AE040 - Siemens macro L8503 - set micron compensation value

Function

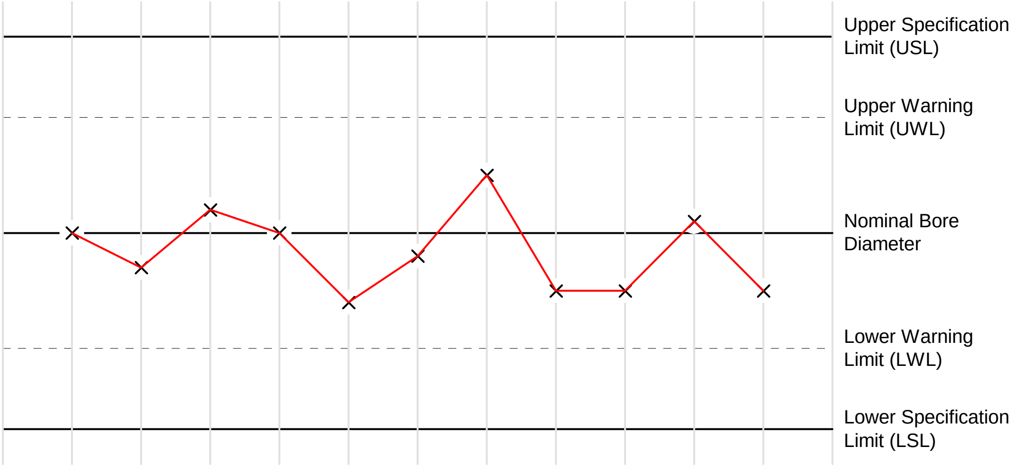
To analyse a series of bore measurements and decide when a tool compensation is required to maintain a bore diameter within set limits. Then compute a micron compensation value to achieve this.

Use:

L8503 (BORE_ID,TOOL_ID,BORE_MEAS,BORE_NOM,BORE_LSL,BORE_USL,BORE_LWL, BORE_UWL,LWL_MAX,UWL_MAX)

Structure:

```
L8503 (     INT Bore ID,
            INT Tool ID,
            REAL Bore measurement,
            REAL Nominal bore size,
            REAL Bore lower specification limit (LSL),
            REAL Bore upper specification limit (USL),
            REAL Bore lower warning limit (LWL),
            REAL Bore upper warning limit (UWL),
            INT Maximum consecutive LWL transgressions,
            INT Maximum consecutive UWL transgressions )
```



Functional Description

The macro uses the current bore diameter measurement to assess the bore condition relative to the inspection parameters, for an unlimited number of bores.

The macro retains the outcomes of previous analyses' for each bore, to identify when a tool compensation is required, and the size of compensation needed to ensure that the tool produces a bore of nominal size.

The bores are identified by the numbers 0, 1, 2, 3 etc. The macro sets a bore condition variable and if necessary a compensation value for each bore type using the default structure shown in Table 1. The bore condition variable can be used in conditional statements outside the macro. Several coding examples are given at the end of this document that show how this might work in practice.

The default block of 200 RAE variables used by Active Edge is RAE[0] to RAE[199]. RAE[0] to RAE[50] are used by standard AE tool compensation macros. RAE[160] to RAE[199] inclusive are used to record persistent measurement outcomes for up to 8 bores within L8503. Note that the variable range is offset in accordance with whichever block of 200 variables is reserved for ActiveEdge use on any given installation. The record sheet at the end of this document can be used to clarify which variables are being used.

If analysis of more than 8 bores in required, the machine control must have five variables mapped above RAE[199] for each additional bore. (e.g. to track 10 bore diameters, the RAE variables must extend to RAE[199] + 5 + 5 = RAE[209]).

Table 1 – Bore Variables

Bore ID	Tool ID	Micron Comp. on dia.	Bore condition
0	RAE[160]	RAE[161]	RAE[162]
1	RAE[165]	RAE[166]	RAE[167]
2	RAE[170]	RAE[171]	RAE[172]
-	-	-	-
n	RAE[160+(5*n)]	RAE[161+(5*n)]	RAE[162+(5*n)]

For each bore ID:
If a bore measurement transgresses either the UWL or LWL (without exceeding the bore specification limits), internal counter variables specific to either the upper or lower transgression limits are incremented. These variables only increment on consecutive measurement transgressions, otherwise they are set to zero.

If either counter reaches the maximum upper or lower transgression limit (UWL_MAX or LWL_MAX), a micron compensation value on diameter is computed and written to the appropriate variable associated with that bore. (Table 1). The compensation value is an integer computed as the nominal bore diameter minus the last bore measurement.

