Chapter 21

STATISTICAL

Edition 4

Overview

Overview

This class allows Statistical Process Control functions on a single REAL value in real time. The outputs may be simply communicated to the operator via a suitable user interface such as a terminal, or if the process is sufficiently well defined and known, can be used to modify the process without operator intervention.

Statistical Process Control is used in applications where long term conditions must be monitored and controlled to optimise process yield.

Function Blocks may be used *independently* or in *tandem* with each other.

There are three full function SPC blocks:-

SPC provides a basic Shewhart control chart for an analogue or real variable sampled on the basis of TIME.

SPC_Event provides similar functionality to the SPC block, but sampling occurs on demand i.e. an EVENT.

The **Histogram** computes the distribution of the real variable and provides running average, standard deviation and provides an indication of capability index.

The final block, **Statistics**, simply provides min, max, mean and standard deviation of a REAL variable.

It is impossible to inspect or test quality into a product; the product must be built right the first time. This implies that the manufacturing processing must be stable and that all individuals involved with the process (including operators, engineers, quality-assurance personnel, and management) must continuously seek to improve process performance and reduce variability on the key parameters.

The first step to improvement is the ability to measure quality. It is therefore important to work out which process variable(s) is(are) important in assessing quality and moreover which variables can be manipulated to influence quality.

In order to assess the product quality there are a number of simple statistical tools that can be used: these include

Control Charts

Histogram

Check sheets

Pareto analysis

Defect concentration diagrams

Cause and effect diagrams

Scatter charts and correlation

Control charts and histograms are tools that can be used on-line and the PC3000 function blocks provide the capability of the routine book keeping associated with running a chart for variables (as opposed to attributes).

A typical control chart is as shown in figure 21-1.

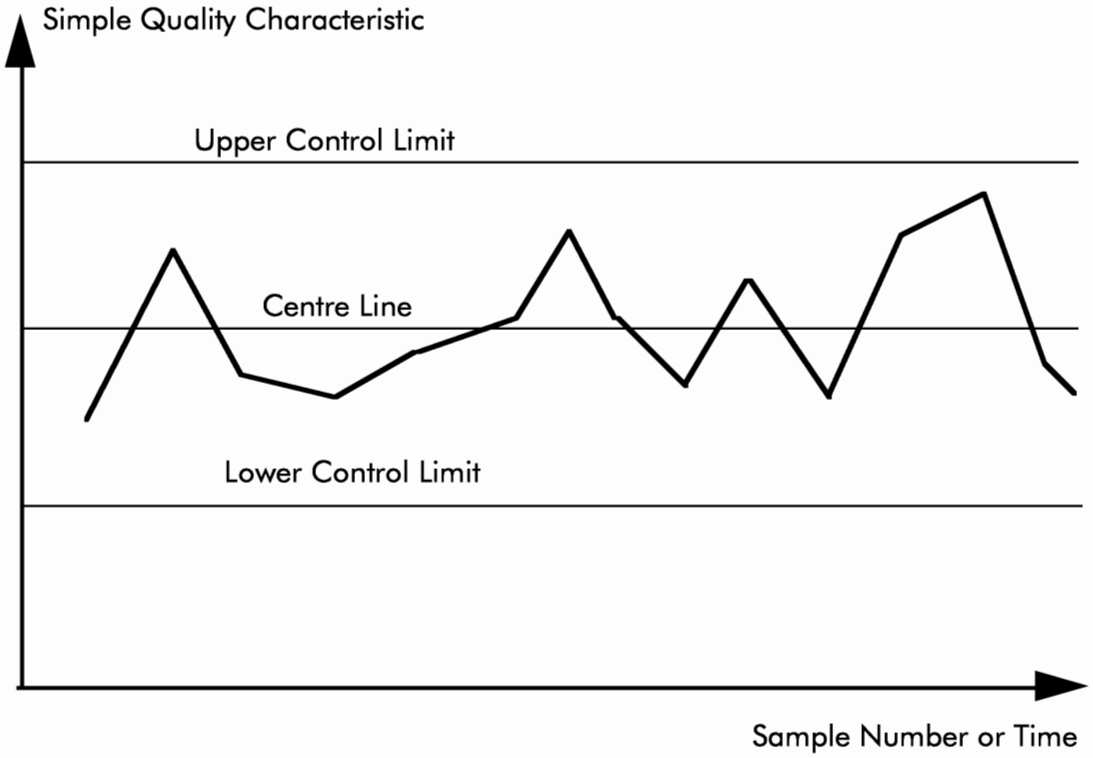

Figure 21-1 A Typical Control Chart

The typical method of producing the chart is to measure the signal of interest at a regular interval (if the sampling is to be performed with respect to time). The samples are then grouped together in small to medium size subgroups (in the case of PC3000 the subgroup size is between 2 and 10).

