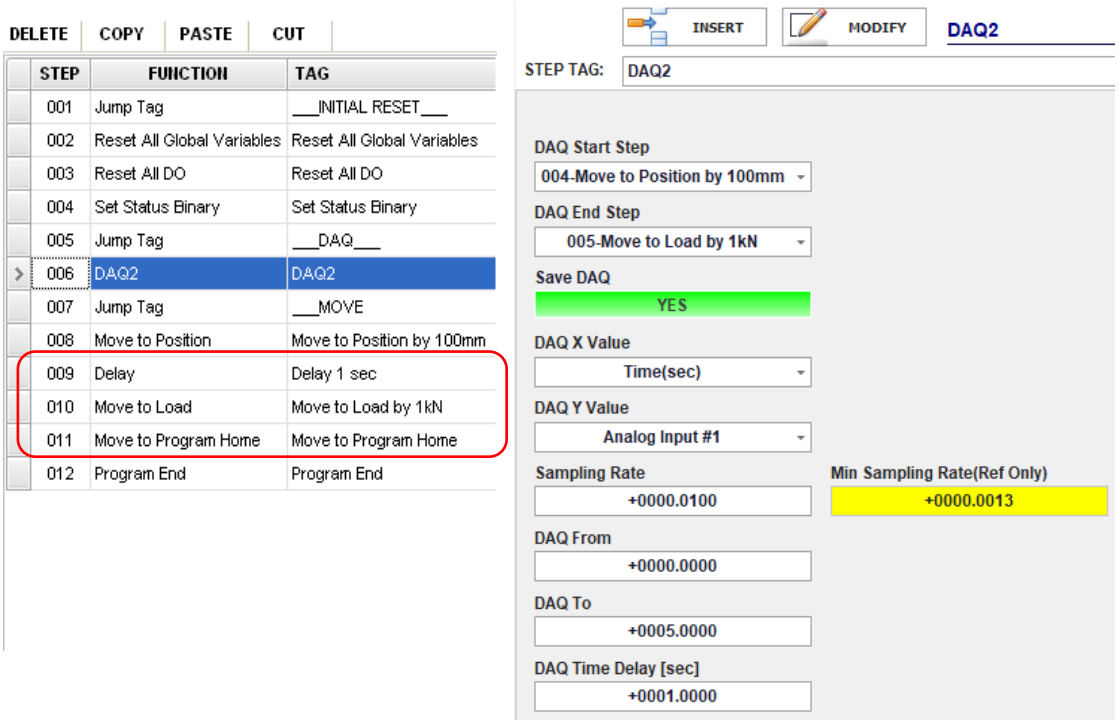
C. DAQ 2



1. **Description:** DAQ2 is similar to DAQ function except for selecting multi steps for data acquisition. It allows you to save multi steps data to a single curve by selecting “DAQ Start Step” and “DAQ End Step”.
   - **Note:** DAQ2 step should be located before the target step.
   - **Note:** The number of DAQs per one program is limited to 10.
   - **Note:** Between “DAQ START STEP” and “DAQ END STEP” is not allowed to have specific functions like loop, jump involved with sequence.
2. **Parameters:**
   - **DAQ Start Step:** Target Step to start acquiring data.
   - **DAQ End Step:** Target Step to finish acquiring data.
   - **Save DAQ:**
     - **YES:** Acquiring Data and show the graph on the main screen. DAQ graph file(*.gph) is to be saved in the local PC.
     - **NO:** Acquiring Data without showing. Graph file will not be saved in the local PC. But it can be used in program for analysis.
   - **DAQ X Value:** You can select one of Analog Input Channel or Time. If you select “Time”, you will see the time frame curve.
   - **DAQ Y Value:** You can select one of Analog Input Channels or Time.
   - **DAQ Sampling Rate:** The rate to gather the data. It is for X values. If you select “Time” for X value and input 0.02 as Sampling Rate, OACIS will gather the data every 0.02 seconds. If you input “0” as sampling rate, OACIS will gather the data as many as possible.
     - **Note:** Maximum data size is 4,000. So you need to consider the rate, and DAQ range to see the reasonable result. For example, if the range is 1000 and the rate is 0.1, you will see the only first 400 range because of the limited size.
   - **Min. Sampling Rate(Ref Only):** This can help you calculate Min sampling rate in the range of DAQ automatically. But it is only for reference. Users should put a proper value in DAQ sampling rate according to the real range of DAQ.
   - **DAQ from:** The X range starting point to gather the data.
   - **DAQ to:** The X range end point to gather the data.
   - **DAQ Time Delay (sec)**: You can set the time delay between each step. During this time delay,

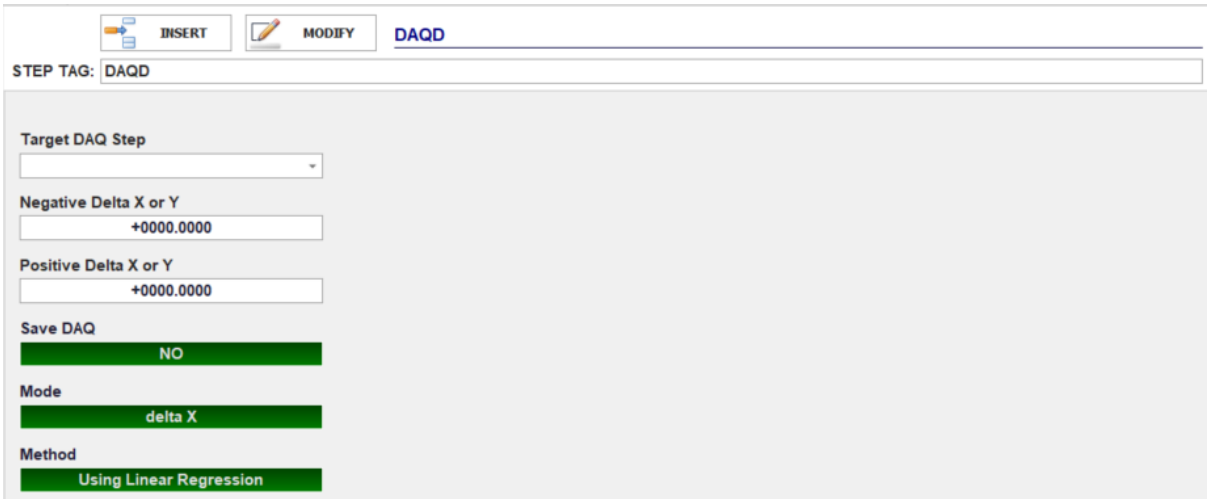
- **Step Tag**:

3. **Example:**



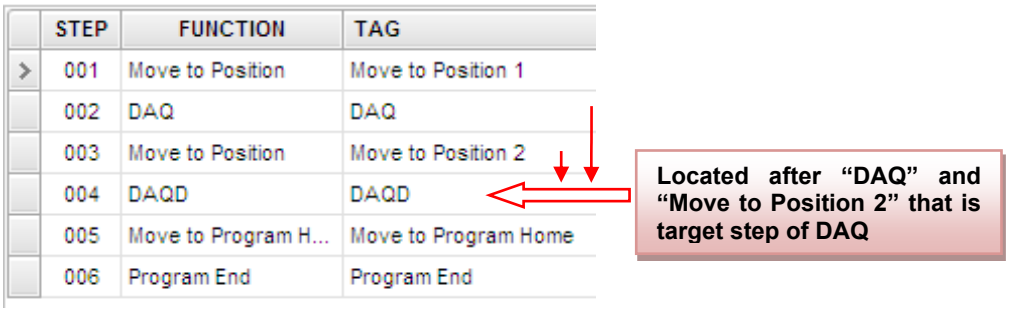
- 'DAQ START STEP' should be located earlier than 'DAQ END STEP'.
- Above example shows the DAQ2 that saves data from step #008 (Move to Position by 100mm) through step #010 (Move to Load by 1kN).

D. DAQD



1. **Description:** You can get the "First Order Differential Curve" of the Target DAQ using this function.

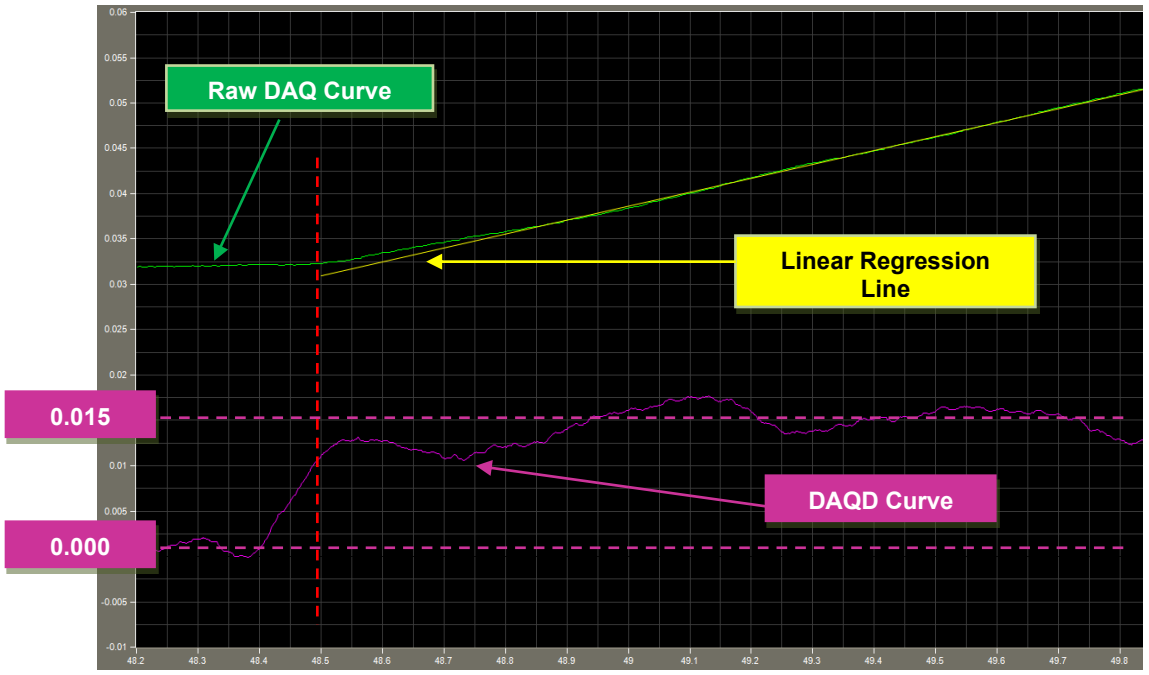
- **Note:** DAQD step should be located after the target step of the Target DAQ.



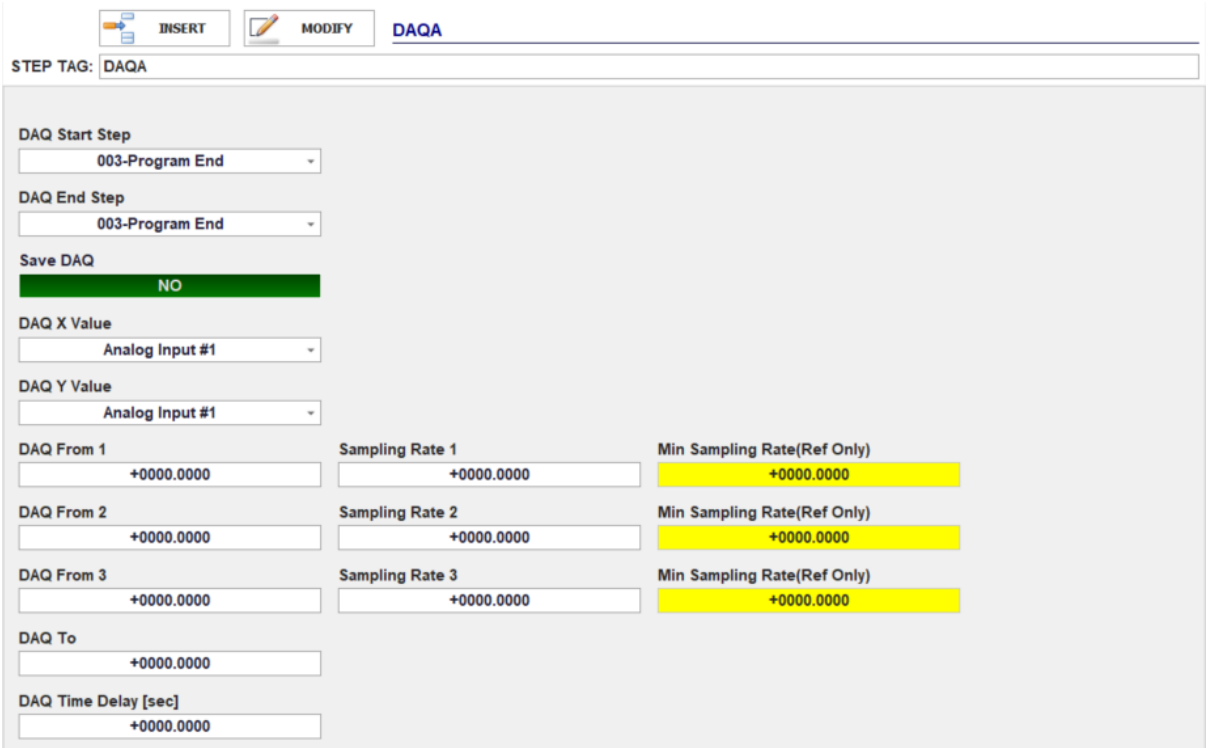
2. **Parameters:**

- **Target DAQ:** Raw Data of the DAQD. DAQ (x, y) -> DAQD (x, δy/δx) with "delta X" mode or (δx/δy, y) with "delta Y" mode.
- **Negative Delta X (or Y) / Positive Delta X (or Y):** You can specify increment of δx (or δy). If you set -0.1 and +0.1 respectively, increment would be 0.2.
- **Save DAQ:**
  - **SAVE:** Acquiring Data and show the graph on the main screen. DAQ graph file(*.gph) is to be saved in the local PC.
  - **NO:** Acquiring Data without showing. Graph file will not be saved in the local PC. But it can be used in program for analysis.
- **MODE:**
  - **Delta X:** OACIS will return differential curve (x, δy/δx).
  - **Delta Y:** OACIS will return differential curve (δx/δy, y).
- **METHOD:**
  - **Using Linear Regression:** OACIS calculates differential point using "Linear Regression" method. You will see more reasonable differential curve. But it takes longer time.
  - **Using Two End Points:** OACIS calculates differential point using only two end points of the increment range like (y2-y1)/(x2-x1). In case, you will see unacceptable curve. But it is pretty faster than "Using Linear Regression" mode.
- **Step Tag:**

3. **Example:**

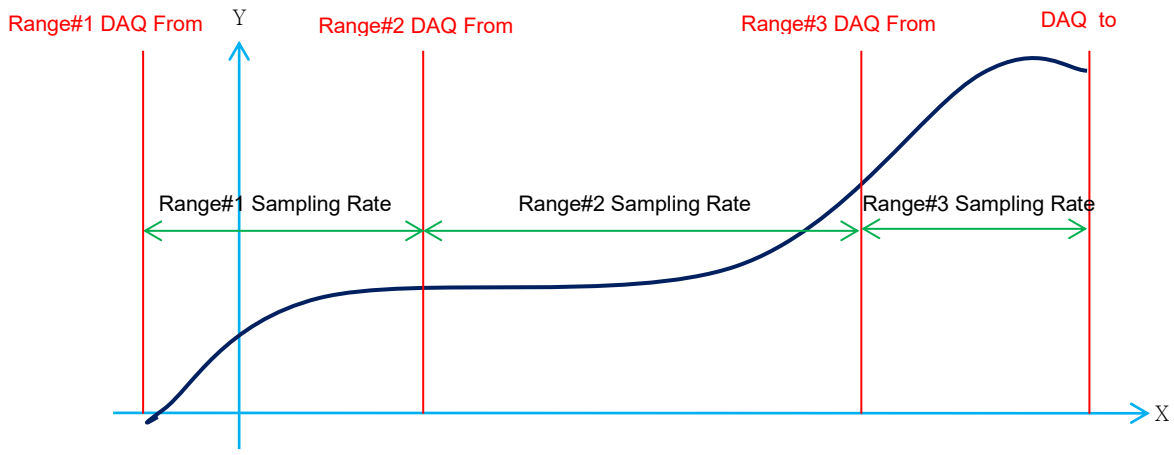


E. DAQA

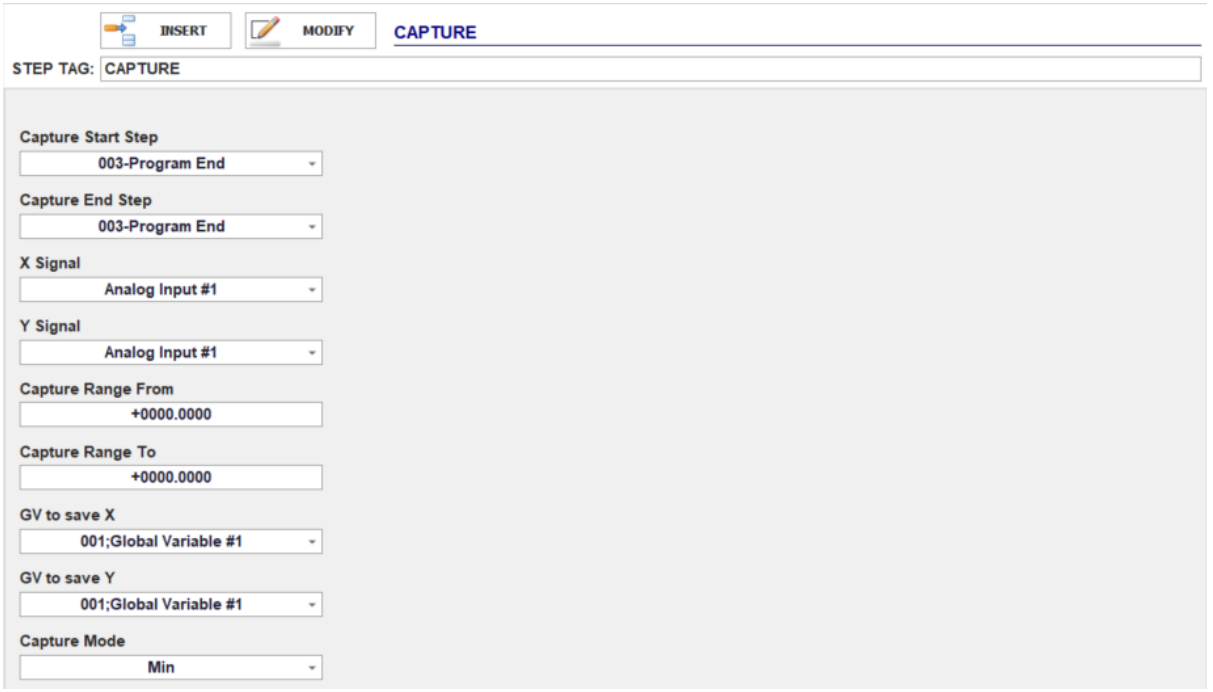


1. **Description:** You can specify three different sampling rate for three different range. And you can select multi steps as target like DAQ2.
   - **Note:** DAQA step should be located before the target step.
   - **Note:** The number of all DAQs (DAQ, DAQ2, DAQD and DAQA) per one program is limited to 10.
   - **Note:** Between “DAQ START STEP” and “DAQ END STEP” is not allowed to have specific functions like loop, jump involved with sequence.

