

# Derived POINTs and their Application

By Joost Boerhout • SKF USA Inc., Condition Monitoring Center – San Diego

## Requirements

Derived POINTs are a feature of all SKF @ptitude Analyst software platforms. However, there are differences in the total derived POINT functionality provided to you based on the software platform installed. The CMSW 7400 platform is the most powerful and provides you all derived POINT functionality (refer to the “Derived POINT functions” section in this article for more detail). This article assumes the reader is very familiar with SKF @ptitude Analyst concepts, such as the hierarchy, machine and POINT nodes, manual node types, etc.

## Abstract

Derived POINTs are intended to compute specific results based on data collected through any of SKF’s machinery monitoring or inspection devices. For example, you might collect a pump’s inlet and outlet pressures by means of two appropriate POINTs. A derived POINT can then use the collected pressure data to compute a pump efficiency index. In another case, you might wish to view an average of vibration for a machine, or an average across a set of measurements. In this way, derived POINTs are very similar to cell “macros” in Microsoft Excel.

## What are derived POINTs?

A standard POINT identifies a machinery measurement’s location and measurement type. Measurement data is then recorded for the POINT using one of SKF’s data acquisition devices. For example, a SKF Microlog Analyzer recording a vibration measurement’s FFT data or a SKF Microlog Inspector/SKF MARLIN recording a gauge’s process data.

A derived POINT outputs data from a formula that uses previously collected measurement POINT data as part of its calculation. A derived POINT’s input – is the output from other POINTs! A derived POINT post processes collected data and transforms it by means of a user programmable formula into an output which can be trended and alarmed upon.

For example, you may wish to record fluid pressure readings from a pipe at two different locations. Assume that data would be collected by a SKF Microlog Inspector/SKF MARLIN device whereby the operator enters the values of two gauges, and records the readings as Pressure A and Pressure B. Once the recorded data is uploaded to SKF @ptitude Analyst, the trends for these two locations can be reviewed. But what if you wanted to view and trend the pressure *difference* between the two POINTs? With SKF @ptitude Analyst, you would create new third POINT of type “derived POINT.” The derived POINT would associate the Pressure A and Pressure B inputs, and compute a new output value simply by describing the required formula (i.e., output = Pressure A – Pressure B). Over time, the derived POINT’s output value provides a new trend of the pressure difference, and allows an appropriate alarm value to be specified.

Derived POINTs can create, trend, and alarm new values that aggregate information from your data collectors that are closely aligned with key parameters on which you need to make decisions.

## Supported POINT types

A derived POINT’s input or set of inputs can be created from any other POINT in the SKF @ptitude Analyst hierarchy, including:

- Any vibration POINT (acceleration, velocity, displacement, SEE, acceleration enveloping, etc.)
- Any process POINT (DC values, pressure, speed, etc.)
- SKF Microlog Inspector/SKF MARLIN inspection POINTs (both Single and Multi Selection are supported)
- OPC POINTs
- XML POINTs
- Other derived POINT output (referred to as a “nested” derived POINT)



POINT types may have different data types. For example, an OPC POINT has an overall value, but an acceleration POINT often has spectrum and waveform data as well (dynamic data). All derived POINTs can operate on simple overall data types. In addition, through the use of derived POINT functions, dynamic data types such as spectrum and waveforms can also be used. Refer to the derived POINT function list (later in this publication) for more detail and availability in your software platform.

## Derived POINT output values

Derived POINTs have just a single output value type; the overall value. The final result of each derived POINT calculation is always a simple overall value that is stored, can be trended, and alarmed upon.

## How and when are derived POINT values calculated?

Derived POINTs are calculated automatically when a new input value presents itself. In the pressure difference example above, each time the A and B pressure values are updated, SKF @ptitude Analyst computes the new value for the pressure difference derived POINT. However, it is possible to force the calculation of the derived POINT at any time by selecting the Edit menu's Calculate Derived POINTs option. Forcing the update will update all values for a derived POINT (i.e., it will first remove the entire derived POINT's trend and rebuild it from scratch). This allows you to change a derived POINT's formula and recalculate its trend based upon the updated formula. First highlight the POINT, machine, or set you want to update, then select the Calculate Derived POINTs menu item. If you highlighted a machine or set, all derived POINTs within that selection will be updated.

In most cases, measurements are dispersed through time. That is, all POINTs are not taken at the same time. In the pressure difference example above, the operator needs to walk from the first measurement location to the second, which takes some time. In extreme cases, there might be a very considerable time lag between measurements. Also, online systems cannot take all measurements at the same time. Depending on which online system is used and how it is programmed, an hour or more between measurements is not inconceivable. This may be a problem for the validity of the derived POINT calculation. The measurements may have to be taken within a defined time window, or else the process that governs the readings may have changed to induce a considerable difference in the outcome. Setup options for Derived POINTs recognize this situation and allows you to program a window of opportunity that allows only input values collected within the specified time window to be factored into the derived POINT's calculation. Note that this does not guarantee the measurements are taken within the window (you will have to do this by setting up an appropriate route or measurement schedule) but it will ensure that any calculation output considers only measurements within that time window.

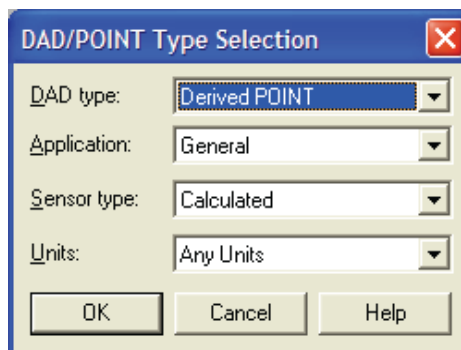
## Creating a derived POINT

To continue with the example described, you will want to create a machine node with two manual POINT types inside. Name the POINTs "Pressure A" and "Pressure B" and enter a few values. Be sure to set the date of the manual entry (i.e., separate your entries by one day or so which makes it easier to read and follow along). The values used by this article for the two POINTs are shown in **Table 1**.

**Table 1.**

Date	Pressure A (PSI)	Pressure B (PSI)
9/05/2009	10.0	9.8
9/04/2009	11.0	9.5
9/03/2009	10.0	12.0
9/02/2009	9.9	10.1
9/01/2009	10.0	9.8

Creating a derived POINT is very similar to a normal POINT; highlight a machine node in the hierarchy, right click and select Insert POINT. Alternatively you may use the Insert menu's New POINT option. In the subsequent dialog, select DAD type as "Derived POINT." For most applications, all other fields in that dialog can be left to the default setting, refer to **Figure 1**.



**Figure 1. Inserting a Derived POINT.**

When the DAD/POINT Type selection is saved, SKF @ptitude Analyst displays the standard "POINT Properties" dialog with the General tab open. This example will build the derived POINT for the pressure difference exemplified above, so we will name our derived POINT "Pressure Difference."

The setup dialog for the derived POINTs has one new tab; Expressions. This is where the desired formula needs to be entered. But before we can create the formula, we first need to create "links" to the inputs of that formula.