Some statistics of the subgroup are computed (in the case of PC3000 mean, standard deviation and range [range is defined as the difference between the largest and the smallest values in the subgroup]).

These are then plotted and studied according to some specified rules with regards to the book keeping of the chart. In PC3000 the book keeping is done automatically (e.g. tests for out of control samples, drifts, biases etc.).

The user has to define the grouping methodology:

Sub_Time defines the basic interval between subgroup samples,

Sub_Size defines the total number of elements of a subgroup,

Group_Time defines the time difference between subgroup samples and

Group_Size fixes the total number of subgroups used for assessment (the maximum is set to 125). See figure 21-2.

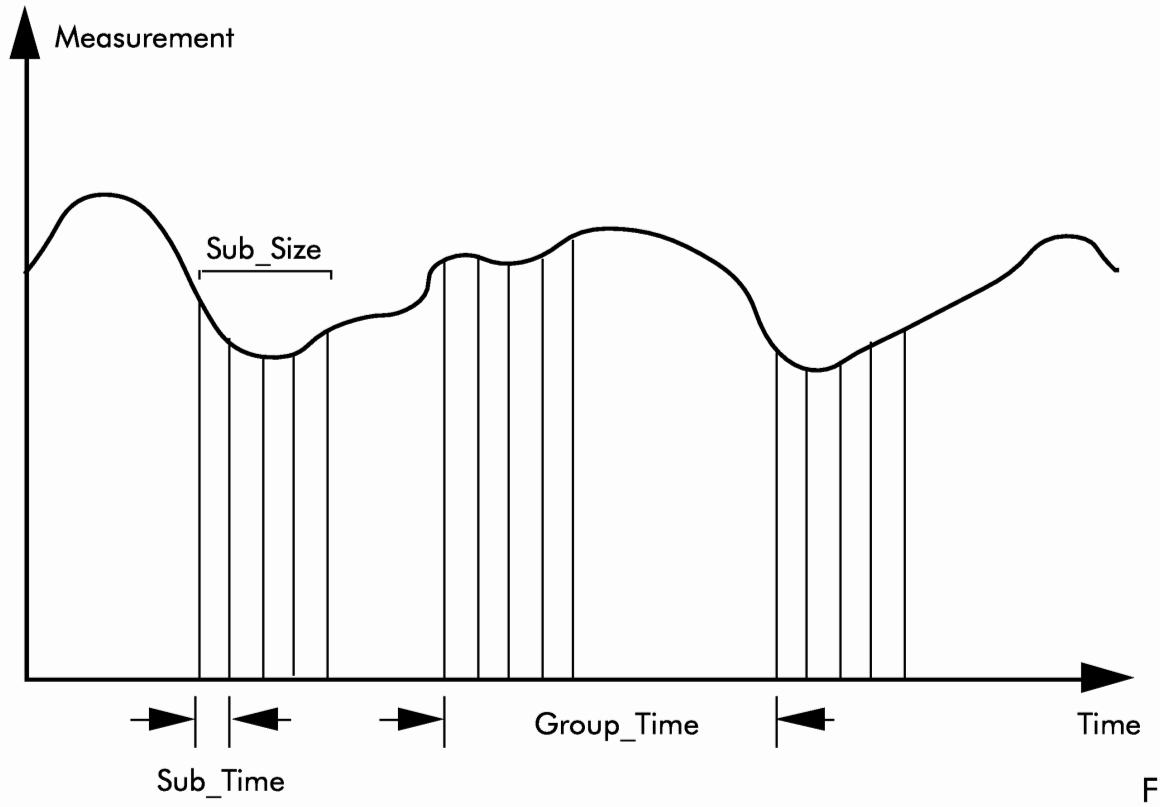

Figure 21-2 A Typical Sampling Process

If the limits for the machine or the process under study are unknown it is possible to use the collected data for obtaining the limits of the charts. The PC3000 function block does the necessary calculation. Most of the time however, the limits are known or fixed and the block is used to perform the basic book keeping of a Shewhart chart.

Another common sampling process for sequential data is to use a moving window whereby a subgroup consists of the last n data points, where n is the subgroup size. This is generally used to detect shifts in the process mean. Unfortunately, with this method of sampling the only reliable check is the out of bounds check as there is a high inter-sample correlation, thus making the bias tests rather too sensitive with high degree of false alarm. Figure 21-3 shows this sampling scheme.

It is sometimes necessary to update the control limits because of changes in the process. This can be done in two ways. The first way is to re-initiate the data gathering phase. Alternatively, just like the moving windows for raw data it is possible to calculate control limits on the basis of moving windows of subgroups.

Here the limits are calculated using the last group size of subgroups. In this way the limits are adjusted slowly from their current limits to the new ones. Of course

while updating the limits most of the tests are not valid; out of bounds tests may however be performed.