2. **Parameters:**
   - **DAQ Start Step:** Target Step to start acquiring data.
   - **DAQ End Step:** Target Step to finish acquiring data.
   - **Save DAQ:**
     - **SAVE:** Acquiring Data and show the graph on the main screen. DAQ graph file(*.gph) is to be saved in the local PC.
     - **NO:** Acquiring Data without showing. Graph file will not be saved in the local PC. But it can be used in program for analysis.
   - **DAQ X Value:** You can select one of Analog Input Channel or Time. If you select “Time”, you will see the time frame curve.
   - **DAQ Y Value:** You can select one of Analog Input Channels or Time.
   - **DAQ from 1 / DAQ From 2 / DAQ From 3 / DAQ To:** You can specify each DAQ range.
   - **Sampling Rate 1 / Sampling Rate 2 / Sampling Rate 3:** You can specify individual sampling rate of each range. For example, If you select “Time” for X value and input 0.02 as Sampling Rate, OACIS will gather the data every 0.02 seconds. If you select “Position” for X value and input 0.02 as sampling rate, OACIS will gather the data every 0.02mm. If you input “0” as sampling rate, OACIS will gather the data as many as possible.
     - **Note:** Maximum data size is 4,000. So you need to consider the rate, and DAQ range to see the reasonable result. For example, if the range is 1000 and the rate is 0.1, you will see the only first 400 range because of the limited size.
   - **Min Sampling Rate(Ref Only):** This can help you calculate Min sampling rate in the range of DAQ automatically. But it is only for reference. Users should put a proper value in DAQ sampling rate according to the real range of DAQ.
   - **DAQ Time Delay (sec):** You can set the time delay between each step. During this time delay, OACIS does not acquire the data.
   - **Step Tag:**
3. **Example:**

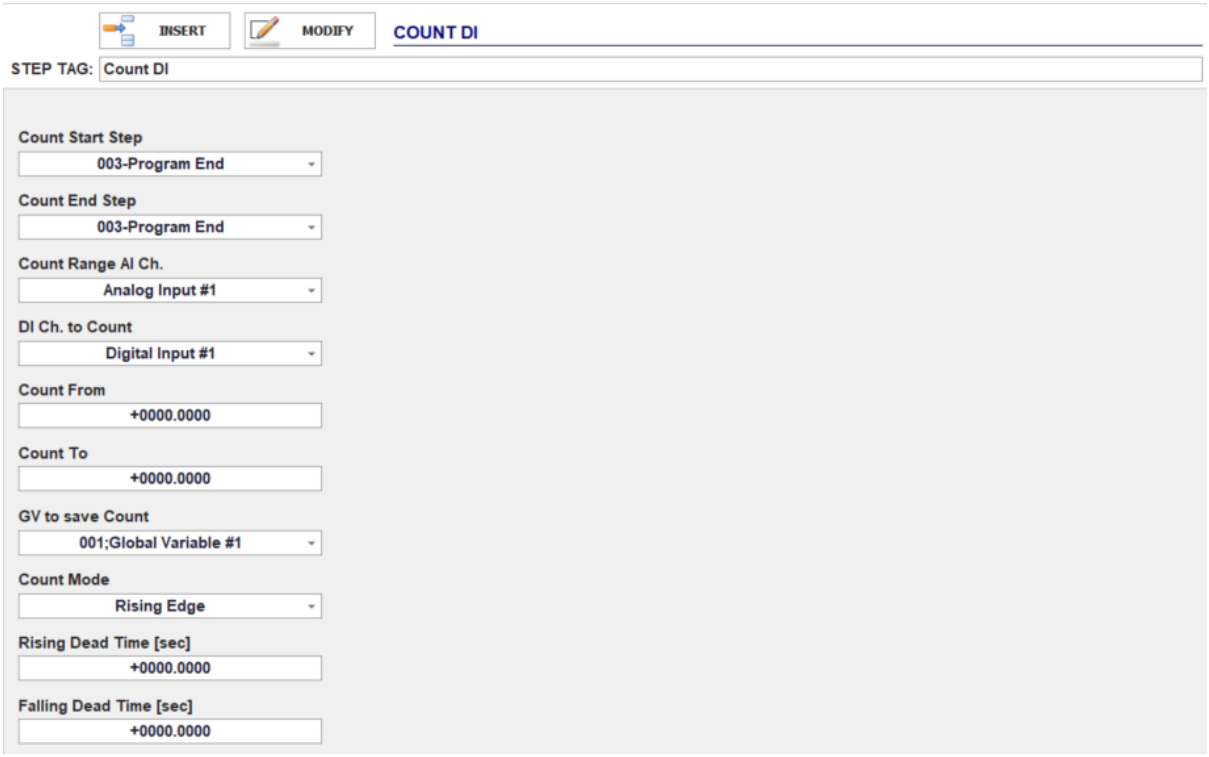


F. CAPTURE



1. **Description:** You can get the Maximum (or Minimum) Y and X data without DAQ while the selected capturing steps are running. You can select multi steps as target like DAQ2. This function does not utilize DAQ for analysis because it returns the X and Y data just by a real-time comparison.
   - **Note:** CAPTURE step should be located before the target step.
2. **Parameters:**
   - **Capture Start Step:** Target Step to start capturing data.
   - **Capture End Step:** Target Step to finish capturing data.
   - **X Signal:** You can select one of Analog Input Channel or Time.
   - **Y Signal:** You can select one of Analog Input Channels or Time.
   - **Capture Range (of X) From:** Start point of the range to capture the data.
   - **Capture Range (of X) To:** End point of the range to capture the data.
   - **GV to save X:** The global variable to save the X value captured until the capturing steps end.
   - **GV to save Y:** The global variable to save the Y value captured until the capturing steps end.
   - **Capture Mode**: You can select one of “Min” and “Max”.
     - **Min:** Save the minimum value during the capturing step.
     - **Max:** Save the maximum value during the capturing step.
   - **Step Tag:**

G. Count DI

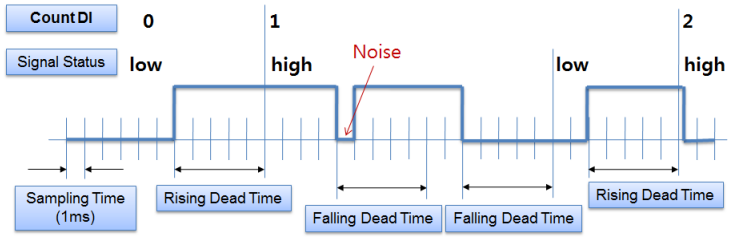


1. **Description:** You can count the pulses of programmable digital inputs in the counting range of a selected AI channel while the steps specified from COUNTING START STEP to COUNTING END STEP are running. You can select the type of pulse in Counting MODE. Dead Time is for removing the signal noise of digital inputs.
   - **Note:** Count DI step should be located before the target step.
   - **Note:** You can use up to 10 Count DI functions per program and use them for the same target step as well.
   - **Note:** Sampling frequency is 1kHz which means 1ms of sampling time.
2. **Parameters:**
   - **Count Start Step:** Target Step to start Count DI.
   - **Count End Step:** Target Step to end Count DI.
   - **Count Range AI Ch.:** You can select one of Analog Input Channels or Time.
   - **DI Ch to Count:** Programmable digital input to be counted.
   - **Count From:** Start point of the range to count DI signal.
   - **Count To:** End point of the range to count DI signal.
   - **GV to Save Count:** Global variable to save the DI value counted until the Count DI steps end.
   - **Count Mode**: You can select one of four options.
     - **Rising Edge:** Count the signal changing from low to high.
     - **Falling Edge:** Count the signal changing from high to low.
     - **Rising and Falling Edge:** Count the signal changes either way starting from low.
     - **Falling and Rising Edge**: Count the signal changes either way starting from high.
   - **Rising Dead Time (sec):** High signals are supposed to be counted when the signal keeps “high” during the rising dead time consecutively from low.
   - **Falling Dead Time (sec):** Low signals are supposed to be counted when the signal keeps “low” during the falling dead time consecutively from high.
   - **Step Tag:**

3. Example:

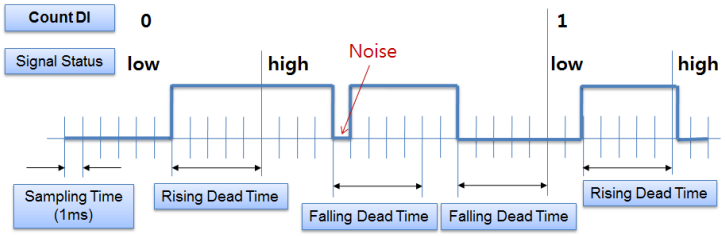
Case 1. Rising Edge

- Rising Dead Time (sec): 0.005
- Falling Dead Time (sec): 0.005
- **Saved Counts: 2**



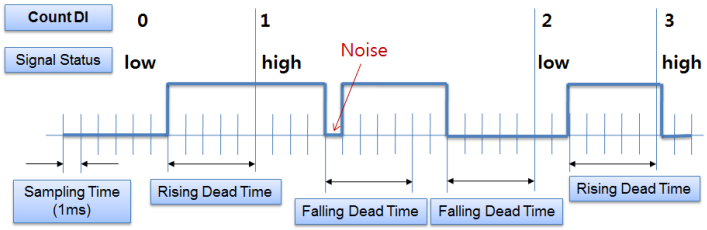
Case 2. Falling Edge

- Rising Dead Time (sec): 0.005
- Falling Dead Time (sec): 0.005
- **Saved Counts: 1**



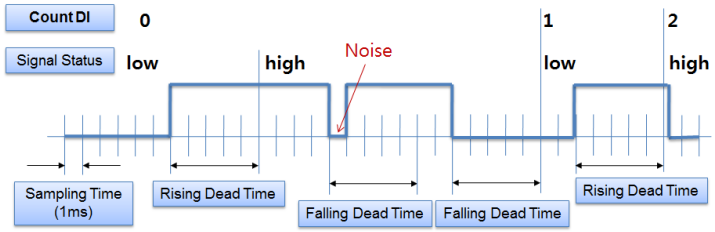
Case 3. Rising and Falling Edge

- Rising Dead Time (sec): 0.005
- Falling Dead Time (sec): 0.005
- **Saved Counts: 3**



Case 4. Falling and Rising Edge

- Rising Dead Time (sec): 0.005
- Falling Dead Time (sec): 0.005
- **Saved Counts: 2**

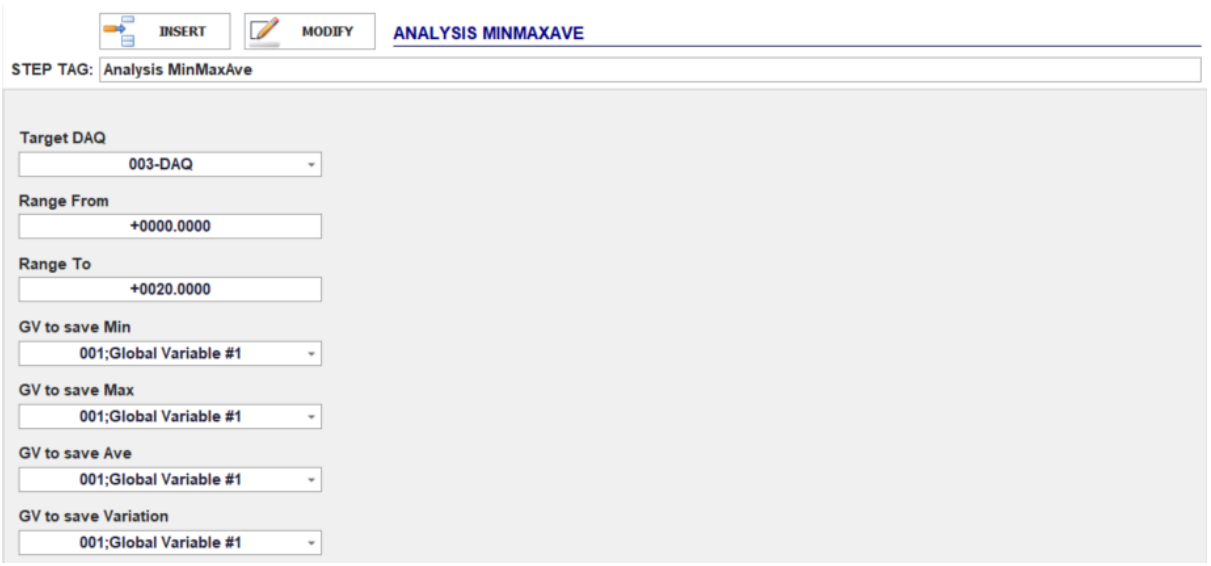


VI. ANALYSIS

: Analysis function is to analyze the data as a result of data acquisition (DAQ). And after the analysis, the result will restore to global variables.

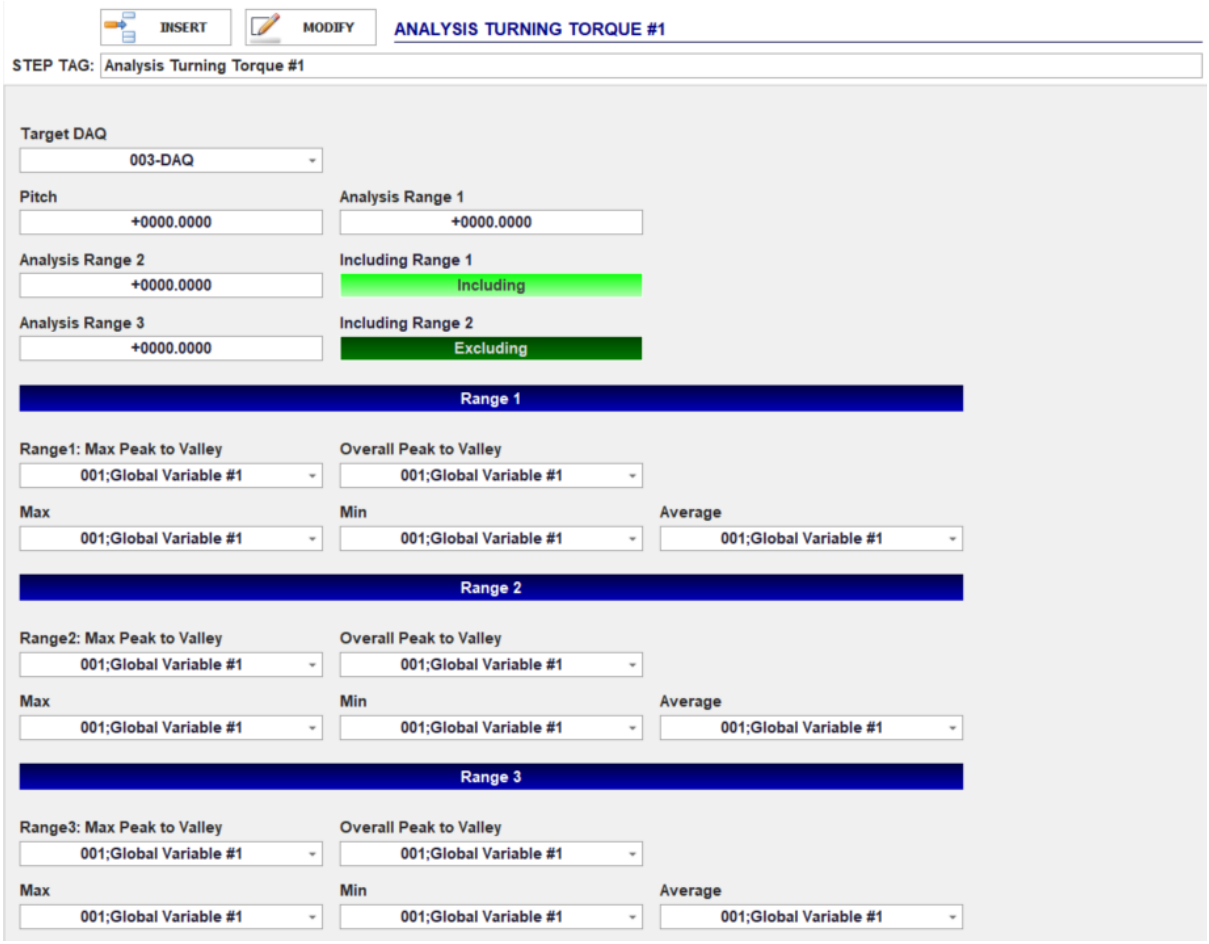
 # Note: Analysis function should be located after target DAQ. That is to say, the step number of analysis function has more higher than the one of target DAQ. If it is located before the target DAQ, you will see the unexpected result.