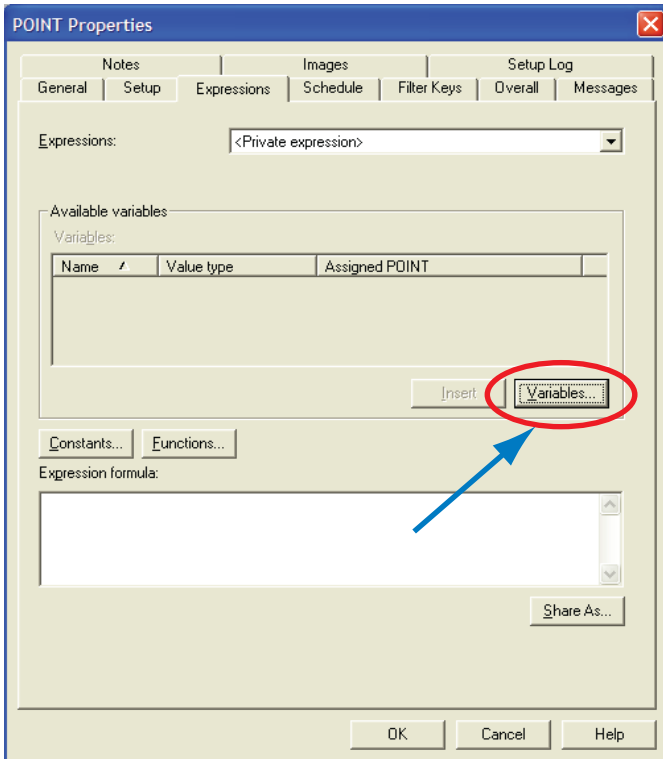


Figure 2. The Expressions tab. Before entering an expression you must create the Variables (indicated by the arrow).

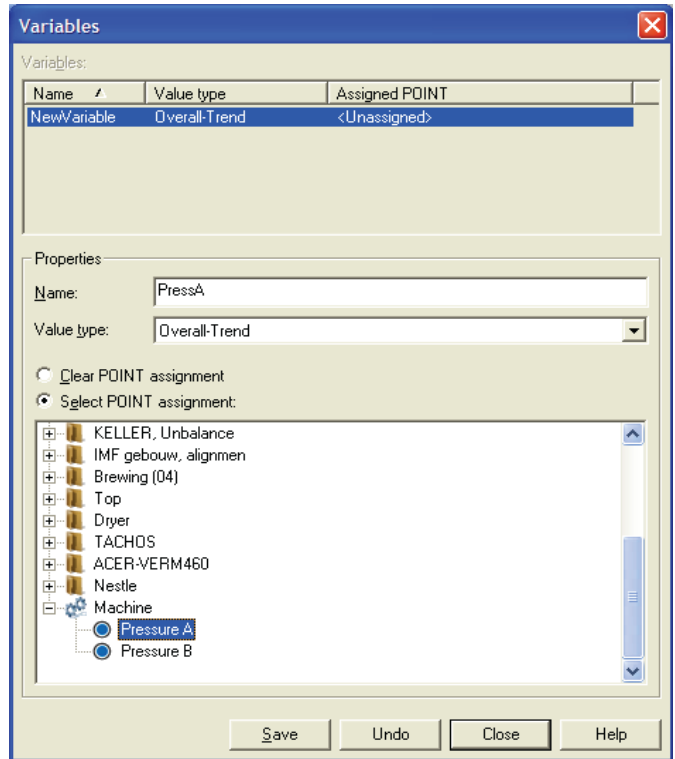


Figure 3. Creating a new variable association.

The Variables dialog allows you to create all the variables that need to be part of your expression. Follow this procedure to create the variables, variable types, and associations:

- 1 Press the Variables ... button in the POINT setup dialog. The "Variables" dialog opens.
- 2 In the Variables dialog, press the Add button, a new variable with the name NewVariable appears.
- 3 In the Name text box, replace the NewVariable string with a more descriptive name. We are going to associate the first variable with our "Pressure A" POINT, hence, we will name it "PressA."
- 4 Next, you will need to select the value type of PressA. This determines whether you make an association to its overall value, time waveform, spectrum, etc. Since PressA is just a simple overall value, you should select Overall-Trend. Note that most expressions will use this value type.
- 5 After the value type is selected, press the "Select POINT assignment" option, this enables the dialog's hierarchy area, in which the machine node is opened for the derived POINT you are creating. In our example, the hierarchy shows both pressure POINTs. Highlight the Pressure A POINT.
- 6 Press the Save button. You have now created the first variable association you need for the pressure difference formula. Follow the steps above to create the second variable associated with the Pressure B POINT.
- 7 Now that we have our two variables defined, all we need to do is write the pressure difference expression. The equation used here is very simple; PressA – PressB. To do this, place the mouse cursor in the Expression formula: text box where you want the variable to be inserted, then click the insert button.

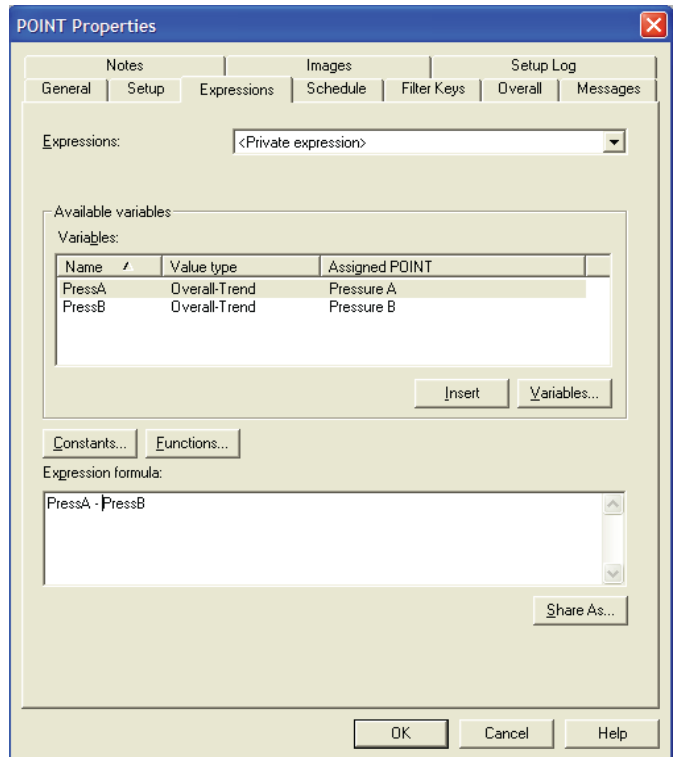


Figure 4. The example derived POINT with pressure difference expression.

- 8 Locate the new derived POINT in the hierarchy, highlight it and select Edit/ Calculate Derived POINTs. This calculates values from all stored data (data stored for the assigned “variable” POINTs) for the new derived POINT. If you created the Pressure A and Pressure B POINTs with the values shown in **Table 1**, your trend will look like the one in **Figure 5**.
- 9 There are a few things to refine. We want to set the alert at 1 PSI and the danger at 2 PSI. However, notice that the third measurement in our trend is negative. This is because the variable values are 10 and 12 for that date, which results in -2. This means we need to use a window alarm with both high as well as low alert and danger levels. This is certainly possible but for this example we want positive values only. In other words, we don’t want to know if the expression goes negative, we just want to know the difference. This can be solved by using a derived POINT function; the Abs() function (refer to section derived POINT functions below). Also notice that the Y-axis units is set to “Any Units,” we want to set this to “delta PSI.”