The micron compensation variable is always set to zero, unless a compensation event is triggered.

If a bore measurement is outside the specification limits, the macro clears all trend data and immediately sets the bore condition variable and a corrective compensation value based on the current measurement. It is the programmer's responsibility to monitor the bore condition variable after L8503 executes and decide what action to take. For example, with reference to Table 2, a sudden bore condition of 3 (i.e. below the

BORE_LSL) may indicate a broken insert, in which case the programmer may wish to swap to a sister tool rather than allow the tool to try and compensate automatically.

When a micron compensation value is generated, a copy of the tool ID is also written to the appropriate variable associated with the bore (Table 1). The copy tool ID does not have to be used, but it can often aid programming flexibility if the 'check comp. status' macro L8506 is not being called immediately after a compensation request. In this case the copy tool ID is available for use much later in the program to query the outcome of the earlier compensation request.

LWL_MAX and UWL_MAX must be set to 1 or more.

If BORE_MEAS exceeds either the LSL or the USL, the bore status variable is set to either 3 or 4 respectively, and the persistent counter variables are cleared.

The appropriate bore condition variable is set each time the macro is executed. It can be used by the manufacturing program in a variety of ways. For example, it should be used to determine whether the bore/part is 'scrap' or 're-workable scrap'.

Table 2 – Bore condition classifications

Bore Condition	Measurement	Meaning
0	Between the LWL and UWL	In spec., within warning limits
1	Between the LWL and the LSL	In spec., outside warning limits
2	Between the UWL and the USL	In spec., outside warning limits
3	Below the LSL	Out of spec., dia. is undersize
4	Above the USL	Out of spec., dia. is oversize

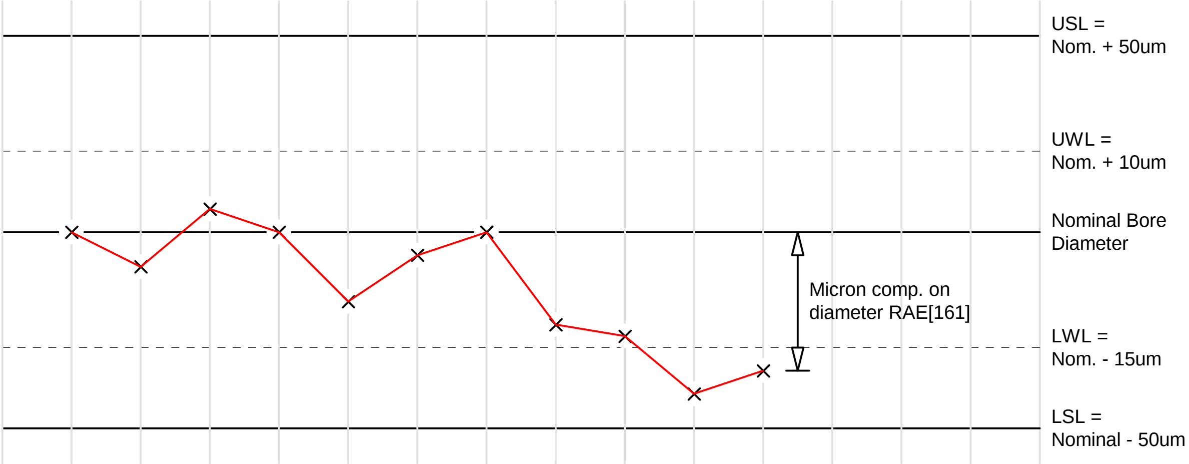
In summary, historic variables relating to a specific bore are automatically initialised (set to zero) in the following cases:

(1) Once a compensation value has been successfully computed.
(2) If a bore measurement falls outside the bore specification limits.
(3) If the inspection parameters are not logical. The macro also generates an s329 warning message in this event.
(4) Forced reset. A reset can be 'manually' forced as shown below. For example if program execution is interrupted and has to be restarted, L8503 can be run with the BORE_MEAS parameter set to -1 as shown below. The macro must be run explicitly for each bore ID, for example

```
L8503(0,x,-1,x,x,x,x,x,x,x)
L8503(1,x,-1,x,x,x,x,x,x,x)
L8503(2,x,-1,x,x,x,x,x,x,x)
.
.
.
```

Example 1 - Set compensation value

In this example, L8503 is set to compute and trigger a compensation after the second consecutive LWL transgression. The compensation amount (RAE[161] in this case) is derived from the last bore measurement, as shown below.



```
;R18 = Tool ID
;R70 = Measured bore size
;R96 = Nominal bore size

L8503 (0, R18, R70, R96, R96-0.05, R96+0.05, R96-0.015, R96+0.01, 2, 2)
```