A. Analysis MinMaxAve

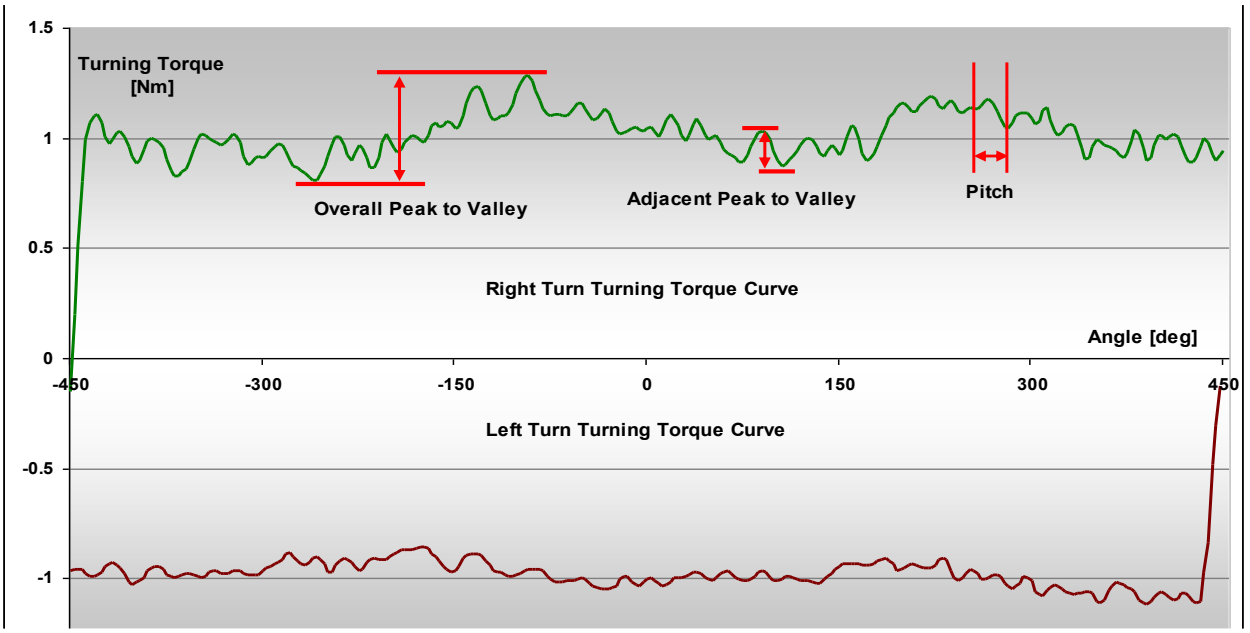


1. **Description:** Find Minimum value, Maximum value, Average value and Variation of the target DAQ then save them into the selected global variables.
2. **Parameters:**
   - **Target DAQ:** Target DAQ for the analysis.
   - **Range From:** Starting point of the range for the Analysis.
   - **Range To:** End point of the range for Analysis.
     - ➢ DAQ Start Point < Analysis Start Point < Analysis End Point < DAQ End Point
   - **GV to Save Min:** Global Variable to save the Minimum value.
   - **GV to Save Max:** Global Variable to save the Maximum value.
   - **GV to Save Ave:** Global Variable to save the Average value.
   - **GV to Save Variation:** Global Variable to save the Variation (= Max – Min).
   - **Step Tag**:

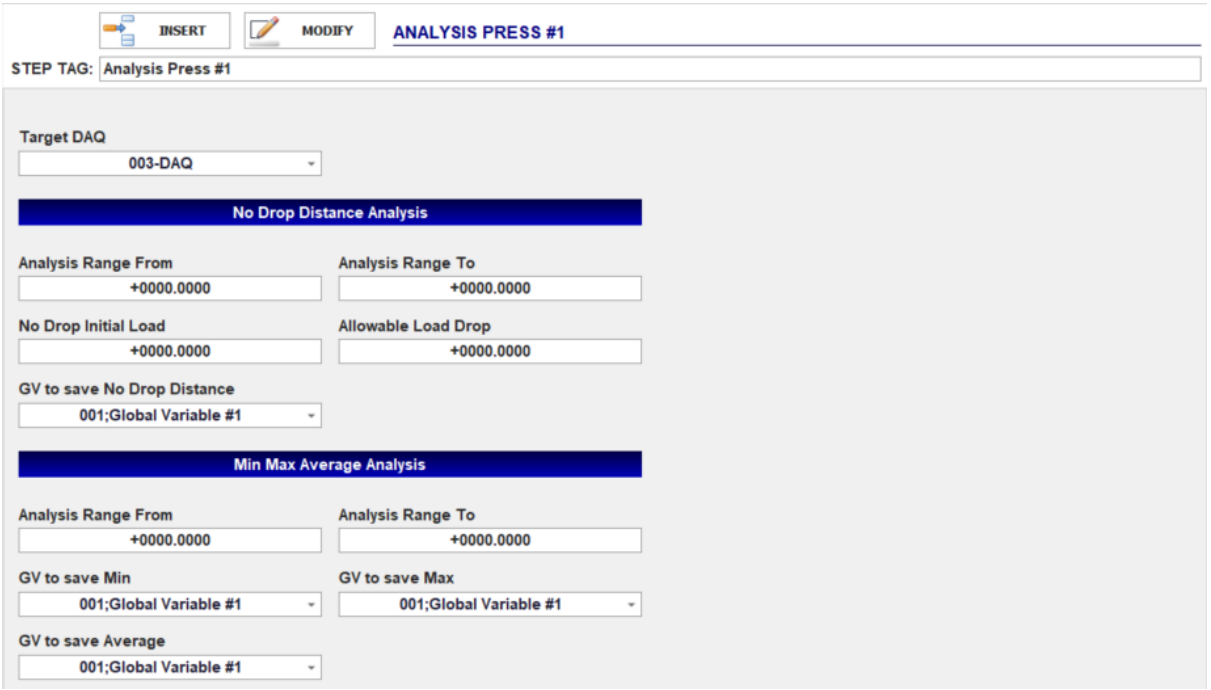
B. Analysis Turning Torque #1



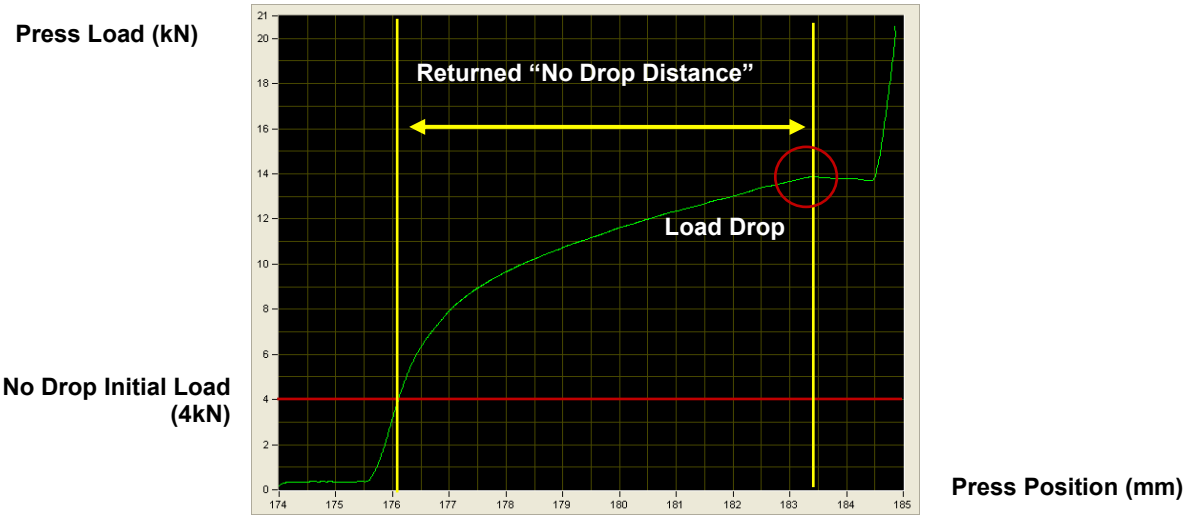
1. **Description:** Special function to analyze turning torque curve. You can find Max Peak to Valley, Overall Peak to Valley, Maximum, Minimum and Average of the specified range. You can specify three ranges.
2. **Parameters:**
   - **Target DAQ:** Target DAQ for the analysis.
   - **Pitch:** Tooth pitch for the analysis.
   - **Analysis Range 1:** If you input 90deg, the first range would be from -90deg to +90deg.
   - **Analysis Range 2:** If you input 180deg, the second range would be from -180deg to +180deg.
     - ➢ Excluding Range 1 or Including Range 1: If you select “Excluding” option, the second range would be from -180deg to -90deg and from +90deg to +180deg. And if you select “Including” option, the second range would be from -180deg to +180deg.
   - **Analysis Range 3:** If you input 360deg, the third range would be from -360deg to +360deg.
     - ➢ Excluding Range 2 or Including Range 2: see the above.
   - **Max Peak to Valley:** Max adjacent Peak to Valley of the specified range.
   - **Overall Peak to Valley:** Variation of the range (= Max – Min).
   - **Max:** Maximum value of the range.
   - **Min:** Minimum value of the range.
   - **Average:** Average of the range.
   - **Step Tag**:



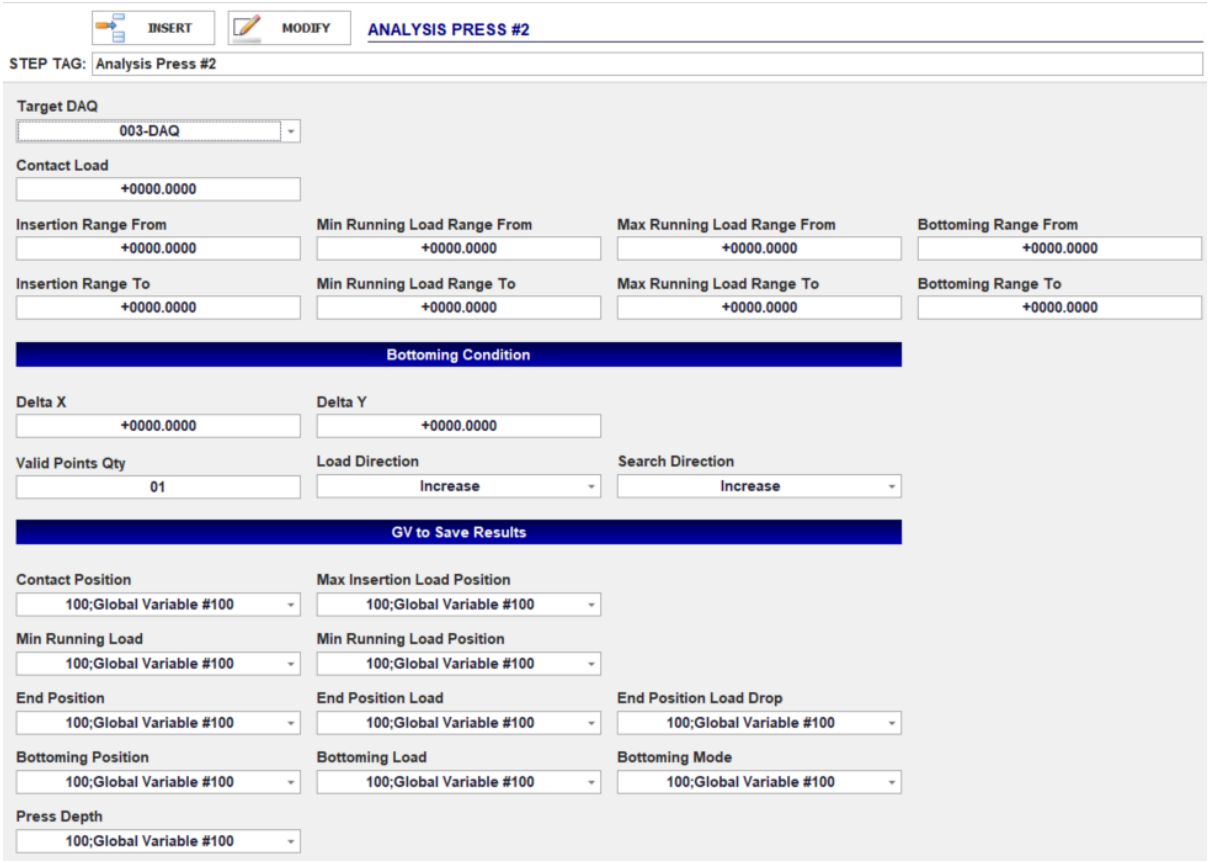
C. Analysis Press #1



1. **Description:** Special function to analyze Pressing curve. It consists of two analyses. One is the analysis to find “No drop distance” and the other one is to find Min, Max and Average value.
2. **Parameters:**
   - **Target DAQ:** Target DAQ for the analysis.
   - **No Drop Analysis Range From:** Starting point of the range for the Analysis.
   - **No Drop Analysis Range To:** End point of the range for the Analysis.
   - **No Drop Initial Load:** Starting condition to monitor the “No Drop Distance”.
   - **No Drop Allowable Load Drop:** If the load drop is bigger than this value, OACIS restart to monitor the “No Drop Distance”.
   - **GV to save No Drop Distance:** The global variable to save the distance while running with no drop.
   - **Step Tag:**
3. **Example:**

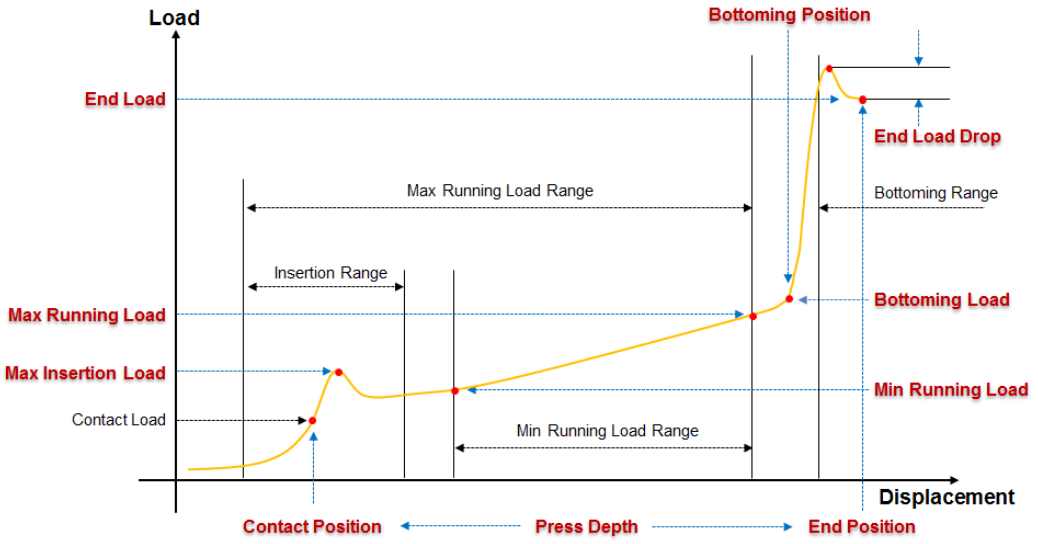


D. Analysis Press #2

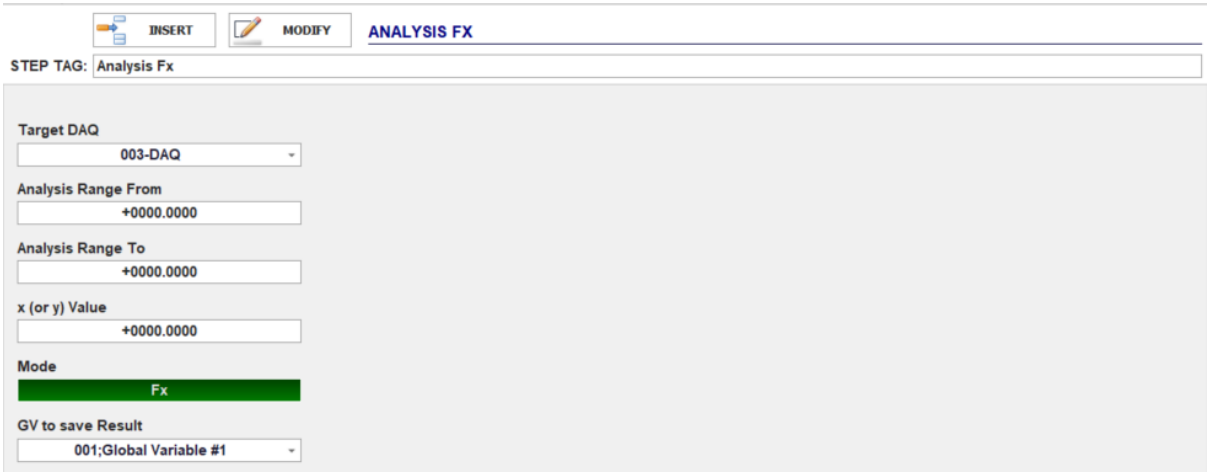


1. **Description:** A special function to analyze Pressing curve. You can get a lot of important information during the press by this function.
2. **Parameters:**
   - **Target DAQ:** Target DAQ for the analysis.
   - **Contact Load:** User input load to decide a contact point.
   - **Insertion Range From/To:** Range to proceed with the insertion.
   - **Min Running Load Range From/To:** Range to keep over the minimum load during the press.
   - **Max Running Load Range From/To:** Range to keep under the maximum load during the press.
   - **Bottoming Range From/To:** OACIS finds "Bottoming Condition" only in this range.

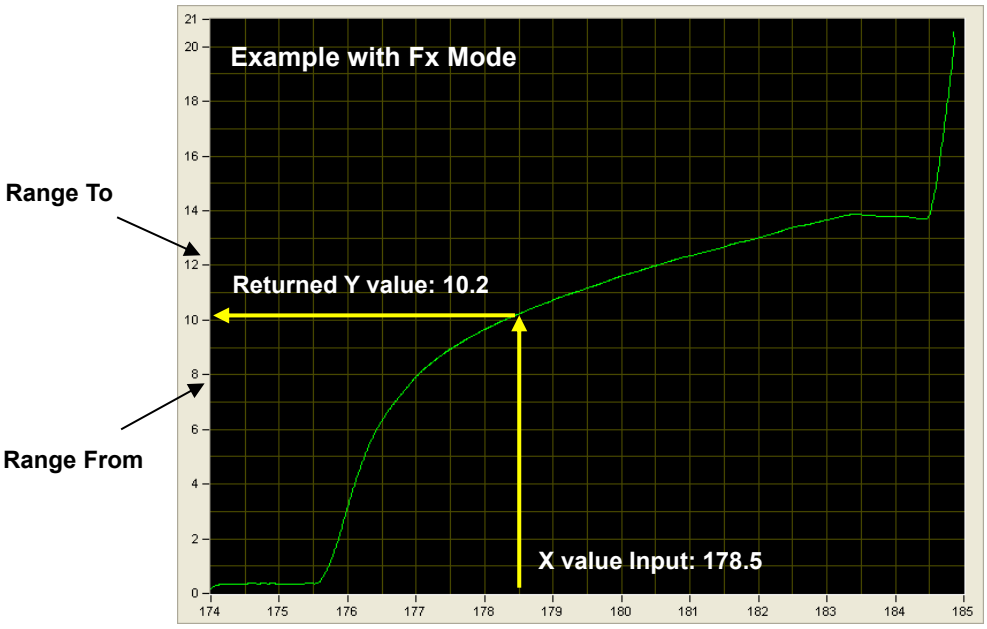
- **Bottoming Condition**
  : Delta X / Delta Y / Valid Points Qty / Load Direction / Search Direction
  - It is possible to put only positive numbers in both Delta X and Delta Y.
  - Delta Y can be used as a signal noise limit on a specific occasion.
  - Bottoming Condition is to be defined by five parameters, Delta X, Delta Y, Valid Points Qty, Load Direction and Search Direction. If the actual "**δy/δx**" is greater than the specified "Delta Y / Delta X" consecutively in the number of times more than the specified "Valid Points Qty", OACIS returns the first point of the first satisfied "**δy/δx**" as the bottoming point.
  - When the load increases in a negative direction, you can select Load Direction as "decrease". Even in this case, you should put the absolute value of Y displacement in Delta Y.
  - If you want to find the bottoming point backwards from the end position, you can choose Search Direction as "decrease". It is useful when there are several bottoming points and if you want to find the last one.
- **Contact Position:** Position value at the Contact Load.
- **Max Insertion Load:** Maximum Load in the Insertion Range.
- **Max Insertion Load Position:** Position value at the Max Insertion Load.
- **Min Running Load:** Minimum load in the Min Running Load Range.
- **Min Running Load Position:** Position value at the Min Running Load.
- **Max Running Load:** Maximum load in the Max Running Load Range.
- **Max Running Load Position:** Position value at the Max Running Load.
- **End Position:** Position value at the end of the press.
- **End Position Load:** Load value at the End Position.
- **End Position Load Drop:** Difference between Max Load and End Position Load.
- **Bottoming Position:** Second position value of the first "δy/δx" that is satisfied with the Bottoming Condition.
- **Bottoming Load:** Load value at the Bottoming Position.
- **Bottoming Mode:** It saves bottoming information to the selected global variable as below.
  - **1:** Pass. OACIS found the bottoming condition in the bottoming range.
  - **2:** Fail. OACIS failed to find the bottoming condition in the bottoming range. In this case, OACIS returns (0, 0) into the bottoming position.
- **Press Depth:** Difference between the Contact Position and the End Position.
- **Step Tag:**