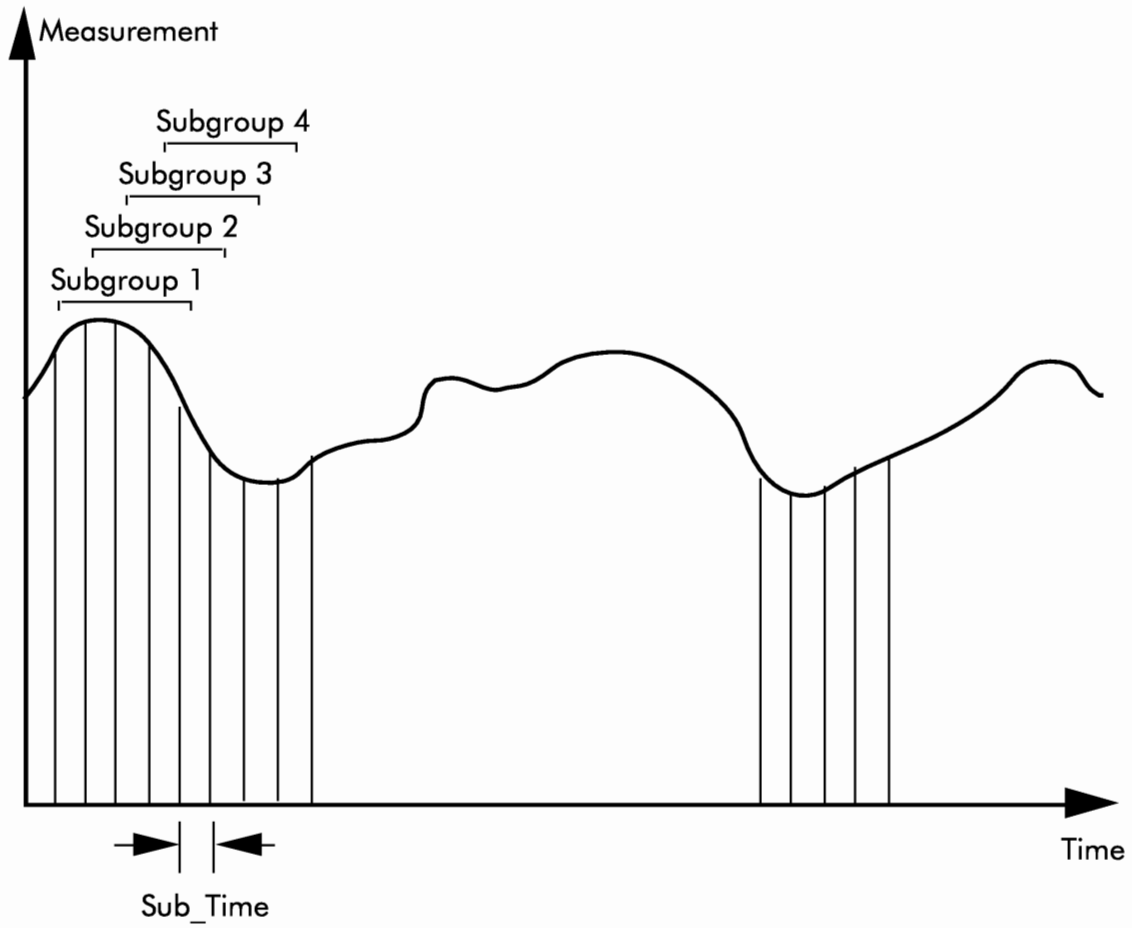

Figure 21-3 A Moving Window Sampling Process

Sampling can of course be either on the basis of time or some other event. For cases where sampling is to be synchronised to events other than time the **SPC_Event** function block can be used.

Another important aspect of statistical quality monitoring and subsequent control is associated with the distribution of the measurments. The **Histogram** function block gives an on-line histogram of the measurement. It also provides the user with running means, standard deviations, and measures of skewness and pointedness of the distribution. In addition, checks are performed for non-normality on the basis of skewness and pointedness of the distribution. A crude capability index on the basis of the number of elements in each of the segments of the histogram is also computed. This can be useful for qualitative matching with the figures from the **SPC** or **SPC_Event** function blocks. This function block provides the user with a powerful yet simple tool to examine the distribution of variables of interest.

An example of a histogram is in figure 21-4. The function block can be used to sample on demand or at regular intervals.