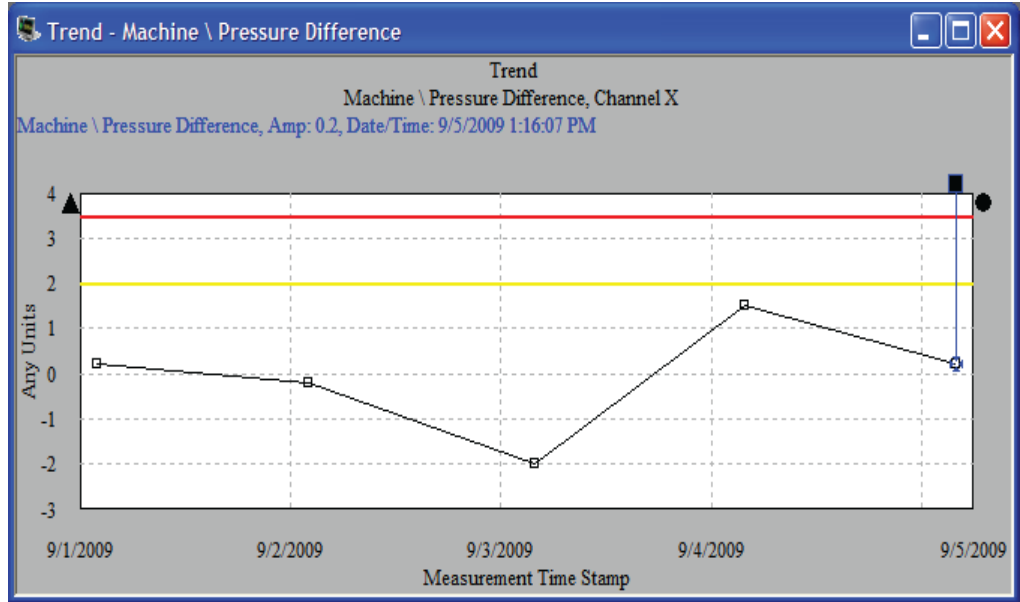


Figure 5. Pressure difference trend.

- 10 So let’s make these changes. Highlight the derived POINT, right-click and select Properties. Select the Setup tab. Change Full scale units to “delta PSI” (refer to **Figure 6**).
- 11 Select the Expression tab and edit the expression. The Abs() function can be inserted by clicking the Functions ... button, selecting the Abs function, and then OK button. Modify it per **Figure 7**.

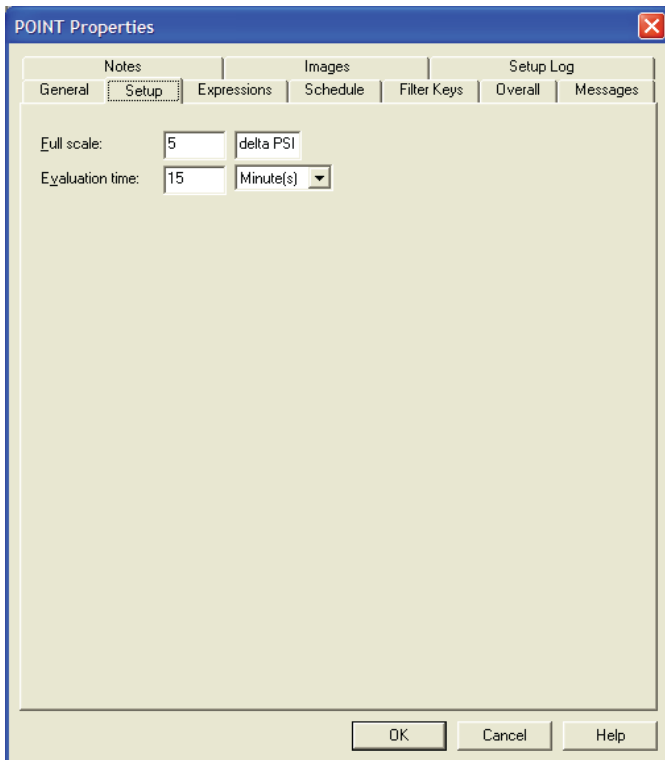


Figure 6. Changes to the derived POINT unit.

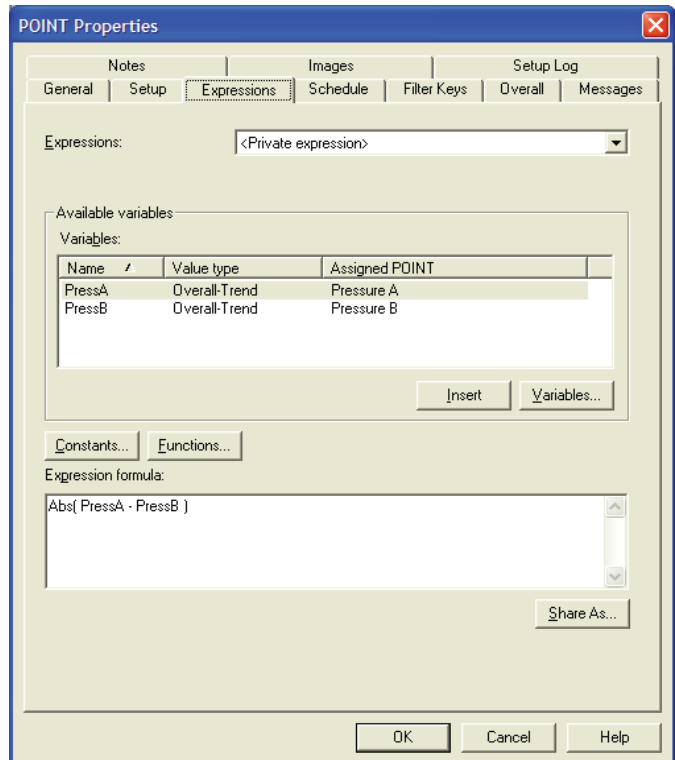


Figure 7. The Expression with Abs() function applied.

- Lastly, select the Overall tab and change the Alert and Danger settings to 1 and 2 PSI respectively.

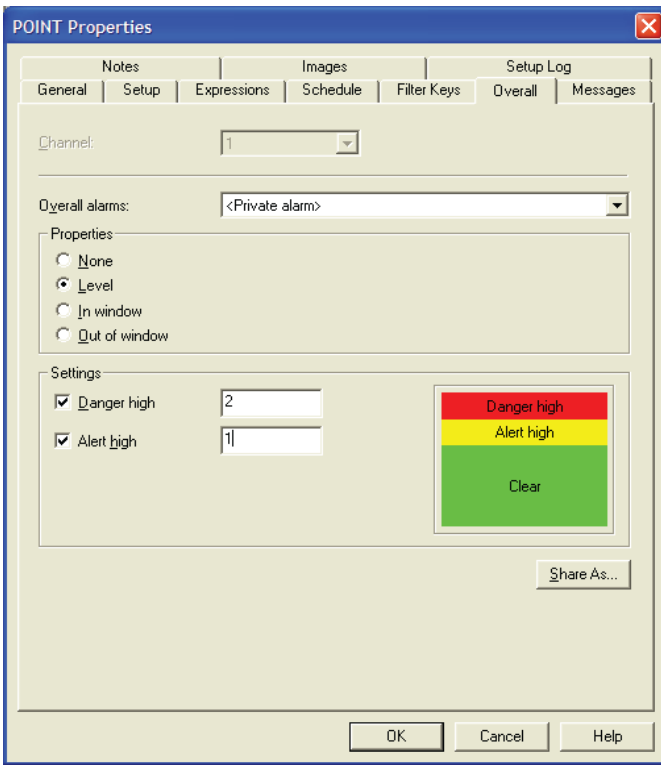


Figure 8. Setting the alarm values for the derived POINT.