E. Analysis Fx

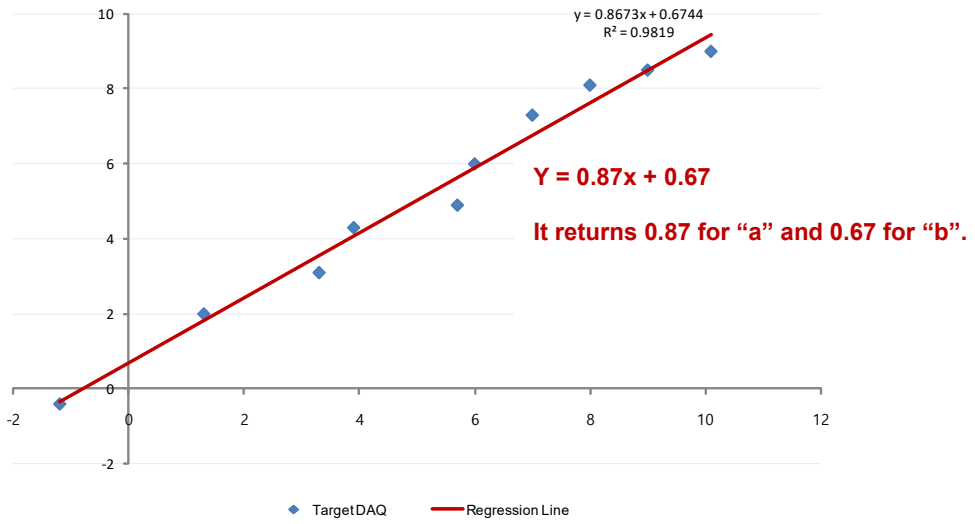


1. **Description:** The function is to return the y-crossed value or x-crossed value of the curve. There are two modes, f(x) and f(y).
2. **Parameters:**
   - **Target DAQ:** Target DAQ for the analysis.
   - **Analysis Range From:** Starting point of the range for the Analysis.
   - **Analysis Range To:** End point of the range for the Analysis.
   - **x (or y) value:** It depends on the mode selection.
   - **Mode:** You can select one of Fx and Fy.
     - Ex: If the selected mode is Fx. And the input x value is 5. The function is to return Y = f(5).
   - **GV to save Result:** The global variable to save the result value.
   - **Step Tag:**
3. **Example:**



F. Linear Regression

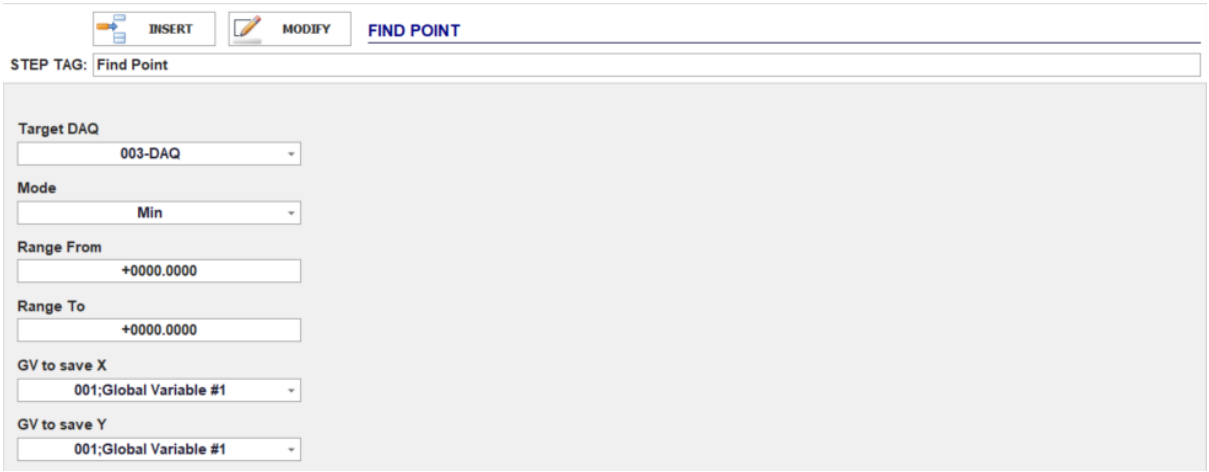
1. **Description:** It returns the "Slope" and "Y Intercept" of the Linear Regression Line that comes from the specified range of the Target DAQ.
   - **Note:** This function should be located after target step of the target DAQ step. For example, if the target step of the Target DAQ is "Move to Load" with step #45, this function's step number should be later than 45. Target DAQ -> Move to Load (target step of the Target DAQ) -> Linear Regression. Otherwise, you will see the unexpected values.
2. **Parameters:**
   - **Target DAQ:** Target DAQ for the analysis.
   - **Analysis Range From:** Starting point of the range for the Analysis.
   - **Analysis Range To:** End point of the range for the Analysis.
   - **GV to Save a (Slope):** The slope of the linear regression line formula (Y = aX + b) driven from the Target DAQ.
   - **GV to Save b (Y intercept):** The Y Intercept of the linear regression line formula (Y = aX + b) driven from the Target DAQ.
3. **Example:**



G. Linear Regression #2

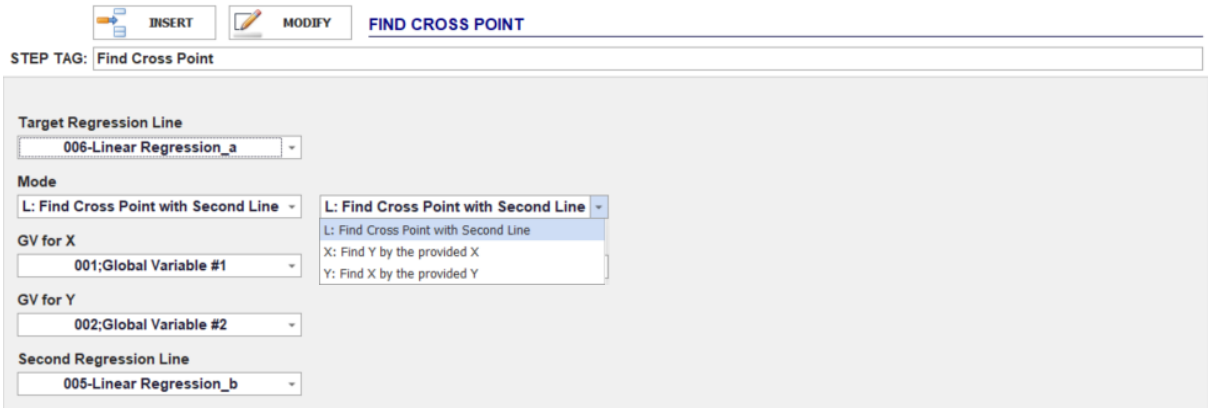
1. **Description:** It is same as “Linear Regression” Function except that it returns “R^2” as well. “R^2” represents reliability of “a” and “b”. Please see the “Linear Regression” function for more details.

H. Find Point



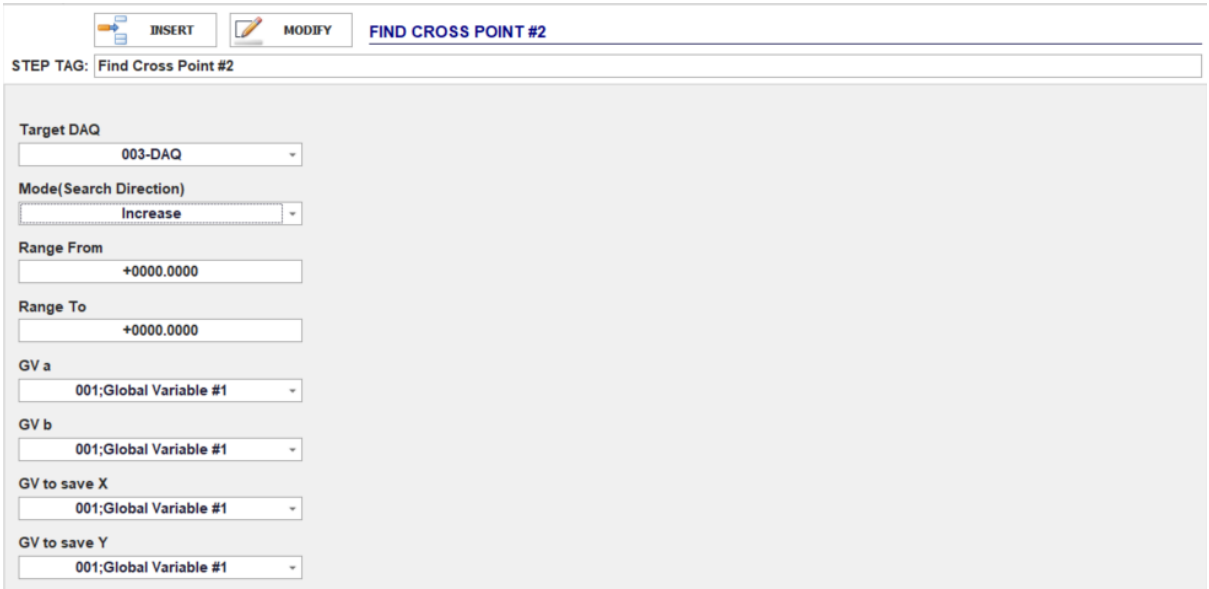
1. **Description:** It returns the Minimum(or Maximum) point (x, y) in the selected target DAQ curve. You can also define your analysis range of the target DAQ.
2. **Parameters:**
   - **Target DAQ:** Target DAQ for the analysis.
   - **Mode:**
     - **Min:** Save the minimum value in the target DAQ.
     - **Max:** Save the maximum value in the target DAQ.
   - **Range From:** Start point of the range for the analysis.
   - **Range To:** End point of the range for the analysis.
   - **GV to save X:** The global variable to save the X value of the captured point whose Y is a Minimum or Maximum value.
   - **GV to save Y:** The global variable to save the Y value which is a Minimum or Maximum.
3. **Step Tag:**

I. Find Cross Point



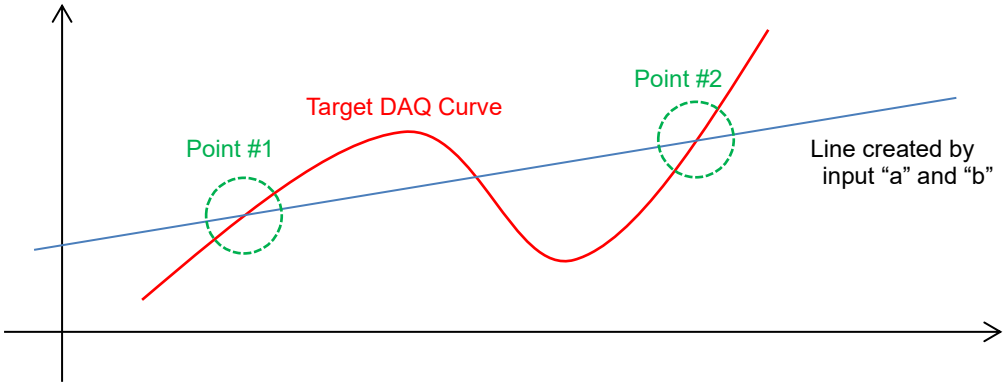
1. **Description:** It returns the point (x,y) information of the crossed point between the selected "Linear Regression Line" and the other line per the selected mode. This step should be located after target regression line step.
2. **Parameters:**
   - **Target Regression Line:** Target Regression Line for the analysis.
   - **Mode**
     - **L: Find Cross Point with Second Line:** It returns the cross point (x, y) of two regression lines (Target Regression Line and Second Regression Line)
     - **X: Find Y by the provided X:** It returns 'Y' value paired with the selected 'X' value on the Target Regression Line. It works like "Analysis Fx" function (with Fx mode).
     - **Y: Find X by the provided Y:** It returns 'X' value paired with the selected 'Y' value on the Target Regression Line. It works like "Analysis Fx" function (with Fy mode).
   - **GV for X:** Global Variable to be used for 'X'.
   - **GV for Y:** Global Variable to be used for 'Y'.
   - **Second Regression Line:** Regression Line to find cross point.
   - **Step Tag:**

J. Find Cross Point #2



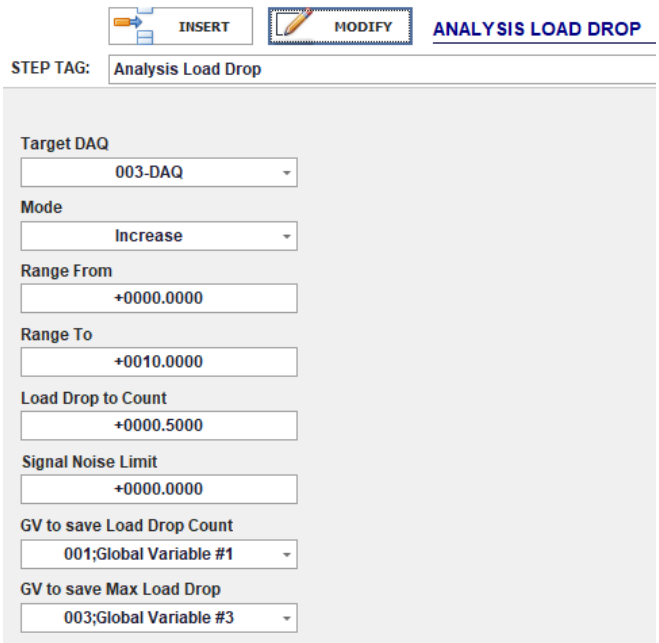
1. **Description:** It returns the crossed point (x, y) information between the selected DAQ curve and the Line created by the input "Slope" and "Y Intercept". If there are more than one crossed points, it returns the first point depending on the selected "Search Direction Mode".

2. **Parameters:**
   - **Target DAQ:** Target DAQ for the analysis.
   - **Mode (Search Direction):**
     - **Increase:** It searches the cross point in the incremental direction.
     - **Decrease:** It searches the cross point in the decremental direction.
   - **Range From:** starting point of the range for the analysis.
   - **Range To:** end point of the range for the analysis.
   - **GV a:** The global variable to be used as "Slope" of the virtual line.
   - **GV b:** The global variable to be used as "Y Intercept" of the virtual line.
   - **GV to save X:** The global variable to save "X" value of the crossed point.
   - **GV to save Y:** The global variable to save "Y" value of the crossed point.
3. **Example:**



If you select "Increase" as Search Direction Mode, it returns "Point #1". And if you select "Decrease", it returns "Point #2".

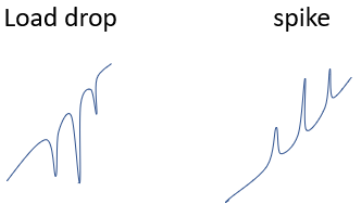
K. Analysis Load Drop



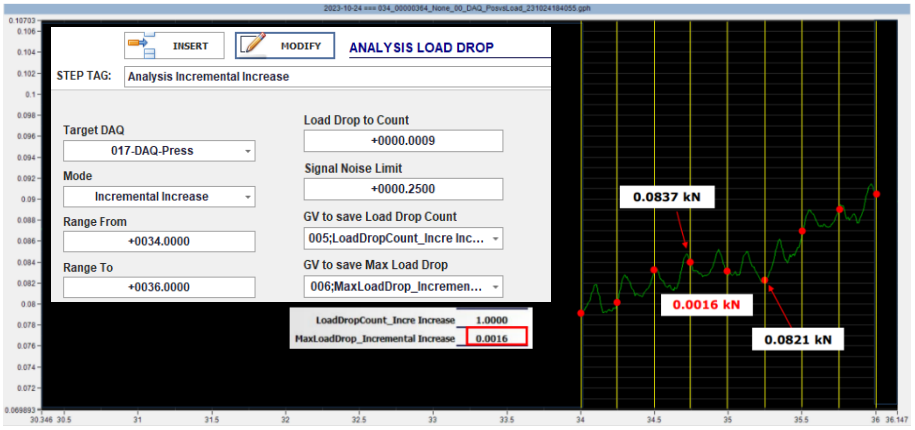
1. **Description:** It returns the "Load Drop Count" and "Max Load Drop" per the input parameters and selected Mode.

2. **Parameters:**

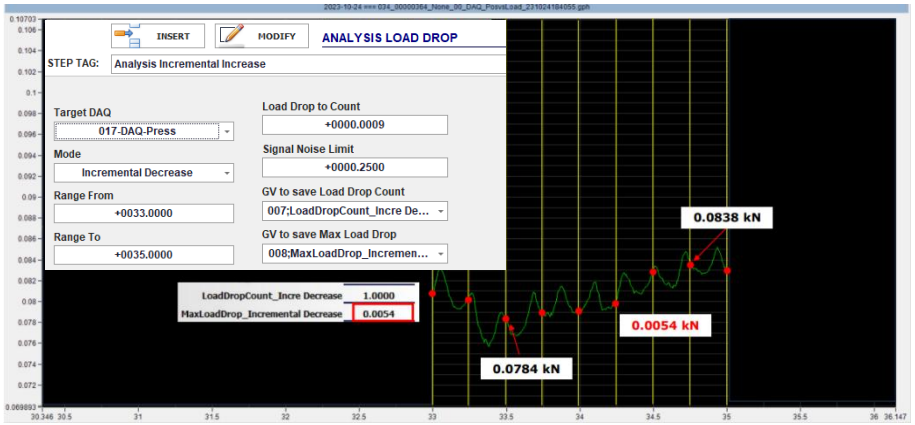
- **Target DAQ:** Target DAQ for the analysis.
- **Mode**
  - **Increase:** It is finding “Load Drop”.
  - **Decrease:** It is finding “Load Spike”.



  - **Incremental Increase:** When you use "Incremental Increase" mode to find load drop, you have to put in "Incremental" which is a pitch for inspection. It returns results by analyzing load drop at every pitch from “Range From”. It ignores values between pitch and pitch when analyzing.
  - **Incremental Decrease:** You can also find load spike in "Incremental Decrease" mode in a same way as Incremental Increase.