Example 2

Typical usage of macro L8503. Cutting a single bore with a single AE tool

```
;R18 = ID of tool used to cut bore 0
;R70 = bore 0 measurement
;R96 = nominal bore size

N100 ;measure bore 0
N110 L8503 (0, R18, R70, R96, R96-0.05, R96+0.05, R96-0.015, R96+0.01, 2, 2)
N130 IF RAE[162]<3 ;bore condition ok/initiate comp
N140 L8504(R18, RAE[161]) ;comp tool by amount computed in L8503. (skips if
RAE[161]=0)
N150 GOTOF ENDCOMP
N160 ELSE
N170 IF RAE[162]==3
N180 ;mark measured part as re-workable scrap
N190 ;mark tool for insert damage/inspection then swap to sister tool
N200 ENDIF
N210 IF RAE[162]==4
N220 ;mark measured part as scrap
N230 ;investigate reason for oversize bore. Run probe calibration cycle?
N240 ENDIF
N250 ENDIF
N260 ENDCOMP:
```

Example 3

Program structure to comp a tool cutting two bores of the same diameter on a single part.

```
;R18 = ID of tool used to cut both type 0 bores
;R70 = first bore (type 0) measurement
;R71 = second bore (type 0) measurement
;R96 = nominal bore size of type 0 bore

N100 ;measure both type 0 bores
N110 L8503 (0, R18, R70, R96, R96-0.05, R96+0.05, R96-0.015, R96+0.01, 2, 2)
N120 L8503 (0, R18, R71, R96, R96-0.05, R96+0.05, R96-0.015, R96+0.01, 2, 2)

;the program would then continue from N130 in Example #2
```

In this example, the macro is run twice - once for each measurement. This is because the bores are the same size. The tool ID stays the same for each macro call because both bores were made by the same tool.

N.B. This method of monitoring identical bores on the same workpiece should only be used if the pre-bored hole size and material condition is the same for each bore. When machining high tolerance parts, even small differences in these metrics can cause size ‘hunting’, which can adversely affect process efficiency.

Example 4

It may be that in addition to the operations in the previous example, a different AE tool is also machining a single bore on the same part. If so, the program could be structured as shown.

```
;R18 = ID of tool used to cut type 0 bores
;R20 = ID of tool used to cut type 1 bore
;R70 = first bore (type 0) measurement
;R71 = second bore (type 0) measurement
;R72 = third bore (type 1) measurement
;R96 = nominal size of type 0 bores
;R97 = nominal size of type 1 bore

N100 ;measure both type 0 bores
N110 L8503 (0, R18, R70, R96, R96-0.05, R96+0.05, R96-0.015, R96+0.01, 2, 2)
N120 L8503 (0, R18, R71, R96, R96-0.05, R96+0.05, R96-0.015, R96+0.01, 2, 2)
N130 IF RAE[162]<3 ;bore condition ok/initiate comp
N140 L8504(R18, RAE[161]) ;comp 1st tool by amount computed in L8503. (skips if
RAE[161]=0)
N150 GOTOF NEXTCOMP
N160 ELSE
N170 IF RAE[162]==3
N180 ;mark measured part as re-workable scrap
N190 ;mark tool for insert damage/inspection then swap to sister tool
N200 ENDIF
N210 IF RAE[162]==4
N220 ;mark measured part as scrap
N230 ;investigate reason for oversize bore. Run probe calibration cycle?
N240 ENDIF
N250 ENDIF

N260 NEXTCOMP:
N270 ;measure type 1 bore
N280 L8503 (1, R20, R72, R97, R97-0.05, R97+0.05, R97-0.015, R97+0.01, 2, 2)
N290 IF RAE[167]<3 ;bore condition ok/initiate comp
N300 L8504(R20, RAE[166]) ;comp 2nd tool by amount computed in L8503. (skips if
RAE[166]=0)
N310 GOTOF ENDCOMP
N320 ELSE
N330 IF RAE[167]==3
N340 ;mark measured part as re-workable scrap
N350 ;mark tool for insert damage/inspection then swap to sister tool
N360 ENDIF
N370 IF RAE[167]==4
N380 ;mark measured part as scrap
N390 ;investigate reason for oversize bore. Run probe calibration cycle?
N400 ENDIF
N410 ENDIF
N420 ENDCOMP:
```

ActiveEdge

RIGIBORE

VARIABLE RECORD SHEET FOR PROGRAMMING WITH SPC MACRO L8503

n	RAE[160+(5*n)]	RAE[161+(5*n)]	RAE[162+(5*n)]							
Bore ID	Copy tool ID	Micron comp. on dia.	Bore condition							
0										