- Save the changes by pressing the OK button. Highlight the derived POINT and select Edit/Calculate Derived POINTs to force the software to update the POINT's values for the new formula. The resulting trend is shown in Figure 9.

## SKF Microlog Inspector/SKF MARLIN Derived POINTs

Derived POINTs are normally computed by the SKF @ptitude Analyst application, however, when connecting to a SKF Microlog Inspector/SKF MARLIN device it is possible to set up the derived POINT so it computes as data is collected on the SKF Microlog Inspector/SKF MARLIN. This type of derived POINT must be specified during setup, and is downloaded to the SKF Microlog Inspector/SKF MARLIN. After data collection, the SKF Microlog Inspector/SKF MARLIN computed derived POINT data is uploaded to SKF @ptitude Analyst with all the other collected data, and is handled by SKF @ptitude Analyst like any other "normal" POINT data. The advantage of having a derived POINT compute in the SKF Microlog Inspector/SKF MARLIN is that the derived POINT values can be viewed directly by the operator in the field. For example, if the aforementioned pressure difference POINT was computed on the SKF Microlog Inspector/SKF MARLIN, and exceeded alarm settings, the operator would be immediately notified and have an opportunity to make corrections in the field.

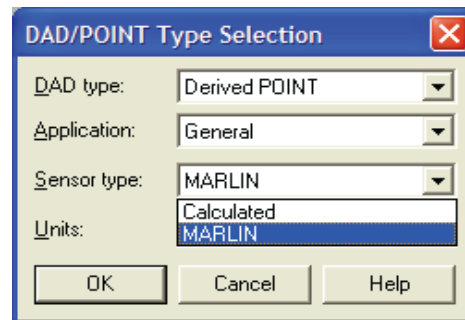


Figure 10. Selecting a derived POINT that gets computed by the SKF Microlog Inspector/SKF MARLIN data collector.

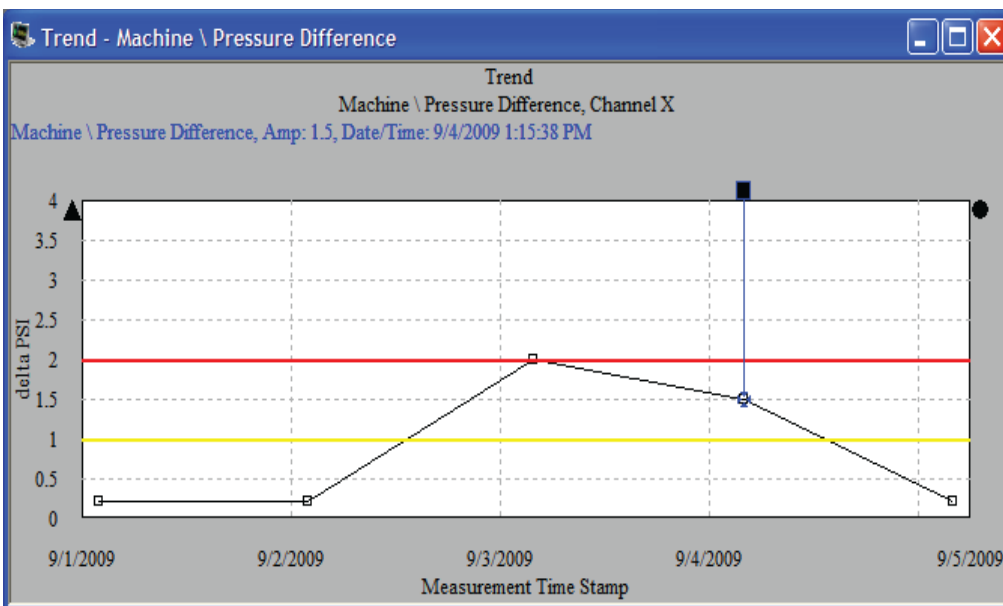


Figure 9. Final pressure difference trend.

## Private and Shared Expressions

SKF @ptitude Analyst allows expressions to be stored privately per POINT or shared across the database. Shared expressions are an easy way to create a single expression that is subsequently used in a variety of POINTs. This helps cut down setup time, and also helps minimize setup mistakes. In addition, changing a shared expression updates all the POINTs that use them. After changing a shared expression, be sure to update the values with Edit/Calculate Derived POINTs.

A list of standard shared expressions can be found under Customize/POINT Attributes/Derived POINTs.

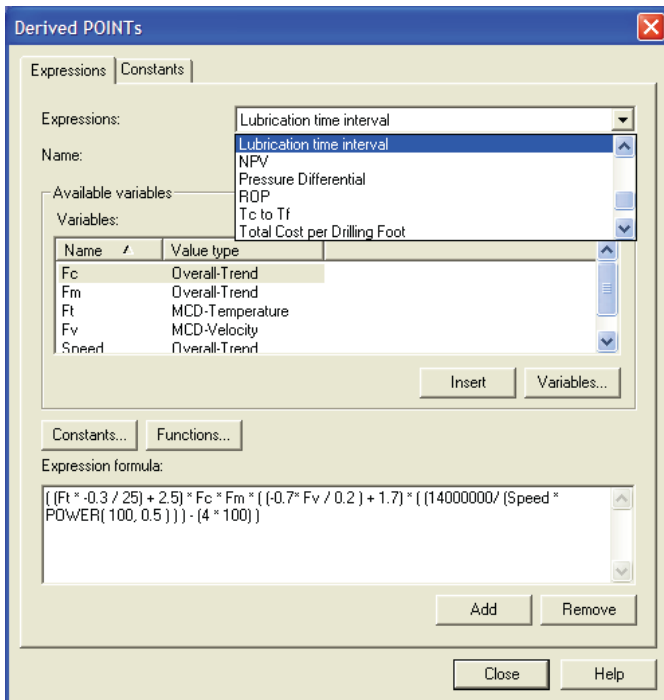


Figure 11. The list of shared expressions. From this dialog, expressions can be updated, added, removed.

To apply a shared expression, create a derived POINT and select the Expression tab. At the top of the dialog is a selection box that allows you to select either “private expression” or any one of the available shared expressions. In the **Figure 12**, the “Lubrication time interval” expression is selected. Notice that the expression formula is grayed out as is standard with shared values in SKF @ptitude Analyst. Also note that the variables needed for the expression formula are created but have not been assigned to their appropriate POINTs yet. You will need to do this through the Variables ... dialog.

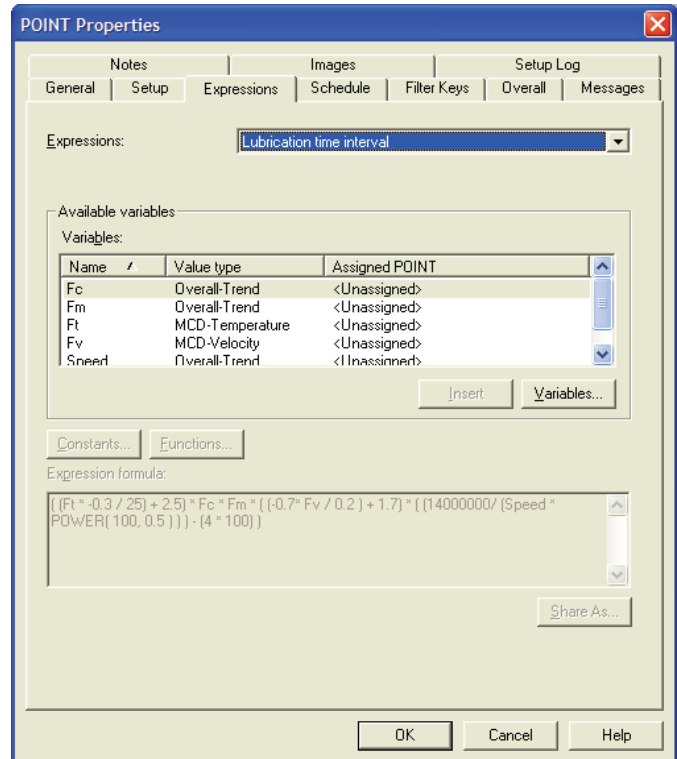


Figure 12. Applying a shared expression.