< Incremental Increase>

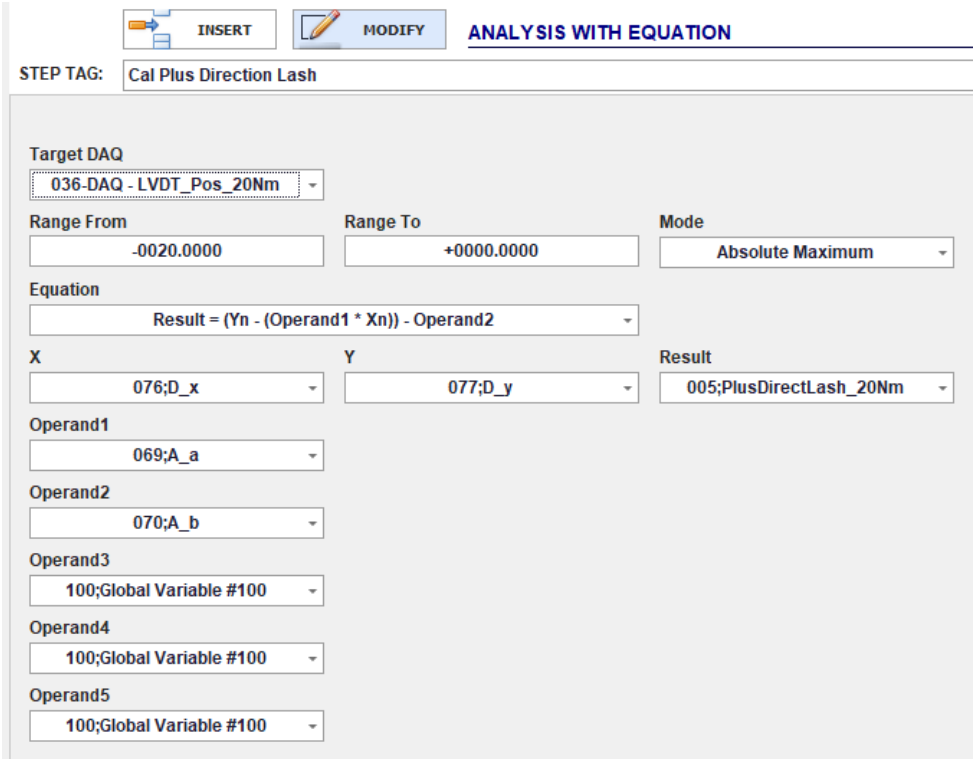


< Incremental Decrease>

- **Range From:** Starting point of the analysis.
- **Range To:** End point of the analysis.
- **Load Drop to Count:** As an input parameter, it is the amount of load drop that users want to count. OACIS is counting the Load Drop (or Spike) more than “Load Drop to Count”.
- **Signal Noise Limit:** OACIS does not count as a load drop if it is less than the signal noise limit.

- **GV to Save Load Drop Count:** Global Variable for counting and storing in “Load Drop Count” when it is greater than the Signal Noise limit and greater than the Load drop to count value.
- **GV to Save Max Load Drop:** Global Variable to save the maximum value among the counted load drops in “Max Load Drop”.

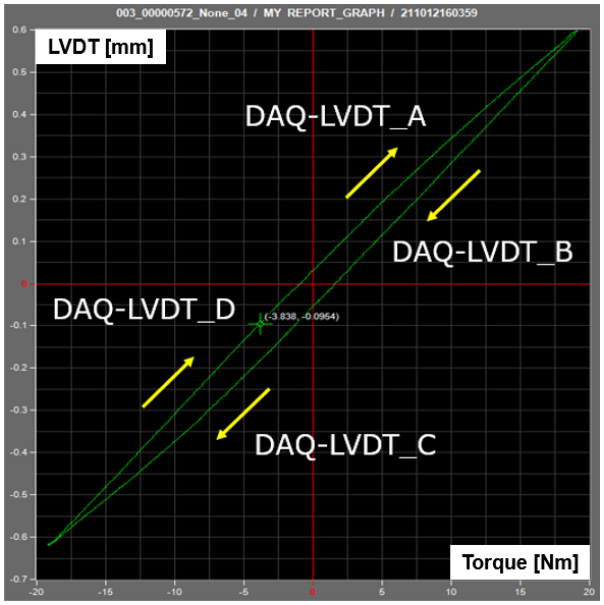
L. Analysis With Equation



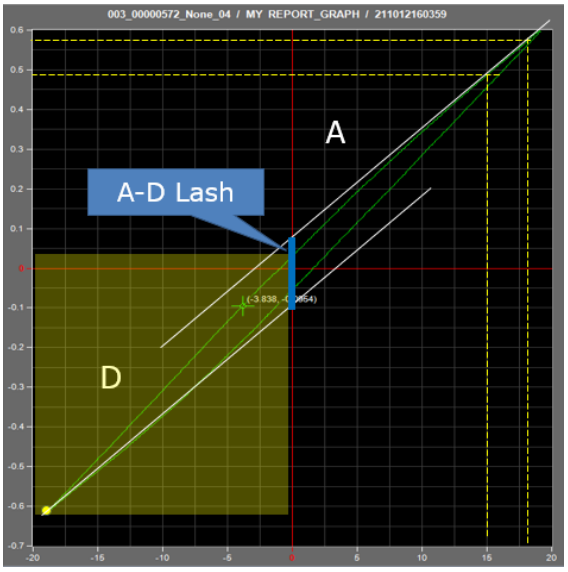
1. **Description:** This specialized analysis function is designed for lash measurement applications. It utilizes two linear equations with identical slopes: a **fixed reference line** and a **variable line** that shifts between the 'Range From' and 'Range To' parameters. The resulting lash value is derived from the distance between the two x-intercepts or y-intercepts.
2. **Parameters:**
   - **Target DAQ:** The graph file to be analyzed.
   - **Range From/To:** The effective range for determining multiple variable lines is defined between 'Range From' and 'Range To'.
   - **Mode:** Select one from the following options: Maximum, Minimum, Absolute Maximum, or Absolute Minimum. The identified coordinates (X, Y) and the resulting lash value (Result) vary depending on the selected mode.
   - **Equation:** There are two methods available: Result=((Operand1*Xn-Yn)/Operand1)-Operand2, which identifies the lash using x-intercepts, and Result=(Yn-(Operand1*Xn))-Operand2, which uses y-intercepts.
   - **X/Y:** The identified coordinates are determined based on the selected mode, analysis range, and equation.
   - **Result:** The resulting value is derived by satisfying all the aforementioned criteria.
   - **Operand1:** The slope of the fixed reference line.
   - **Operand2:** Y-intercept of the fixed reference line.
   - **Step Tag:**
3. **Example:**

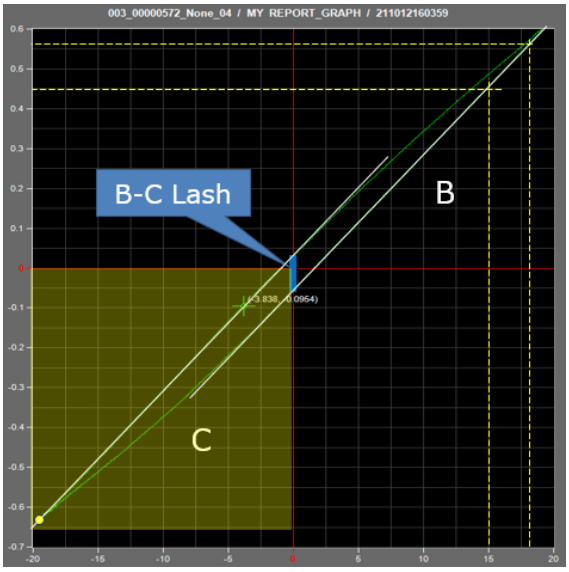
Here is an example graph. DAQ-LVDT_A and DAQ-LVDT_D graphs represent the positive direction lash

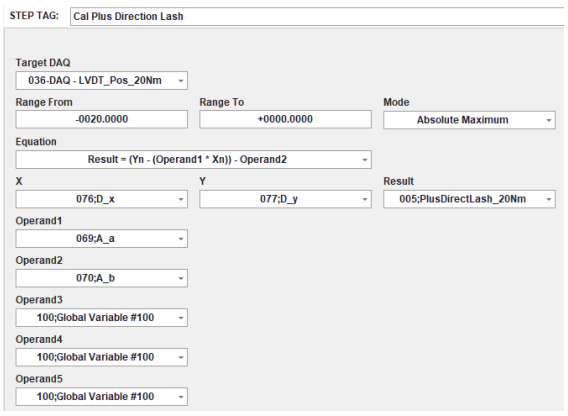
measurement and DAQ-LVDT_B and DAQ-LVDT_C do negative.

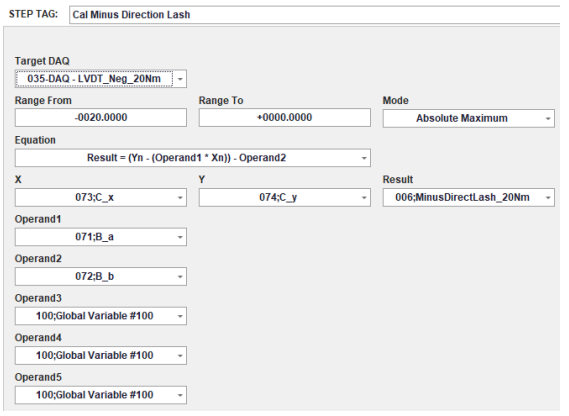


Fixed reference lines A and B are determined based on the coordinates at 15 Nm and 18 Nm. Using the slope of Line A, the system identifies the coordinates of D where the lash is maximized; similarly, Line B is used to locate C.

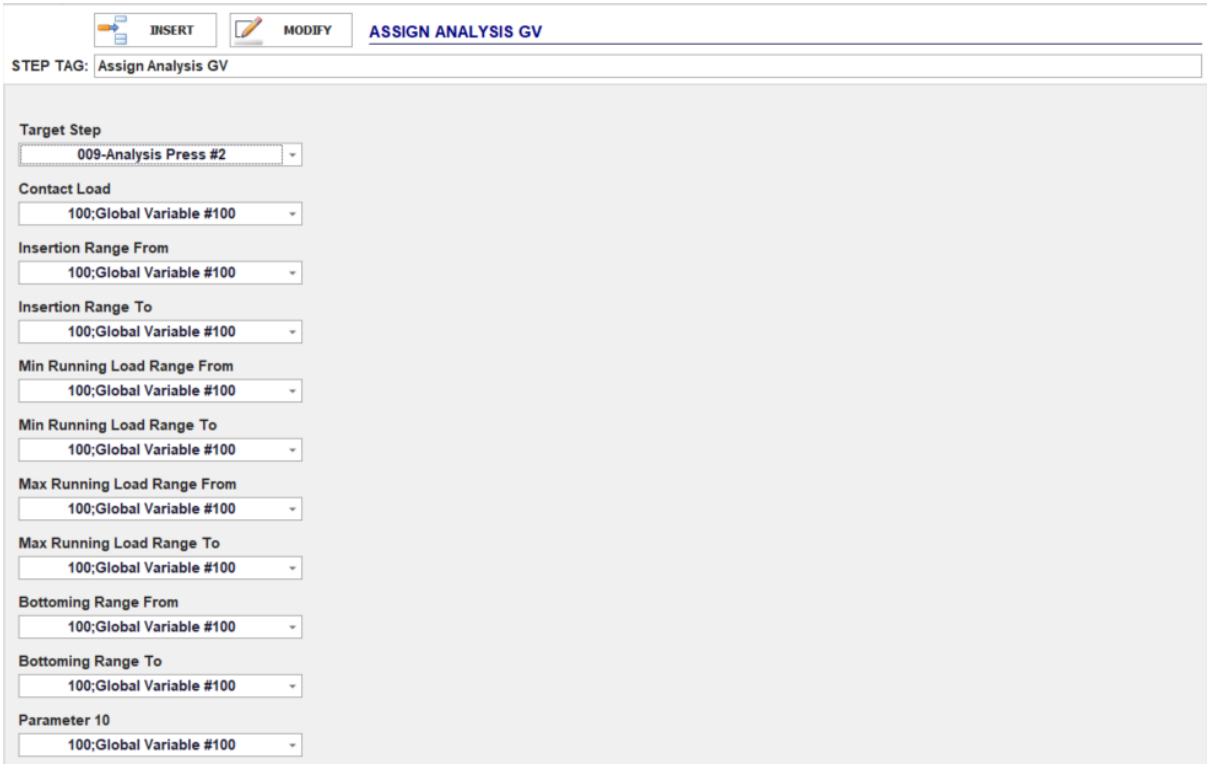








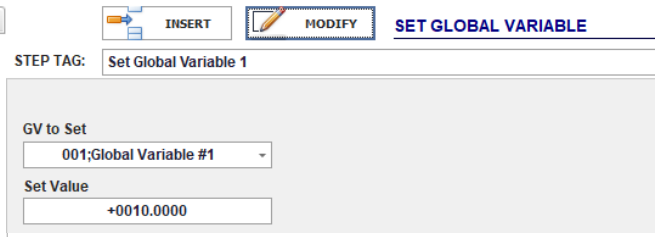
M. Assign Analysis GV



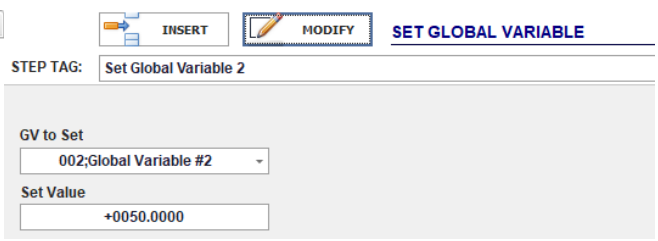
1. **Description:** It allows you to assign Analysis Parameters like “Range From” and “Range To” by using Global Variables dynamically. This step should be located just before the Target Analysis Step.
2. **Parameters:**
   - **Target Step:** Target step to assign Global Variables for the Analysis.
   - **Global Variable for Parameter #1 ~ #10:** Global Variables to be used as the Analysis Parameters of the Target Analysis.
   - **Step Tag:**
3. **Example:**

Here is an example program. There are 7 steps.
You set Global Variable1 as 10 and Global Variable as 50.

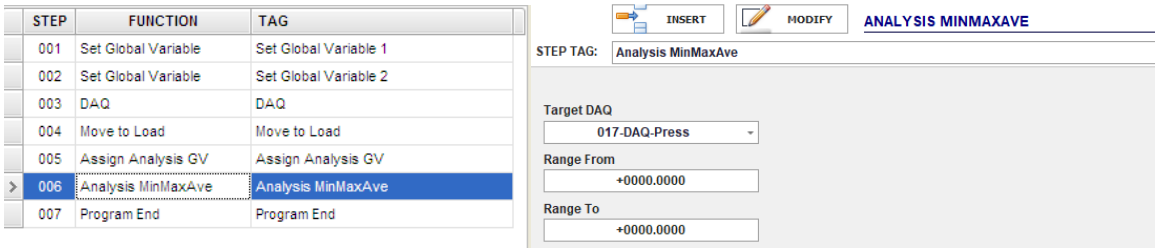
STEP	FUNCTION	TAG
001	Set Global Variable	Set Global Variable 1
002	Set Global Variable	Set Global Variable 2
003	DAQ	DAQ
004	Move to Load	Move to Load
005	Assign Analysis GV	Assign Analysis GV
006	Analysis MinMaxAve	Analysis MinMaxAve
007	Program End	Program End



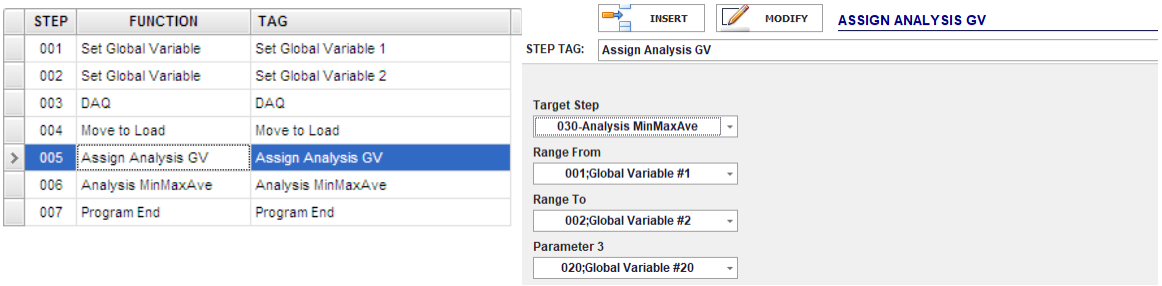
STEP	FUNCTION	TAG
001	Set Global Variable	Set Global Variable 1
002	Set Global Variable	Set Global Variable 2
003	DAQ	DAQ
004	Move to Load	Move to Load
005	Assign Analysis GV	Assign Analysis GV
006	Analysis MinMaxAve	Analysis MinMaxAve
007	Program End	Program End



Add “Analysis MinMaxAve” function with “Analysis Range From” and “To” set as zero.

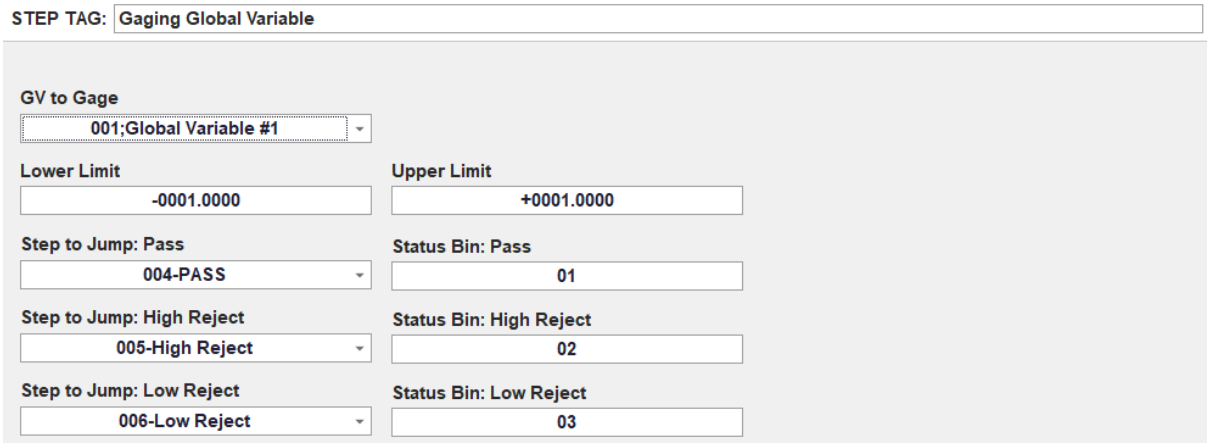


You added “Assign Analysis GV” just before above “Analysis MinMaxAve” step. And select the “Analysis MinMaxAve” as Target Analysis. When you select the Target Analysis, you can see the Global Variable for Parameters Name Tags are changed to proper name. And by assigning “Global Variable #1” and “Global Variable #2” as the analysis range of the Target Analysis step, the “Analysis MinMaxAve” step will analyze within 10 to 50.