## Constants

You will notice that many expressions use constant values. Common examples are 3.1415 (Pi), gravity 9.8 m/s<sup>2</sup>, etc. SKF @ptitude Analyst enables you to either use the defined constants, or create new ones. The Customize/POINT Attributes/Derived POINT ... dialog shows all defined constants (Select the Constants tab).

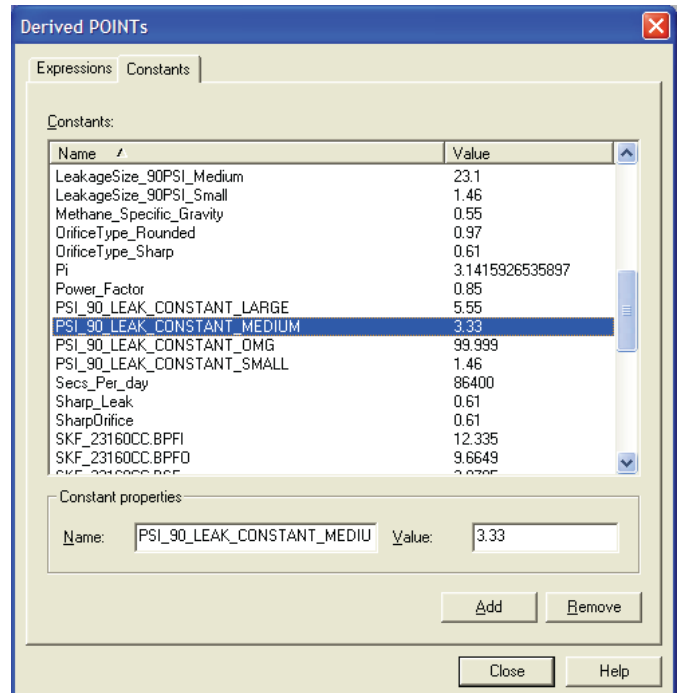


Figure 13. Predefined expression constants.

Applying a constant in an expression formula is very easy. The following expression:  $\text{Sine}(\text{Casing1Vib} / 3.1415)$  can be rewritten as  $\text{Sine}(\text{Casing1Vib} / \text{Pi})$ .

POINTS, etc. in the template. Once you apply the template, you will need to re-establish the variable associations in the derived POINTS in accordance with the expression for each POINT.

## Machine Templates

Derived POINTS can be part of a machine template also. Templates facilitate automation in creating hierarchies and therefore are a good way to quickly create a group of similar machines. Creating a template with one or more derived POINTS follows the normal procedure. First create your machine normally as part of the hierarchy. Include any derived POINTS you need, including their expressions. Then create a template and drag and drop the machine,

## Derived POINT functions

The derived POINT has a set of functions you may use as part of your formula. Just like Excel, a simple formula may consist of only the four prime arithmetic operators; add, subtract, multiply and divide. However, for many real world applications that isn't enough. For example, you may wish to calculate the maximum of two input values. This can be done with the  $\text{Max}()$  function. Therefore, over 40 derived POINT functions are available to perform specialized calculations that allow you to tailor your results.

**Table 2. Derived POINT functions.**

Function	Description	Platform*
Log(x)	Calculates the logarithm of x.	All
LN(x)	Calculates the natural logarithm of x.	All
Sin(x)	Calculates the sine of x. X must be specified in degrees.	All
ArcSin(x)	Calculates the arc sine of x. The value returned is in degrees.	All
Cos(x)	Calculates the cosine of x. X must be specified in radians.	All
ArcCos(x)	Calculates the arc cosine of x. The value returned is in degrees.	All
Tan(x)	Calculates the tangent of x. X must be in degrees.	All
ArcTan(x)	Calculates the arc tangent of x. The value returned is in degrees.	All
PercentChange(?)	Calculates the percentage change between two adjacent measurements. What is computed is: $(\text{Value} - \text{PreviousValue}) / \text{PreviousValue}$ .	All
Power(a, b)	Raises a to the power of b.	All
DeltaTime(x)	Calculates the difference in seconds between two adjacent measurements.	All
DeltaValue(x)	Calculates the difference in value between two adjacent measurements.	All
Deg2Rad(x)	Converts angle values in degrees to radians.	All
Abs(x)	Calculates the absolute value of x.	All
Operating Time	This function is available only by selecting 'Operating Time' for Application Type when creating the POINT. It computes the amount of time an asset has been operative in the time unit specified in the create POINT dialog.	All
Total Operating Time	This function is available only by selecting 'Total Operating Time' for Application Type when creating the POINT. It computes the total amount of time an asset has been operative in the time unit specified in the create POINT dialog.	All
Ceil(x)	Calculates the smallest (closest to negative infinity) integer that is not smaller than the original number.	All
Floor(x)	Calculates the largest (closest to positive infinity) integer that is not greater than the original number.	CMSW 7400 only
Round(x)	Calculates the integer closest to the original number.	CMSW 7400 only
Trunc(x)	Removes any fractional part from x.	CMSW 7400 only
Sqrt(x)	Calculates the square root of x.	All
XMag(Twf, cA, cB, spdFactor)	Computes the cross magnitude value (as opposed to cross phase) between two time waveform signals. The waveform signals are referenced by Twf, the waveform variable and cA or cB, the channel indicators. The specific frequency the cross magnitude value is computed for is determined by the speed value associated with the measurement and the provided spdFactor: $f = 1 / (s * \text{spdFactor})$ .  This function operates on multi-channel measurements only.	CMSW 7400 only

\* SKF @ptitude Analyst 2012 Edition or higher



**Table 2. Derived POINT functions.**

Function	Description	Platform*
XPhase(Twf, cA, cB, spdFactor)	Computes the cross phase value (as opposed to cross magnitude) between two time waveform signals. The waveform signals are referenced by Twf, the waveform variable and cA or cB, the channel indicators. The specific frequency the cross phase value is computed for is determined by the speed value associated with the measurement and the provided spdFactor: $f = 1 / (s * \text{spdFactor})$ .  This function operates on multi-channel measurements only.	CMSW7400 only
HAL(iSpectrum, iFrequency)	Calculates the Harmonic Activity Index for the frequency specified by iFrequency in iSpectrum. iFrequency must be in Hertz. The returned value is a dimensionless indicator of the likelihood that a harmonic pattern at iFrequency exists.	CMSW 7400 only
CrestFactor(iDynamic)	Calculates the crest factor. The crest factor is defined as the time waveform's Peak value divided by its RMS value. iDynamic is the time waveform. The resulting value is dimensionless.	CMSW 7400 only
CTA_PK_PK(iDynamic, iPeriod)	Calculates the Cyclic Time Average of a time waveform using averaging period iPeriod. iDynamic is the time waveform. iPeriod must be specified in seconds. The returned value is in the same engineering units as iDynamic and Peak to Peak detection type.	CMSW 7400 only
CTA_RMS(iDynamic, iPeriod)	Calculates the Cyclic Time Average of a time waveform using averaging period iPeriod. iDynamic is the time waveform. iPeriod must be specified in seconds. The returned value is in the same engineering units as iDynamic and RMS detection type.	CMSW 7400 only
Contribution2(iDynamic, iPeriod)	Calculates the contribution (in percentage) of the energy periodical to iPeriod. iDynamic is the time waveform. iPeriod is specified in multiples of running speed including fractional values.	CMSW 7400 only
Contribution3(iDynamic, iPeriod, iSpeed)	Calculates the contribution (in percentage) of the energy periodical to iPeriod. iDynamic is the time waveform. iPeriod is specified in multiples of running speed including fractional values. iSpeed is the machine speed value to use. iSpeed may be a fixed value, an overall value from a trend, or a speed value extracted with Speed() from a time waveform or spectrum.	CMSW 7400 only
Speed(iDynamic)	Extracts the machine speed from iDynamic. iDynamic must be either a time waveform or spectrum. The returned value is in Hertz.	CMSW 7400 only
Min(x, y)	Calculates the minimum value between x and y. Returns either x or y.	CMSW 7400 only
Max(x, y)	Calculates the maximum value between x and y. Returns either x or y.	CMSW 7400 only
Smax(iTwf)	Calculates the maximum vector magnitude of an Orbit. iTwf must refer to a 2 channel time waveform value.	CMSW 7400 only
PeakValue(iFFT, iBandIndex)	Calculates the peak value found in the frequency band with index iBandIndex. iBandIndex starts at 1. Returns zero when no frequency band exists for the supplied index number. Returns a peak value with the same units and detection type as iFFT.	CMSW7400 only
Energy_Value(iFFT, iBandIndex)	Calculates the overall energy found in the frequency band with index iBandIndex. iBandIndex starts at 1. Returns zero when no frequency band exists for the supplied index number. Returns a band overall energy value with the same units and detection type as iFFT.	CMSW 7400 only
Mod(x, y)	Calculates the remainder after the division x/y. E.g. Mod(7, 3) = 1.	CMSW7400 only
Average(x, n)	Calculates the average of x using a sliding window of n values.	All
Average(Dynamic, c)	Calculates the average value of a dynamic signal (FFT or waveform) for the specified channel 'c'. For non-multi-channel measurements, 'c' should be set to 1 (one).	All
Stdev(x, n)	Calculates the standard deviation of x using a sliding window of n values. There should be at least two values in x.	CMSW 7400 only
Stdev(Dynamic, c)	Calculates the standard deviation of a dynamic signal (FFT or waveform) for the specified channel 'c'. For non-multi-channel measurements, 'c' should be set to 1 (one).	CMSW 7400 only
ROC(x, iDays)	Calculates the rate of change (in days) between two adjacent measurements.	CMSW 7400 only
Alarm_Status(p)	Determines and stores the current alarm status for POINT p. Notice that this function does not re-evaluate past alarm conditions and therefore must be operated immediately when new alarm conditions are set for POINT p.  Returned values are:  -1 No alarm level (when POINT p has no values, no set alarm) 0 Safe 1 Alert 2 Danger	CMSW 7400 only
Skew(x, n)	Calculates the Skew for x using a sliding window of n values. There should be at least 4 values in x.	CMSW 7400 only
Skew(Dynamic, c)	Calculates the Skew of a dynamic signal (FFT or waveform) for the specified channel 'c'. For non-multi-channel measurements, 'c' should be set to 1 (one).	CMSW 7400 only

\* SKF @ptitude Analyst 2012 Edition or higher



**Table 2. Derived POINT functions.**

Function	Description	Platform*
Kurtosis(x, n)	Calculates the Kurtosis of excess for x using a sliding window of n values. There should be at least 3 values in x.	CMSW 7400 only
Kurtosis(Dynamic, c)	Calculates the standard deviation of a dynamic signal (FFT or waveform) for the specified channel 'c'. For non-multi-channel measurements, 'c' should be set to 1 (one).	CMSW 7400 only
CUSUM(x, m)	Calculates the accumulative difference of x against the model parameter m.	CMSW 7400 only
Forecast_Exp(x, n, Alarm)	Calculates a forecasted alarm in days of x using a sliding window of n values. Uses an exponential fitting function.	CMSW 7400 only
Forecast_Lin	Calculates a forecasted alarm in days of x using a sliding window of n values. Uses a linear fitting function.	CMSW 7400 only
Count_Spikes(Twf, c, Th, Bias)	Counts the number of spikes in the waveform indicated by 'Twf' for the specified channel 'c'. A spike is defined as a signal level equal or higher than the specified threshold value 'Th'. The specified Bias value is subtracted from the signal prior to evaluation against the threshold value.  Use the Average() function to de-trend signals with strong DC bias i.e.:  Count_Spikes( twf, c, th, Average( twf, c ) )  For non-multi-channel measurements, 'c' should be set to 1.	CMSW 7400 only

\* SKF @ptitude Analyst 2012 Edition or higher

## Examples

The following examples assume that you have read and understand the previous section “Creating a Derived POINT.” For brevity, only the discussion and formulas are shown, not the actual POINT setup details. In many examples, we refer back to the Pressure difference example discussed above.

### Example 1: Calculating a POINT’s average trend

Many parameter trends show significant variation, variation that may hinder a clear understanding of what is actually happening. The variation may be due to noise, process, or other machine related sources. A common method to gain a clearer picture is to generate the average value trend. This can be easily achieved with the Average() function, which implements a rolling average. This function requires two parameters: a) the associated variable and b) the averaging length. For example, if we wish to smooth out a trend variable with the name VarA, the expression Average( VarA, 3) will

use three consecutive values to reproduce a new averaged value. Increasing the averaging window from 3 to 5 creates a smoother, less bumpy trend. The maximum number of averages supported is 99. Choosing the proper averaging window size depends on the behavior of the parameter you are averaging. Experiment with this feature to achieve the best value for your purposes.

One particularly interesting application of a rolling average is to determine a break-away trend. This is a phenomenon whereby the actual collected overall value is rapidly becoming much larger than the trend of that overall value. To compute a breakaway trend for a variable with the name VarA, the expression is:

- $\text{VarA} - \text{Average}(\text{VarA}, 3)$ . If the collected value is more or less stable, the averaged value should be close to the collected value, and therefore the break-away trend is more or less a flat line hovering around zero.