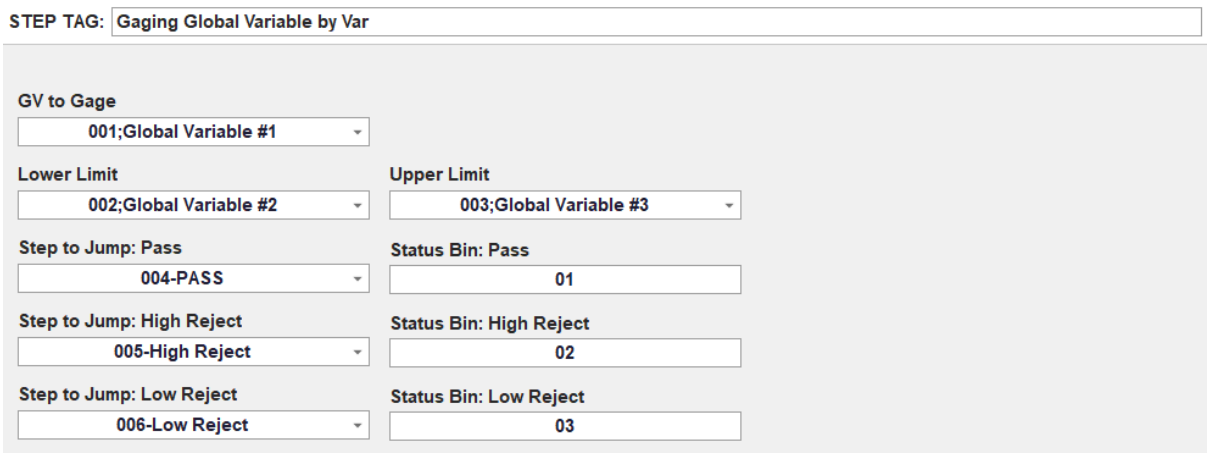
VII.GAGE

A. Gaging Global Variable



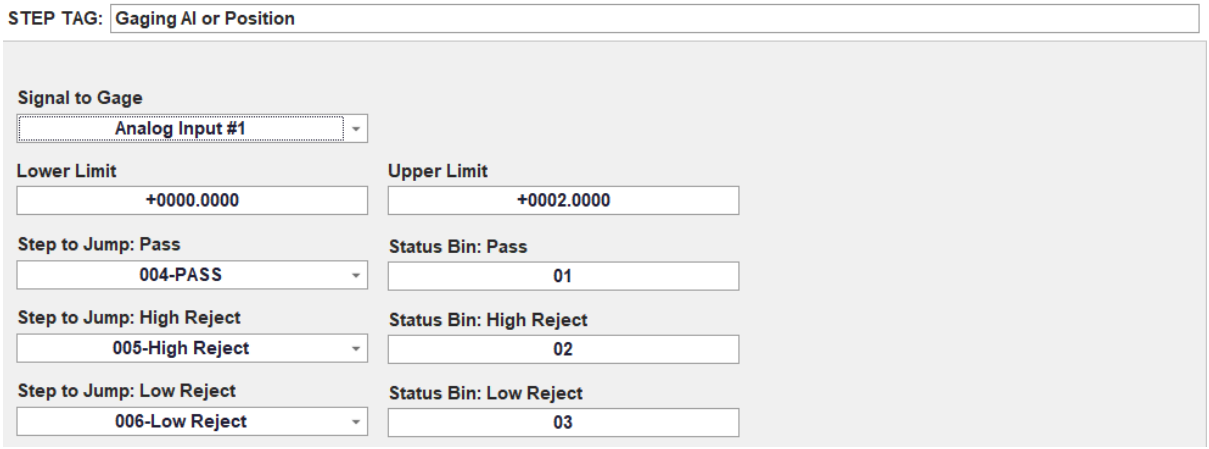
1. **Description:** Evaluate the selected global variable based on the lower limit and upper limit. Then set the Status Binary and jump to the selected step.
2. **Parameters:**
   - **GV to Gage:** The Global Variable to evaluate.
   - **Lower Limit:** Lower Limit to pass.
   - **Upper Limit:** Upper Limit to pass.
   - **Step to Jump: Pass**: In the case of that the selected Global Variable is no smaller than Lower Limit or no greater than Upper Limit.
     - **Step to Jump**: Target Step to Jump in the case of “Pass”
     - **Status Bin: Pass:** New value of the Status Bin before jump to the target step. If you input -1, jump to the target step without changing the Status Bin.
     - **Ex:** On the above picture, if the value is 0, it jumps to step 4 and set Status Binary as 1.
   - **Step to Jump: High Reject**: In the case of that the selected Global Variable is greater than Upper Limit.
     - **Step to Jump**: Target Step to Jump in the case of “High Reject”
     - **Status Bin: High Reject:** New value of the Status Bin before jump to the target step. If you input -1, jump to the target step without changing the Status Bin.
     - **Ex:** On the above picture, if the value is 2, it jumps to step 005 and set Status Binary as 2.
   - **Step to Jump: Low Reject**: In the case of that the selected Global Variable is smaller than Lower Limit.
     - **Step to Jump**: Target Step to Jump in the case of “Low Reject”
     - **Status Bin: Low Reject:** New value of the Status Bin before jump to the target step. If you input -1, jump to the target step without changing the Status Bin.
     - **Ex:** On the above picture, if the value is -2, it jumps to step 006 and set Status Binary as 3.
   - Step Tag:

B. Gaging Global Variable by Var



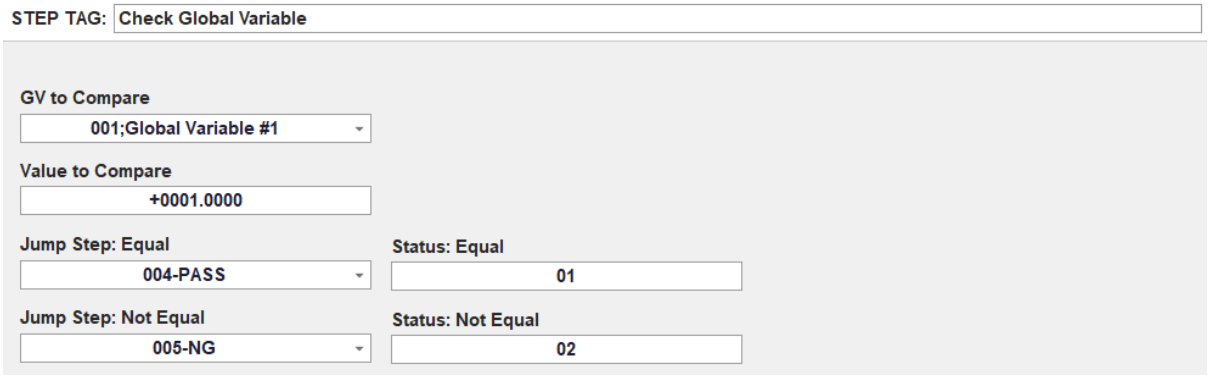
1. **Description:** It is same as “Gaging Global Variable” except that the Lower Limit and Upper Limit are to be assigned by the selected global variables.(refer to the “Gaging Global Variable” for details)

C. Gaging AI or Position



1. **Description:** Evaluate the current value of the selected Analog Input Channel or Axis Position. The other function is same as “Gaging Global Variable”.

D. Check Global Variable



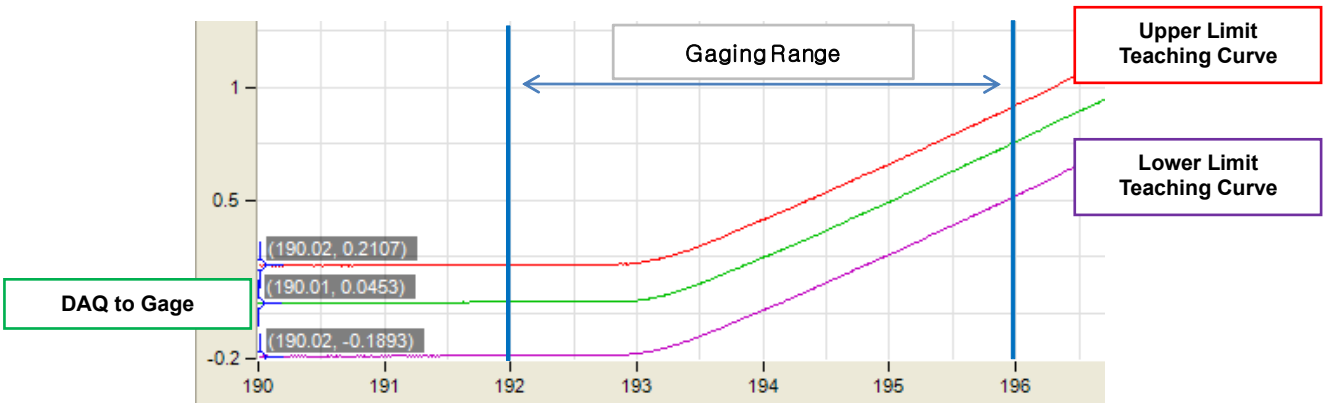
1. **Description:** Compare the selected global variable with custom input value then set the Status Binary and jump to the selected step.
2. **Parameters:**
   - **GV to Compare:** The Global Variable to compare.
   - **Value to Compare:** Custom input value to be compared with the selected global variable.
   - **Jump Step: Equal**: The case of that the selected Global Variable is equal to the custom input value.
     - **Step to Jump**: Target Step to Jump in the case of “Equal”
     - **Status: Equal:** New value of the Status Bin before jump to the target step. If you input -1, jump to the target step without changing the Status Bin.
   - **Jump Step: Not Equal**: The case of that the selected Global Variable is different from the custom input value.
     - **Step to Jump**: Target Step to Jump in the case of “Not Equal”
     - **Status: Not Equal:** New value of the Status Bin before jump to the target step. If you input -1, jump to the target step without changing the Status Bin.
3. **Step Tag:**

E. Gaging DAQ by Teaching

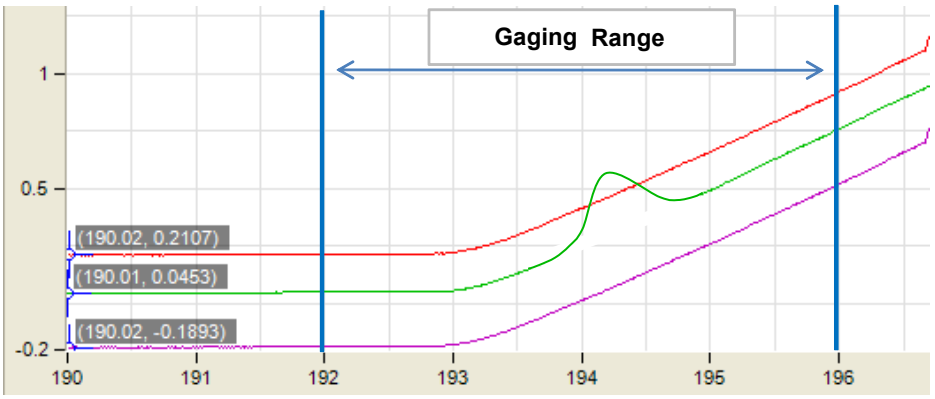
STEP TAG: Gaging DAQ by Teaching

DAQ to Gage: 004-DAQ	
Lower Limit: 01;None	Upper Limit: 02;None
Range From: +0000.0000	Range To: +0010.0000
Step to Jump: Pass: 005-PASS	Status Bin: Pass: 01
Step to Jump: High Reject: 008-High Reject	Status Bin: High Reject: 02
Step to Jump: Low Reject: 009-Low Reject	Status Bin: Low Reject: 03
Step to Jump: Both Reject: 007-Out of Range	Status Bin: Both Reject: 04

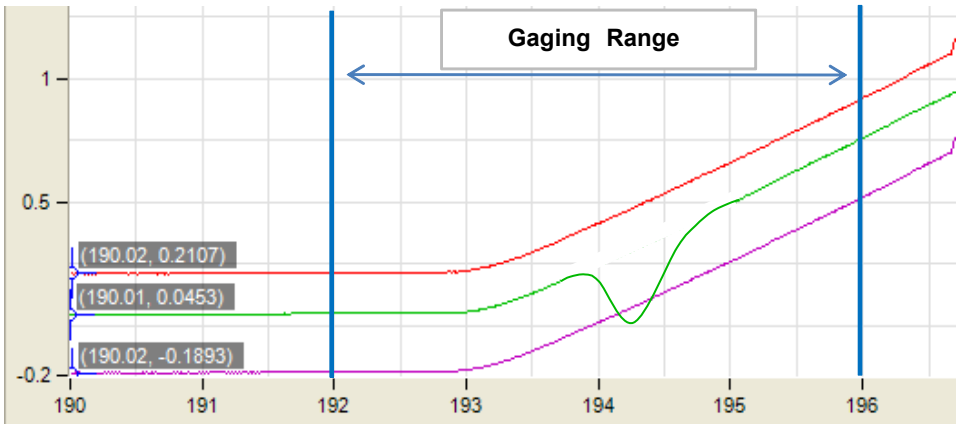
1. **Description:** It evaluates the selected DAQ based on the lower limit Teaching Curve and the upper limit Teaching Curve. Then, it sets the Status Binary and jumps to the selected step. You can gage the whole DAQ curve within the specified range not just one point by using this function. Lower Limit Teaching Curve and Upper Limit Teaching Curve make a window to evaluate the selected DAQ curve.
2. **Parameters:**
   - **DAQ to Gage:** The DAQ to evaluate.
   - **Lower Limit:** Lower limit teaching curve to pass. “00:No Limit” option is also available.
   - **Upper Limit:** Upper limit teaching curve to pass. “00:No Limit” option is also available.
   - **Range From / To:** The range to be evaluated.
   - **Step to Jump: Pass**:



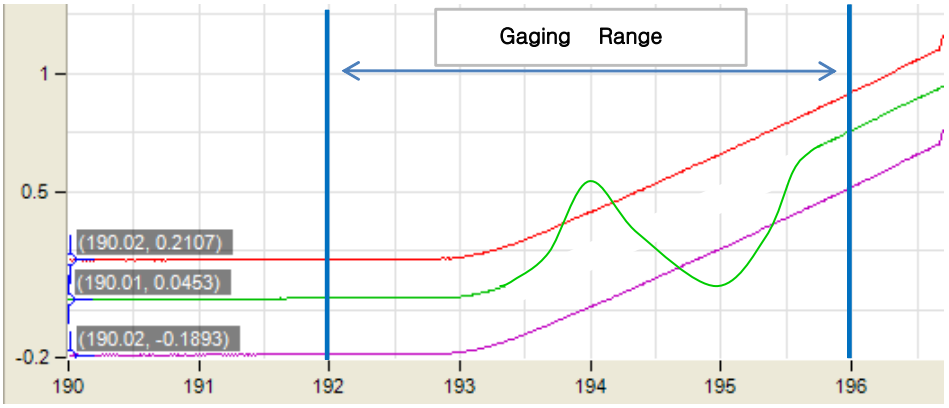
- **Step to Jump: High Reject:**



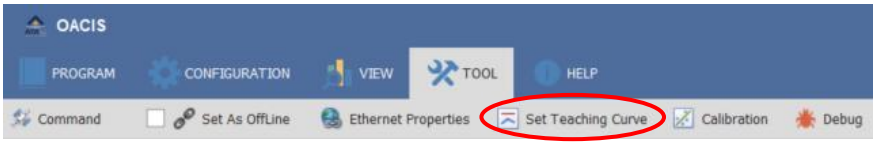
- **Step to Jump: Low Reject:**

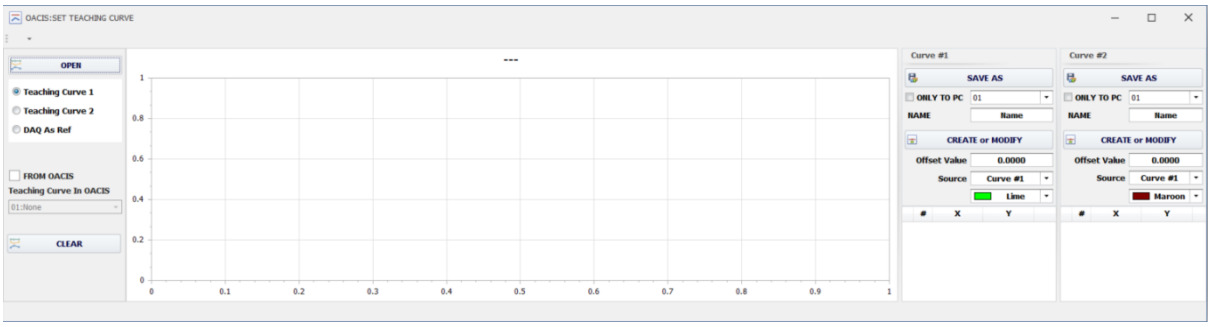


- **Step to Jump: Both (High and Low) Reject:**

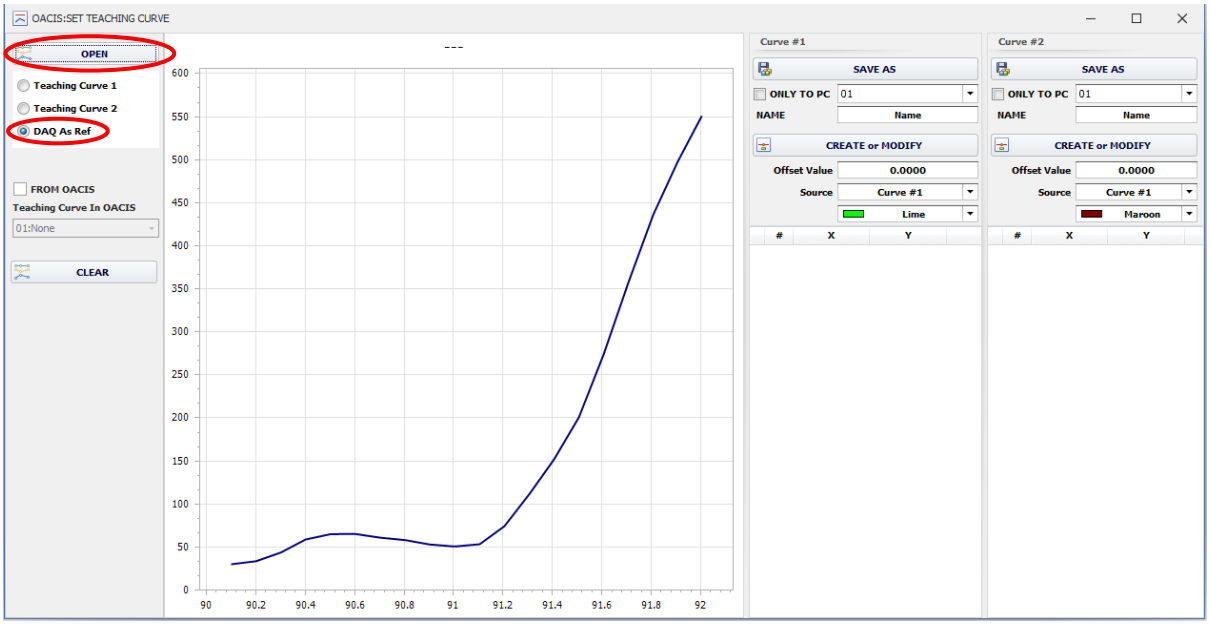


3. **How to Make Teaching Curve:** Here we describes on “How to make Teaching Curves” that would be used “Gaging DAQ by Teaching” function.
   - Save Reference DAQ curve with *.gph file format by running the proper cycle.
   - Open “TEACHING DAQ CURVE” window by clicking the menu. It requires Password input.