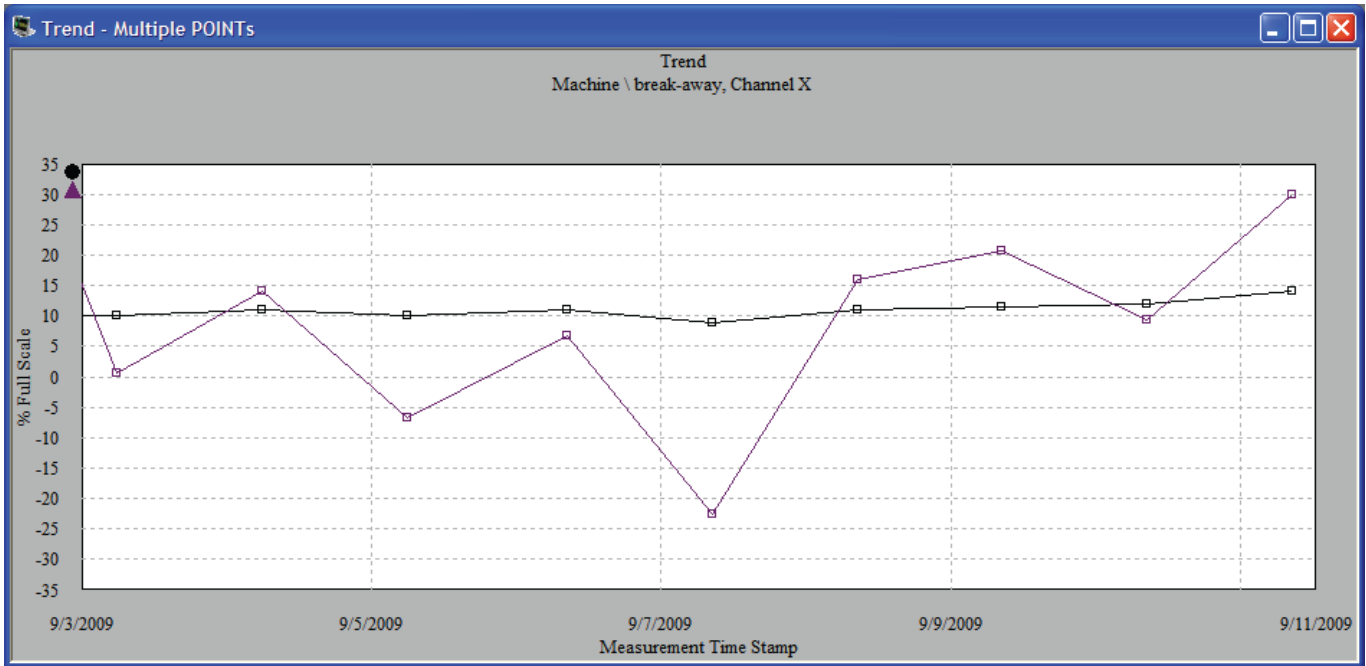


Figure 14. Example trend with breakaway trend (purple). Notice the break-away trend showing a strong positive development in the last measurement indicating that the collected overall value is increasingly higher than the average.

### Example 2: Calculating an asset's average vibration

Another typical use of derived POINTs is to compute the average vibration of an asset (e.g., machine, motor, dryer roll, gearbox, etc.). In particular, if several identical assets are monitored, the average vibration trend may better indicate a problem with a particular asset. It is common practice that assets are monitored with a variety of POINT types, such as acceleration, velocity, acceleration enveloping, etc. For the averaging function to work well, you should only average identical POINT types. In addition, identical POINT types may have a very big difference in bandwidth (i.e., end frequency), thereby have a big difference in captured energy – these should be treated separately as well.

For example, consider a paper machine dryer section whereby each cylinder is monitored with a single acceleration POINT, two different velocity POINTs (one with end frequency at 100 Hz, and one with an end frequency at 500 Hz), and two acceleration enveloping POINTs (one using env. filter 2, and using env. filter 3). If you wish to create a derived POINT that trends the average vibration of the entire dryer section, you should pick only one of these POINT types. Don't include in one average expression both acceleration and velocity POINTs. Also, don't create an average expression with both velocity POINTs, as their energy content is so different. If desired, simply create average trends for the various POINT types you like to trend.

If the dryer section has 5 cylinders numbered 1 through 5, with each acceleration POINT labelled as Cyl\_X\_Acc with "X" being the cylinder number, the following expression creates the average trend for the acceleration POINT values:

$$(Cyl\_1\_Acc + Cyl\_2\_Acc + Cyl\_3\_Acc + Cyl\_4\_Acc + Cyl\_5\_Acc) / 5$$

### Example 3: Calculating the likelihood of machine unbalance

Commonly, unbalance shows as a high 1x machine speed spectral line that is dominating the spectrum. Assuming a velocity POINT, Casing1Vib, measured in IPS pk, we set up a spectral band around the 1x running speed. Let's also assume that this is the first spectral band associated with this POINT. The peak value at running speed versus total energy in the spectrum gives us a measure of how "predominant" the 1x running speed is. This can be calculated by:

$$\text{Peak\_Value}( \text{Casing1Vib\_FFT}, 1) / \text{Casing1Vib}$$

This expression will trend higher (towards 1.0) with a more predominant 1x machine speed present.

### Example 4: Calculating simple compressor efficiency

Compressor relative efficiency can be expressed as:

$$\theta_{\text{Compressor}} = \frac{\frac{P_2}{P_1} - 1}{T_2 - T_1}$$

With "P" and "T" being pressure and temperature respectively, and position "1" at the suction and "2" at the discharge locations. (Refer to "A working guide to process equipment, Edition 1, Page 391").

We want to set up a derived POINT where the operator in the field can not only collect the temperature and pressure values but instantly see the result of the calculation. This allows the operator in the field to immediately make adjustments when necessary and possible. To create such a POINT, select DAD type = Derived POINT and Sensor Type = MARLIN.

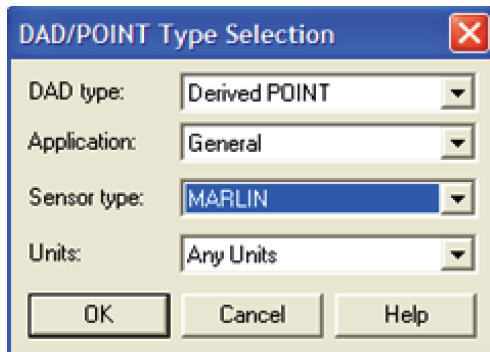


Figure 15. Selecting a derived POINT with the SKF MARLIN data collector.

Set up the four input parameters by associating them with the measurements and enter the expression per the formula below. The result is shown in Figure 16.

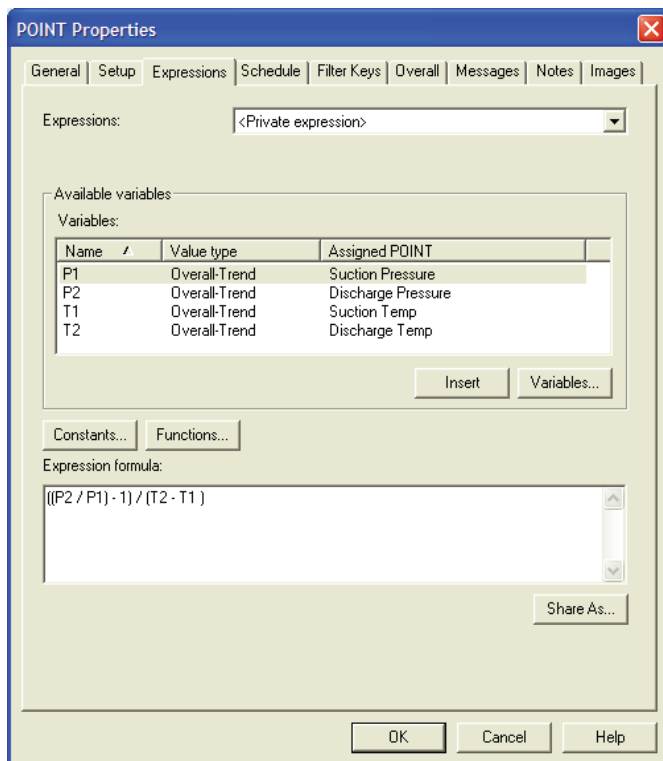


Figure 16. Expression for computing relative compressor efficiency.

### Example 5: Separating Synchronous from non-synchronous energy

Many rotating machines have spectra showing machine speed and its harmonics, and other synchronous frequencies such as blade pass, gear mesh, etc. The spectra may also exhibit non-synchronous signals from other sources, such as cavitations, looseness, etc. The change in the balance of the synchronous versus non-synchronous energy may indicate physical or operational change that is worth investigating. Through a derived POINT, this ratio can be easily trended.

Create a frequency band for all synchronous frequencies you consider having an important energy signature. Since frequency bands have alarm limits, you may wish to tailor these to your specific needs as well. However, if all you are interested in is the synchronous versus non-synchronous energy ratio, just set the alarm limits sufficiently high so they will not be triggered and therefore not show up on an alarm report. For this example, let's assume that you have created three bands around the fundamental, first, and second harmonic for POINT Brg1Vib. The formula to trend the synchronous versus non-synchronous energy ratio is:

$$\frac{(\text{Energy\_Value}(\text{Brg1Vib\_FFT}, 1) + \text{Energy\_Value}(\text{Brg1Vib\_FFT}, 2) + \text{Energy\_Value}(\text{Brg1Vib\_FFT}, 3))}{\text{Brg1Vib}}$$

Whereby Brg1Vib\_FFT is associated with the Brg1Vib's spectrum, the numbers 1 through 3 refer to spectral bands 1 through 3, and Brg1Vib is the overall value of the measurement.

As you see, the numerator adds up all the synchronous energy in the bands that is then divided by the total energy, which is simply the overall value for that POINT. As an example, a gas turbine will have most of the energy measured centered around machine speed and its harmonics, so the trend for this derived POINT should be perhaps between 0.8 and 1.0.

### Example 6: Calculating an Oil whip or Oil whirl trend.

Oil whip and oil whirl are phenomena that may occur in journal bearings. They appear as a frequency that is typically between 40% to 50% of machine speed. Oil whirl is considered severe when amplitudes exceed 40% of the normal bearing clearance. Refer to Mitchell, John S. "An Introduction to Machinery Analysis and Monitoring" for more information on oil whip/whirl.

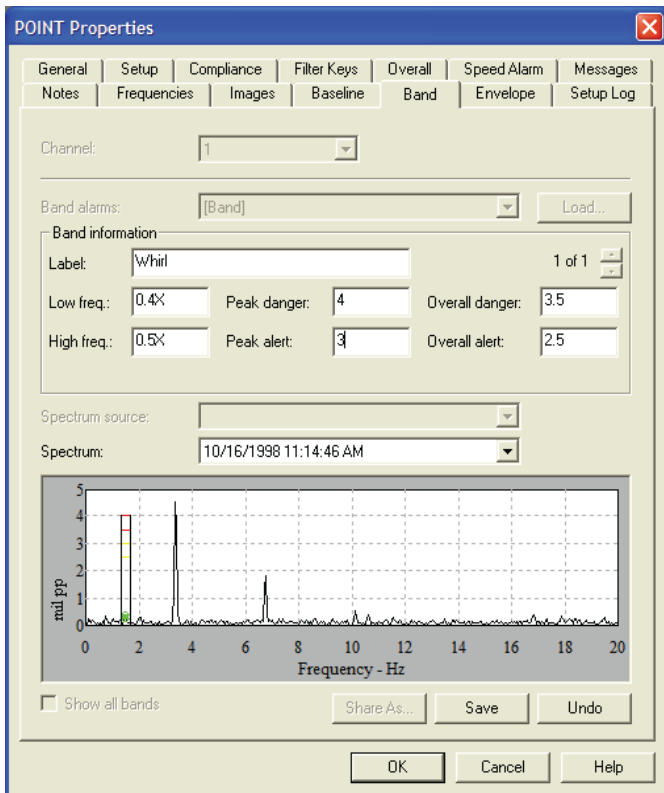


Figure 17. Setting up a oil whirl band.

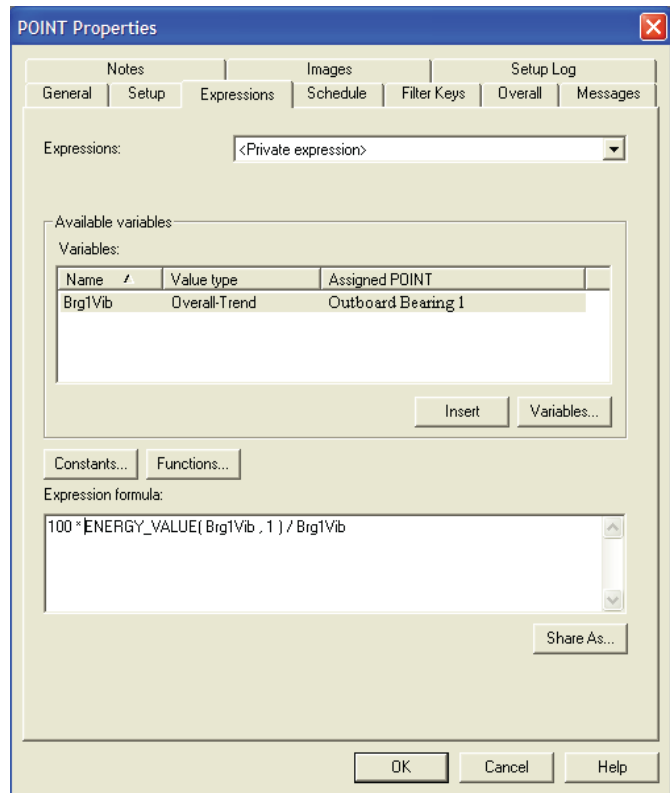


Figure 18. Expression for percentage of energy tracking oil whirl.

The derived POINT function Energy\_Value() can easily trend the oil whip or oil whirl sub-synchronous energy. To be able to track the sub-synchronous energy, define a band that runs from 0.4x to 0.5x running speed. **Figure 17** shows the Band setup. Since this POINT has no other bands than the “whirl” band, the band index is “1.” Though we can easily just track the energy with Energy\_Value

(Brg1Vib, 1), in this case we want to know the ratio of the whirl energy versus all energy in percentage. **Figure 18** displays this expression:

$$100 * \text{Energy\_Value}( \text{Brg1Vib\_FFT}, 1 ) / \text{Brg1Vib}$$

Whereby Brg1Vib\_FFT is associated with the Brg1Vib’s spectrum, and Brg1Vib is the overall value of the measurement.

Please contact:

**SKF USA Inc.**

**Condition Monitoring Center – San Diego**

5271 Viewridge Court • San Diego, California 92123 USA

Tel: +1 858-496-3400 • Fax: +1 858-496-3531

**Web: [www.skf.com/cm](http://www.skf.com/cm)**

© SKF, Microlog, MARLIN, and @ptitude are registered trademarks of the SKF Group.

Microsoft and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners.

© SKF Group 2012

The contents of this publication are the copyright of the publisher and may not be reproduced (even extracts) unless prior written permission is granted. Every care has been taken to ensure the accuracy of the information contained in this publication but no liability can be accepted for any loss or damage whether direct, indirect or consequential arising out of the use of the information contained herein. SKF reserves the right to alter any part of this publication without prior notice.