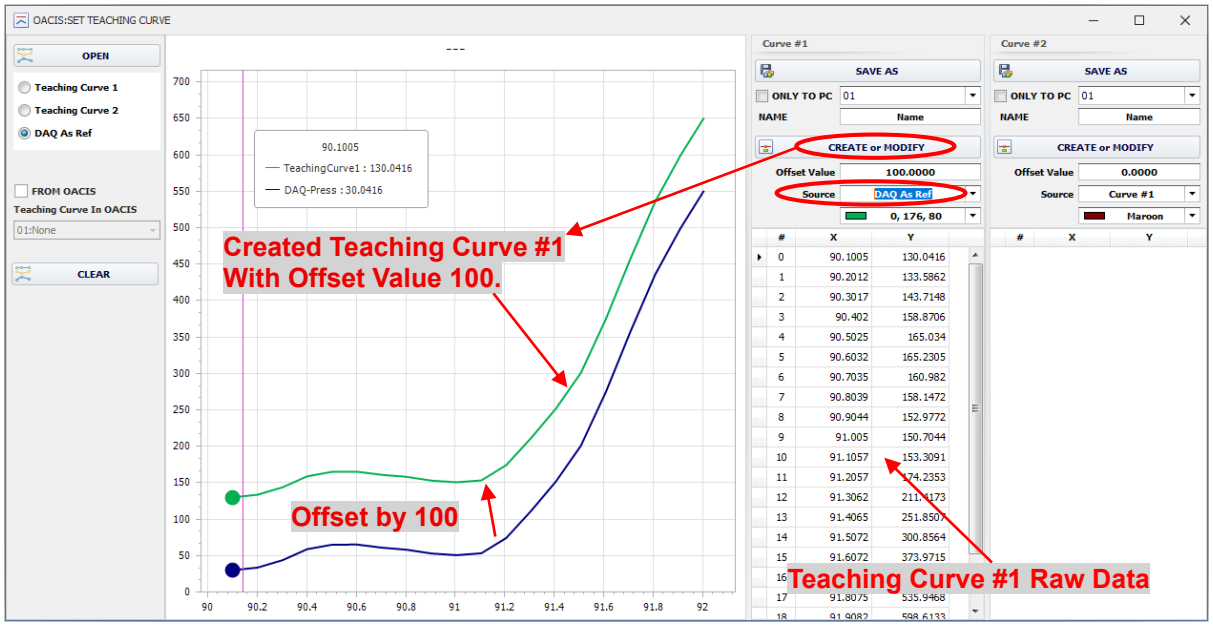




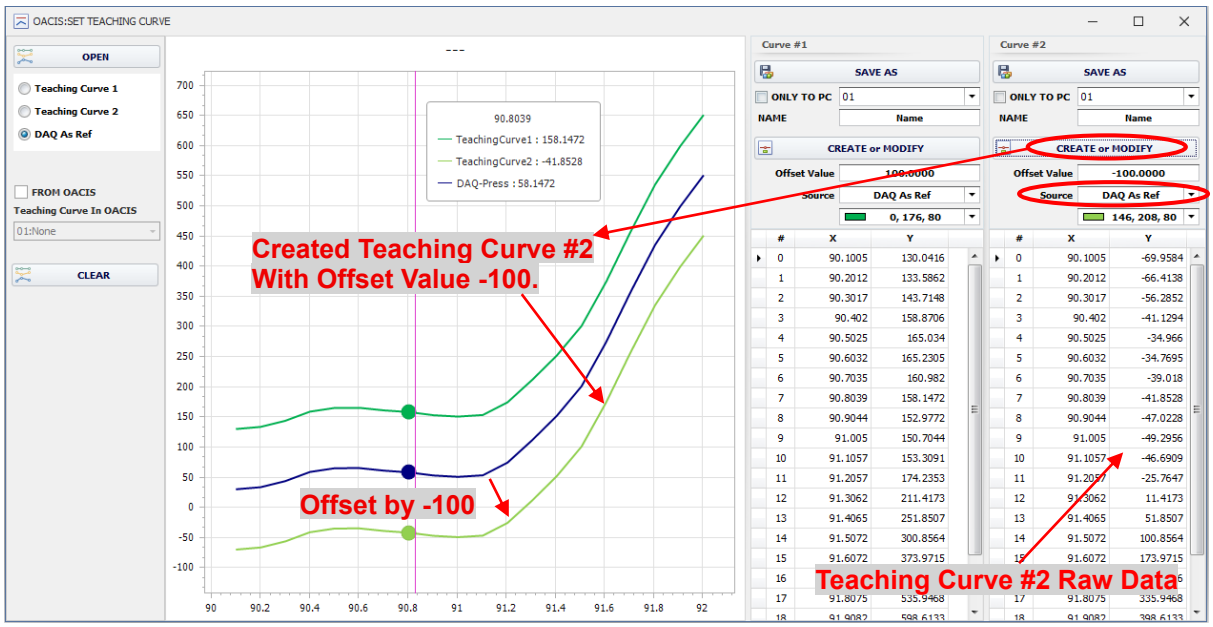
- Open DAQ as Reference Curve (From PC)



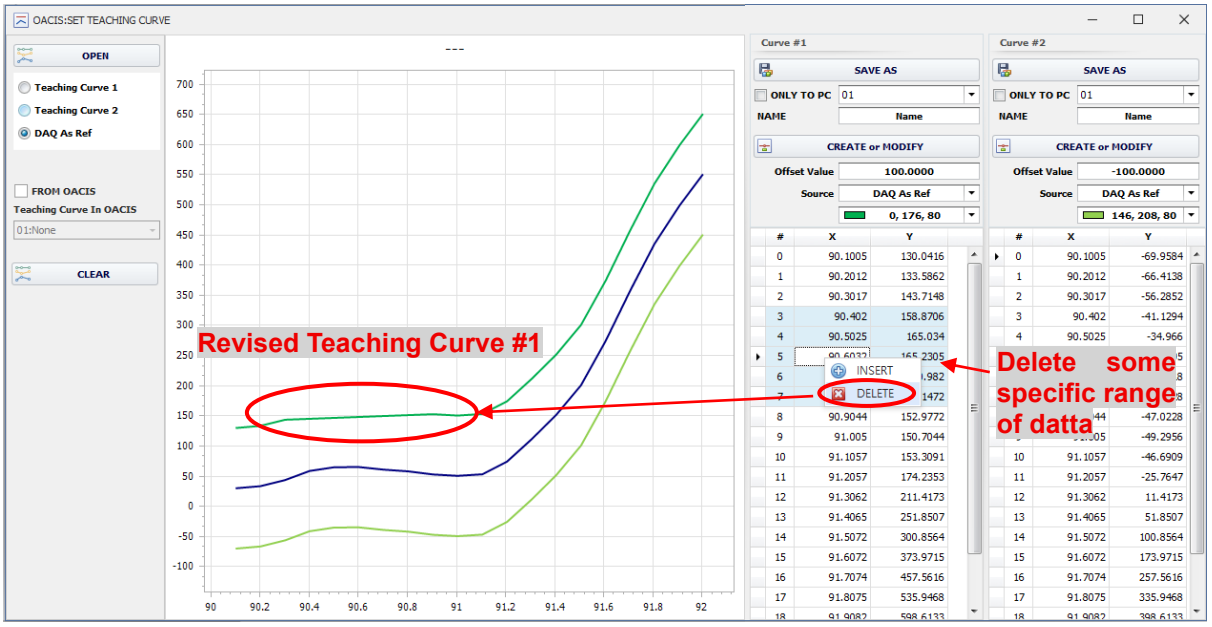
- Create Teaching Curve #1



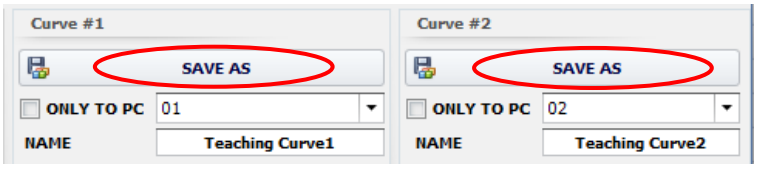
- Create Teaching Curve #2



- Modify (delete / insert / change) the Raw Data manually as well.



- Save Teaching Curve #1 and #2 as the assigned Teaching Curve Number and Name.



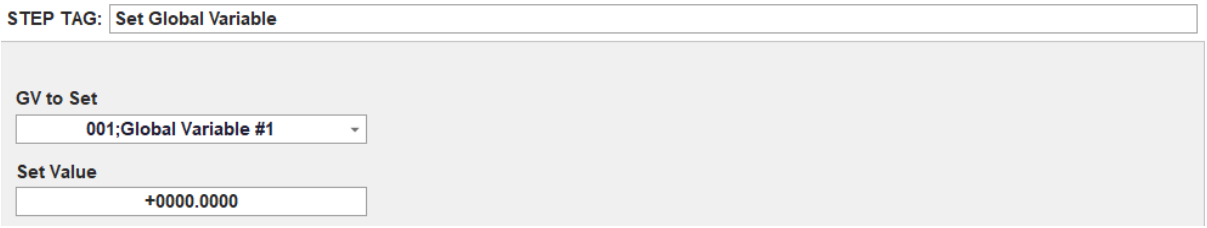
- Then you can use the saved Teaching Curve at the “Gaging DAQ by Teaching” function.

VIII. MATH

A. Reset All Global Variables

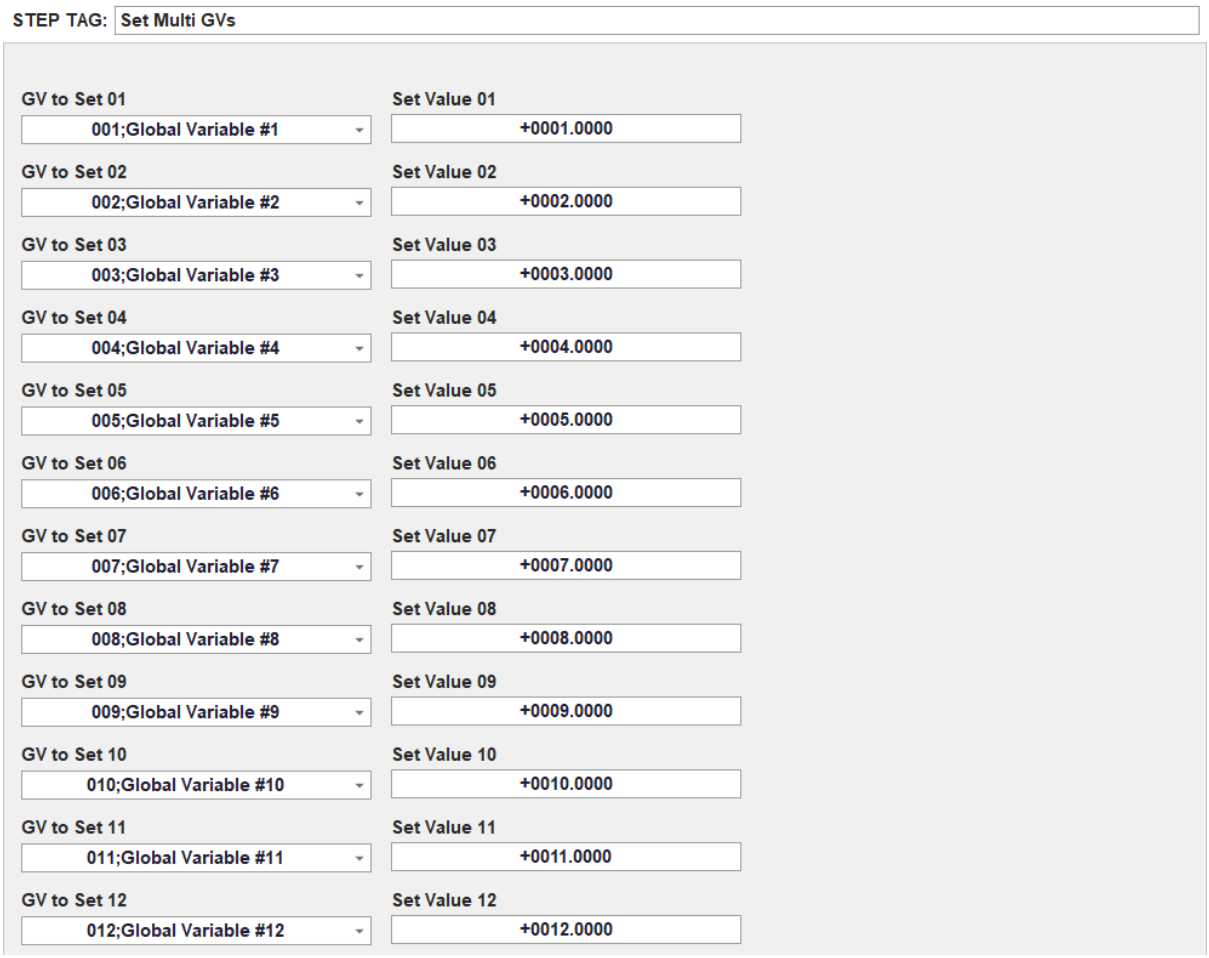
1. **Description:** Reset All Global Variables (excluding “System Variables”) as Zero.
2. **Parameters:**
   - **Step Tag:**

B. Set Global Variable



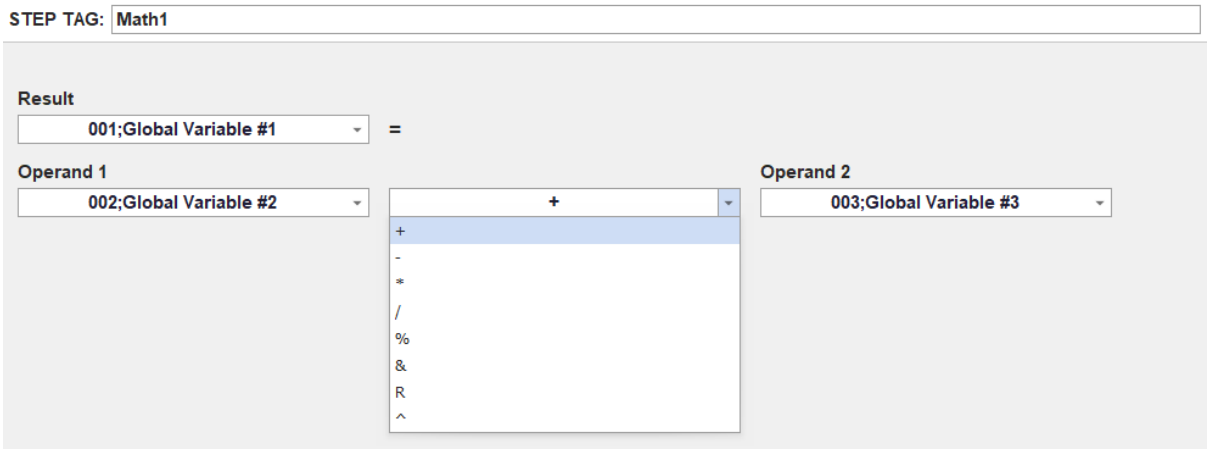
1. **Description:** Set the selected Global Variable as the input value.
2. **Parameters:**
   - **Global Variable to Set**: Global Variable to set as new value.
   - **Set Value:** New value for the selected Global Variable.
   - **Step Tag**:

C. Set Multi GVs



1. **Description:** It is same as “Set Global Variable” except that you can set a number of GVs at the same time. It is useful to classify GVs such as Constants, Gaging Limits and Target Positions.
2. **Parameters:**
   - **GV to Set**: Global Variables to set as new values. You can set Max. 12 GVs.
   - **Set Value:** New values for the selected Global Variables.
   - **Step Tag**:

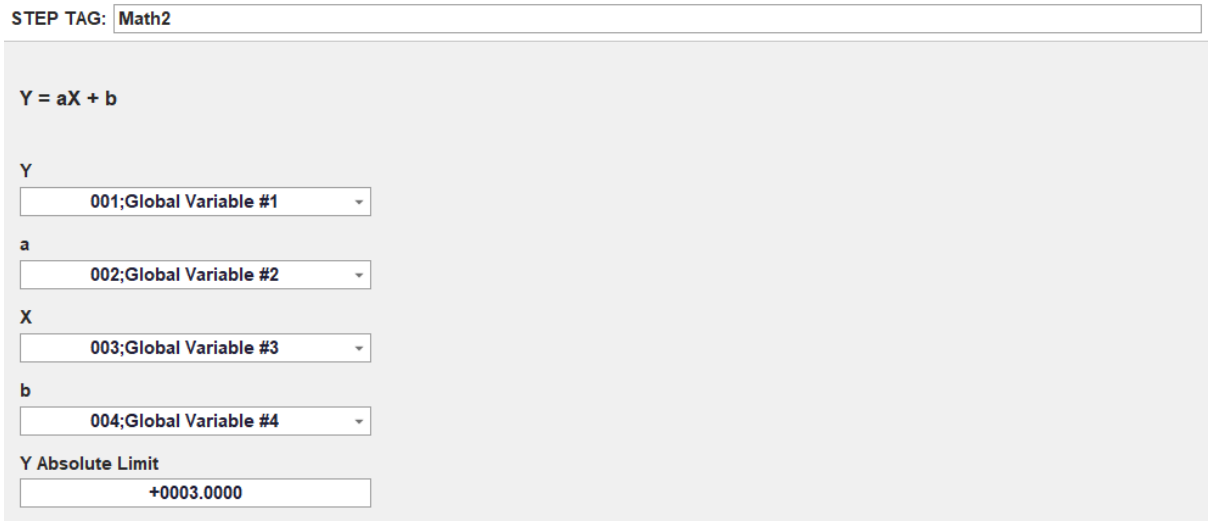
D. Math1



1. **Description:** You can do the four fundamental arithmetic operations by using this function. You need to select three Global Variables and one of the eight operations.

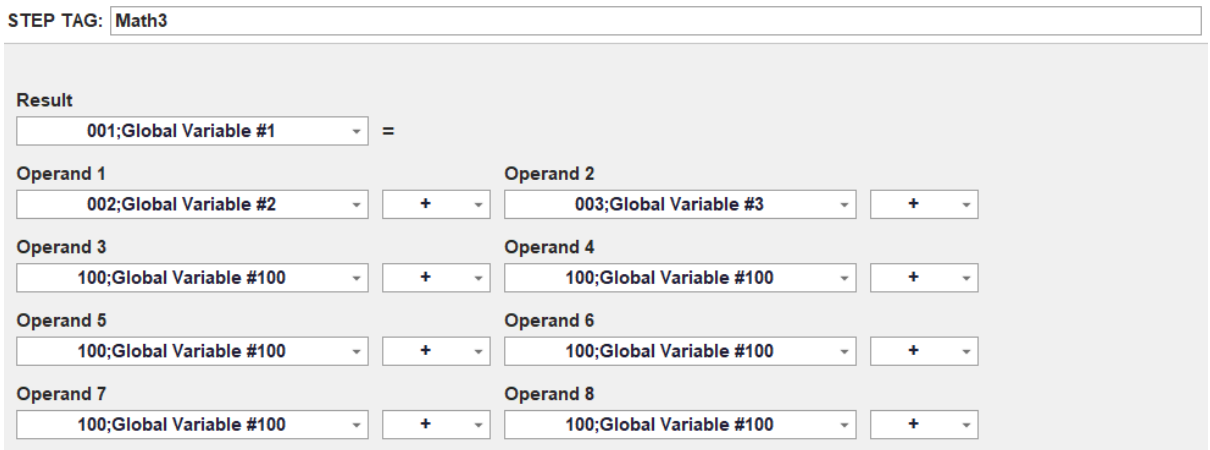
2. **Parameters:**
   - **+, -, *, /, %**(remainder), **&**(quotient), **R**(root), **^**(square).
3. **Example:**
   - **IF**
     - Global Var#2 = 2
     - Global Var#3 = 3
   - **The results are the same as below,**
     - GV1 = GV2 % GV3 = 2 % 3 = **2**
     - GV1 = GV2 & GV3 = 2 & 3 = **0**
     - GV1 = GV2 R GV3 = 2 R 3 = **1.2599**
     - GV1 = GV2 ^ GV3 = 2 ^ 3 = **8**

E. Math2



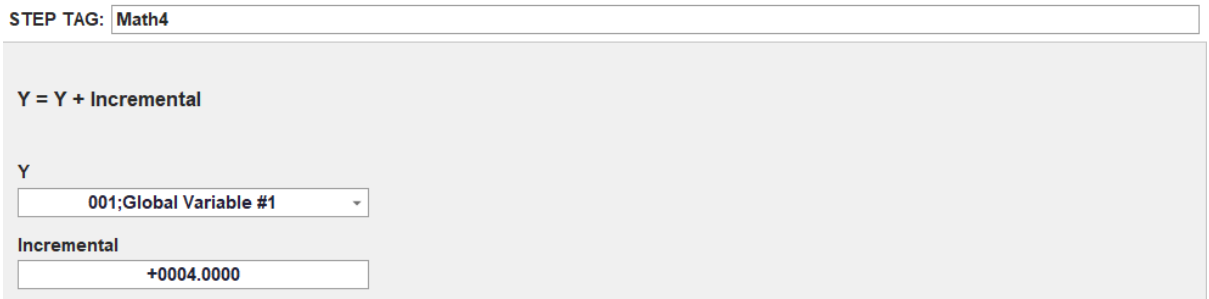
1. **Description:** You can do calculate an equation of the first degree like Y = aX + b. You need to select each global variable for Y, a, X and B.
   - **Example:** Global Var#1 = Global Var#2 x Global Var#3 + Global Var#4.
   - **Y Absolute Limit:** The result can't be over "Y Limit". If the result is 5 and "Y Absolute Limit" is 4, this function returns 4 instead of 5.

F. Math3



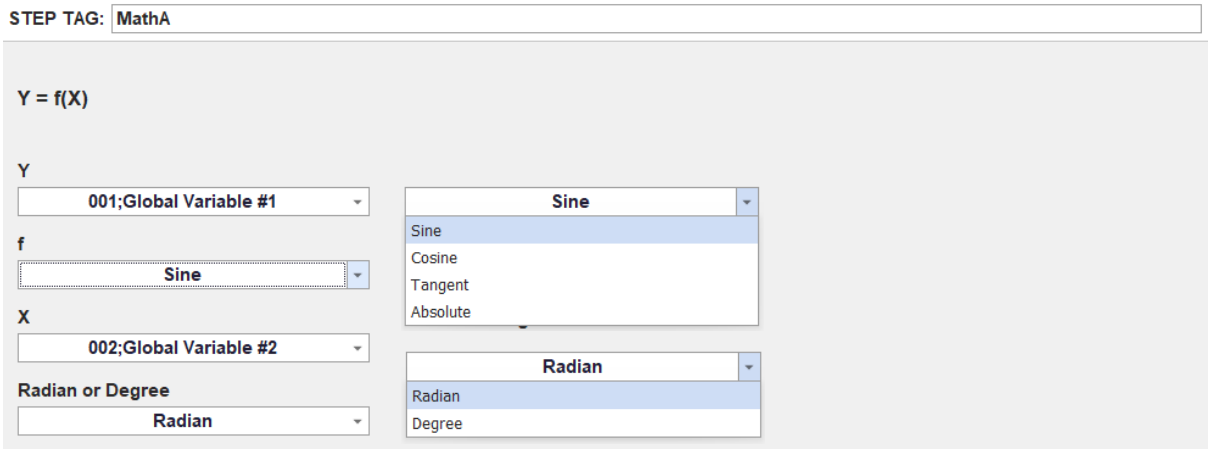
1. **Description:** You can do calculate with 10 operands and 9 operators.
2. **Parameters:**
   - **+, -, *, /, %**(remainder), **&**(quotient), **R**(root), **^**(square).
3. **Example:**
   - **IF**
     - Global Var#2 = 2
     - Global Var#3 = 3
     - …
     - Global Var#11 = 11
   - **And If you set as below,**
     - GV1 = GV2 + GV3 * GV4 - GV5 + GV6 + GV7 + GV8 + GV9 + GV10 + GV11
     - GV1 = 2 + 3 x 4 - 5 + 6 + 7 + 8 + 9 + 10 + 11 = **66**
     - OACIS returns **66** for Global Var#1.
     - It is calculating sequentially. It does now follow general mathematics rules.
     - 2 + 3 = 5 -> 5 x 4 = 20 -> 20 – 5 = 15 -> 15 + 6 + 7 + 8 + 9 + 10 + 11 = **66**
   - **And If you set as below,**
     - GV1 = GV2 R GV3 ^ GV4 % GV5 * GV6 & GV7 + GV8 / GV9 + GV10 + GV11
     - GV1 = 2 R 3 ^ 4 % 5 * 6 & 7 + 8 / 9 + 10 + 11 = **22**
     - OACIS returns **22** for Global Var#1.
     - 2 R 3 = 1.2599 -> 1.2599 ^ 4 = 2.5198 -> 2.5198 % 5 = 2 -> 2 * 6 = 12 -> 12 & 7 = 1 -> 1 + 8 = 9 -> 9 / 9 = 1 -> 1 + 10 + 11 = **22**

G. Math4



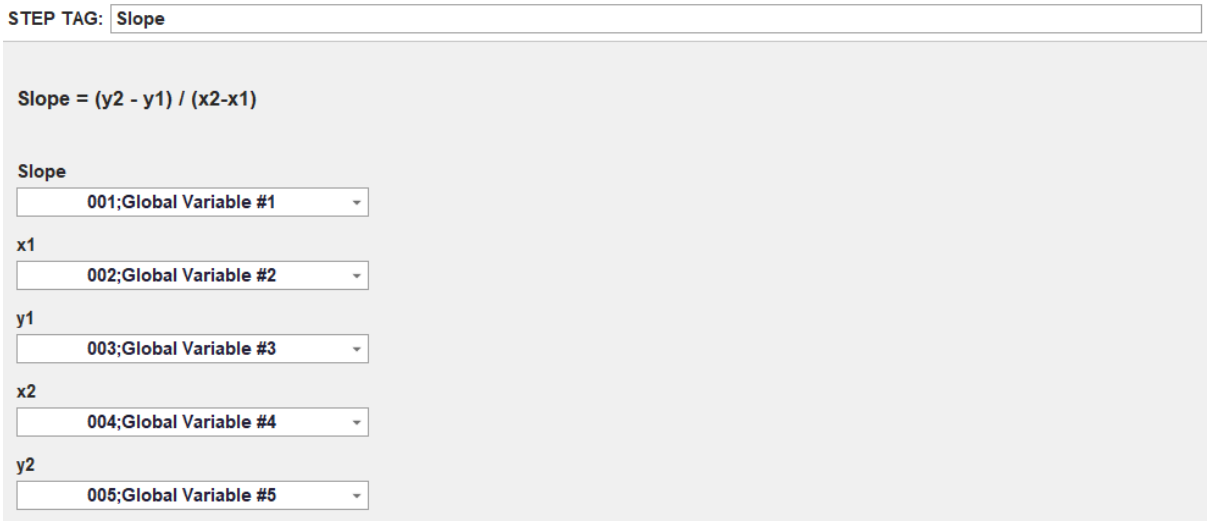
1. **Description:** You can easily increase a global variable as many as the incremental value. It is the same as GV #1 = GV #1 + Incremental Value.
2. **Parameters:**
   - Incremental value could be Positive or Negative.
3. **Example:**
   - If GV #1 =3, Incremental Value = -1, then GV #1 = 2 for the first trial, GV #1 =1 for the second trial and GV #1 = 0 for the third trial.

H. MathA



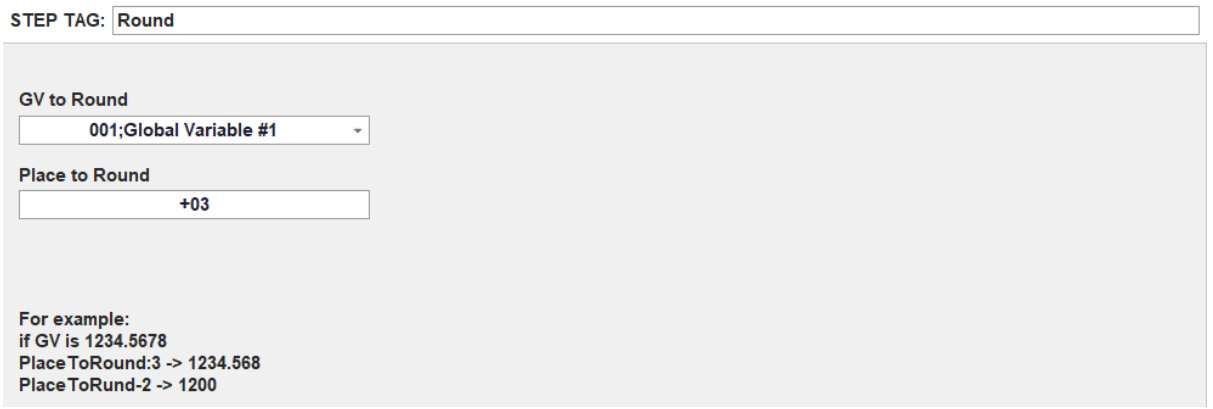
1. **Description:** You can calculate trigonometric and absolute function with a radian or degree unit.
2. **Parameters**
   - **Sine / Cosine / Tangent / Absolute**
3. **Example:**
   - If you select Sine, Radian and GV #2 =1, then OACIS returns 0.8415 in GV #1.
   - If you select Cosine, Degree and GV #2 = 360, then OACIS returns 1 in GV #1.
   - If you select Absolute and GV #2 = -2, then OACIS returns 2 in GV #1.

I. Slope



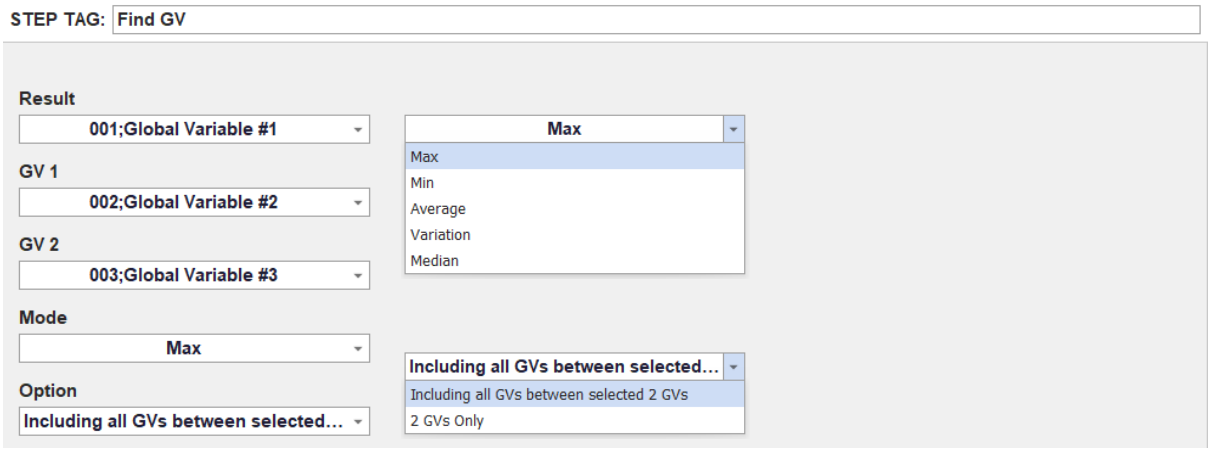
1. **Description:** You can easily calculate the slope between two points. Of course, you can also find the slope by using other Math functions.
2. **Example:**
   - If (x1, y1) =(1, 1) and (x2, y2) =(2, 3), then OACIS returns 2 in GV #1.

J. Round



1. **Description:** You can round a real number according to place to round.
2. **Example:**
   - If GV #1 = 1234.5678 and Place to Round = 3, then OACIS returns 1234.568 in GV #1.
   - If GV #1 = 1234.5678 and Place to Round = 0, then OACIS returns 1235 in GV #1.
   - If GV #1 = 1234.5678 and Place to Round = -2, then OACIS returns 1200 in GV #1.

K. Find GV



1. **Description:** You can find a specific value to be satisfied with one of the five modes in the range of global variable selection option.
2. **Parameters:**
   - **Mode:**
     - **Max:** Maximum Value of the range.
     - **Min:** Minimum Value of the range.
     - **Ave:** Average Value of the range.
     - **Variation:** Difference between Max and Min of the range
     - **Median:** Value of the mid GV in the range. In case of Two Values Only in GLOBAL VARIABLE SELECTION OPTION, OACIS returns average value of the two GVs. In case of Including All Values between Two Values, OACIS returns value of the mid GV if the number of GVs in the range is odd and returns average value of the two mid GVs if even.
   - **Option:**
     - **2 GVs Only:** Selected two GVs. It means that the number of valid GVs is 2.
     - **Including all GVs between selected 2 GVs:** All values between the selected two GVs. The number of valid GVs can be more than 2.
3. **Example:**
   - Assume that GV#10=10, GV#11=11, GV#12=20, GV#13=13, GV#14=14, GV#15=15.
     - If you select “GV#10” in GLOBAL VARIABLE #1, “GV#15” in GLOBAL VARIABLE #2 and **“Max”** mode, then the result is **15** in “Two Values Only” option and **20** in “Including All Values Between Two Values.
     - If you select “GV#10” in GLOBAL VARIABLE #1, “GV#15” in GLOBAL VARIABLE #2 and **“Variation”** mode, then the result is **5** in “Two Values Only” option and **10** in “Including All Values Between Two Values.
     - If you select “GV#10” in GLOBAL VARIABLE #1, “GV#14” in GLOBAL VARIABLE #2 and **“Median”** mode, then the result is **12.5** in “Two Values Only” option and **20** in “Including All Values Between Two Values.

IX. FIELDBUS

A. Capture Serial From Fieldbus

STEP TAG: Get Serial From Fieldbus

1. **Description:** OACIS reads serial numbers stored in ASCii at this function step.
2. **Parameters:**
   - **Step Tag:**

REVISION

v1.00: Engineering Released

v1.40:

- **Added “Assign Analysis GV” function (VI.I)**

v1.41:

- **Added “Move to Load by Var. #2” function (II.M)**
- **Added “Set AI or Position by Var.” function (III.H)**
- **Added “Set As Abs Value by Var.” function (III.I)**

v1.51:

- **Added “Move to Position with Limited Load” function (II.N)**
- **Added “Send Out Data” function (III.J)**
- **Added “DAQD” function (V.D)**
- **Added “Linear Regression #2” function (VI.J)**
- **Added “Math3” function (VIII.E)**

v1.52:

- **Added “DAQA” function (V.E)**

v1.53:

- **Added “Start Hold Load” / “End Hold Load” functions (II.O)**

v1.54:

- **Added “Move to Position by Var #2” function (II.H)**
- **Added “CAPTURE” function (V.F)**
- **Added “Find Point” function (VI.K)**

v1.55:

- **Updated “Document Format”**

v1.56:

- **Updated “Document Format”.**

v1.57:

- **Updated “All function images” (II, III, IV, V, VI, VII, VIII)**
- **Added “Two Error and Stop buttons in Move to AI” (II.K)**
- **Added “Jump by Condition” function (IV.D)**
- **Added “Jump by Multi Conditions” function (IV.E)**
- **Added “Acceptable Min. Sampling Rate in DAQ” (V.B)**
- **Added “Acceptable Min. Sampling Rate in DAQ2” (V.C)**
- **Added “Estimated Samples Count in DAQA” (V.E)**
- **Added “Analysis Press #2” function (VI.D)**
- **Updated “Gaging DAQ by Teaching” function (VII.E)**
- **Added “%&R^ in Math1” (VIII.C)**
- **Added “%&R^ in Math3” (VIII.E)**
- **Added “Math4” function (VIII.F)**
- **Added “MathA” function (VIII.G)**
- **Added “Slope” function (VIII.H)**
- **Added “Round” function (VIII.I)**
- **Added “Find GV” function (VIII.J)**

v1.58:

- Updated “Move to Position with Limited Load” function (II.O)
- Added “Deactivate” function (II.Q)

v1.59:
- Updated “Headers & Footers” format

v1.60:
- Updated “Image Size & Resolution”

v1.61:
- Added “Count DI” function (V.G)

v1.62:
- Added “Jump by Condition #2” function (IV.E)
- Added “Set Multi GVs” function (VIII.C)

v1.63:
- Revised “All Damaged Images”

v1.64:
- Updated “Start Hold Load / End Hold Load” function (II.P)

v1.65:
- Updated “Analysis Press #2” function (VI.D)

v1.66:
- Revised “CAPTURE” function (V.F)

v1.67:
- Added “Wait to Pause” function (IV.K)
- Downsized “All contents”

v1.68:
- Added “Move to Load by Var #3” function (II.R)

v1.69:
- Modified “Move to AI” function (II.K) Typo
- Added “Fieldbus In & Out” (I.C)
- Added “Capture Serial From Fieldbus” function (IX.A)

v1.70:
- Added “Caution” (II. I)

v1.71:
- Page format Updated

v1.72:
- Specific Load mode Added in Dynamic Move to Position (II. I)

v1.73:
- Analysis Load Drop Updated in Analysis (VI.I)
- Specific Load mode Updated in Dynamic Move to Position (II. I)
- Added “Move to Load by Var #4” function (II.R)

v2.01:
- Updated OACIScom v5

v2.02:
- Modified “Minor Errors”
- Added “Analysis With Equation” function (VI.L)

v05.00.01:
- Updated New Version

v05.00.02:
- Updated Old Images