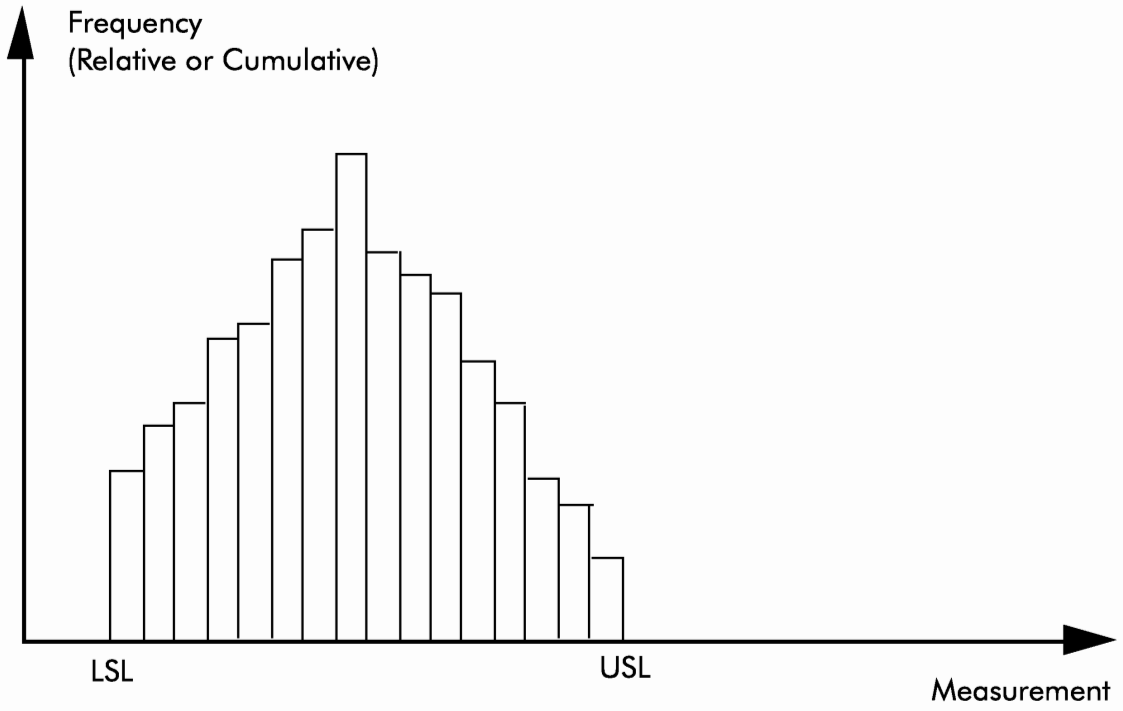

Figure 21-4 A Typical Histogram

In short, the PC3000 function blocks process the raw data to form a set of indicators which are easy to understand and interpret. The rest of the userware can then be employed to initiate certain maintenance or corrective operations to bring the process back into statistical control should it begin to drift.

Standards

The suite of SPC functions blocks adhere to the following Ford Motor Company test examples for SPC software. See Reference 3 for more details.

Tests 1,2 and 3 : These relate to the basic requirements for X bar, *s* charts which are correct indication of out of control conditions, accuracy of calculations and computation of capability indices for normal distributions.

Test 10 : Moving averages, moving *s* and moving *R* charts.

Tests 12 and 13 : Correct indication of X bar, *R* chart out of control conditions and computation of the capability indices.

Tests 8 and 9 associated with tooling change and tool wear are adhered to in so far as the indication of the out of control conditions are concerned but the corrections for the calculation of the capability indices in these cases are not performed.

Misuses of SPC

The Shewhart control chart (like all statistical techniques) must be used in cases where there is sufficient amount of confidence that the process to be monitored obeys the required characteristics envisaged in the derivation of the charts. For the charts to hold any validity at all:

The process must be under statistical control. This implies that systematic variations have been eliminated. The stability check at the end of the data gathering phase of the SPC function block goes some way in the direction of establishing stability but should not be used as the sole indication of the stability of the process.

The sampling must be such that the subgrouping ensures that the variations are primarily captured in the subgroups and intergroup variation is purely random (except when out of control conditions arise).

Two very common problems encountered in applications of SPC are correlated data and effect of intersample variations. In both cases the control limits as computed by the standard calculations (as is done in the SPC function block) may be totally wrong and the limits have to be modified after a postmortem analysis is performed.

Note that continuing with the standard SPC on these processes may lead to very misleading conclusions about the process characteristics and its capability.

References

(1) Statistical Process Control: Instruction Guide, Ford Motor Company, EU 880 A, 1985.

(2) Process Capability: Guidelines, Ford Motor Company, EU 882 A, 1990.

(3) Test Examples for SPC Software, Statistical Methods Office, Ford Motor Company, EU 33793 A, 1991.

(4) Statistical Quality Control, E.L. Grant and R.S. Leavenworth, 6th Edition, McGraw-Hill Book Co., 1988.

(5) Introduction to Statistical Quality Control, D.C. Montgomery, Second Edition, John Wiley and Sons, 1991.

(6) Biometrika tables for statisticians, E.S. Pearson and H.O. Hartley, Cambridge University Press, 1966.

SPC FUNCTION BLOCK

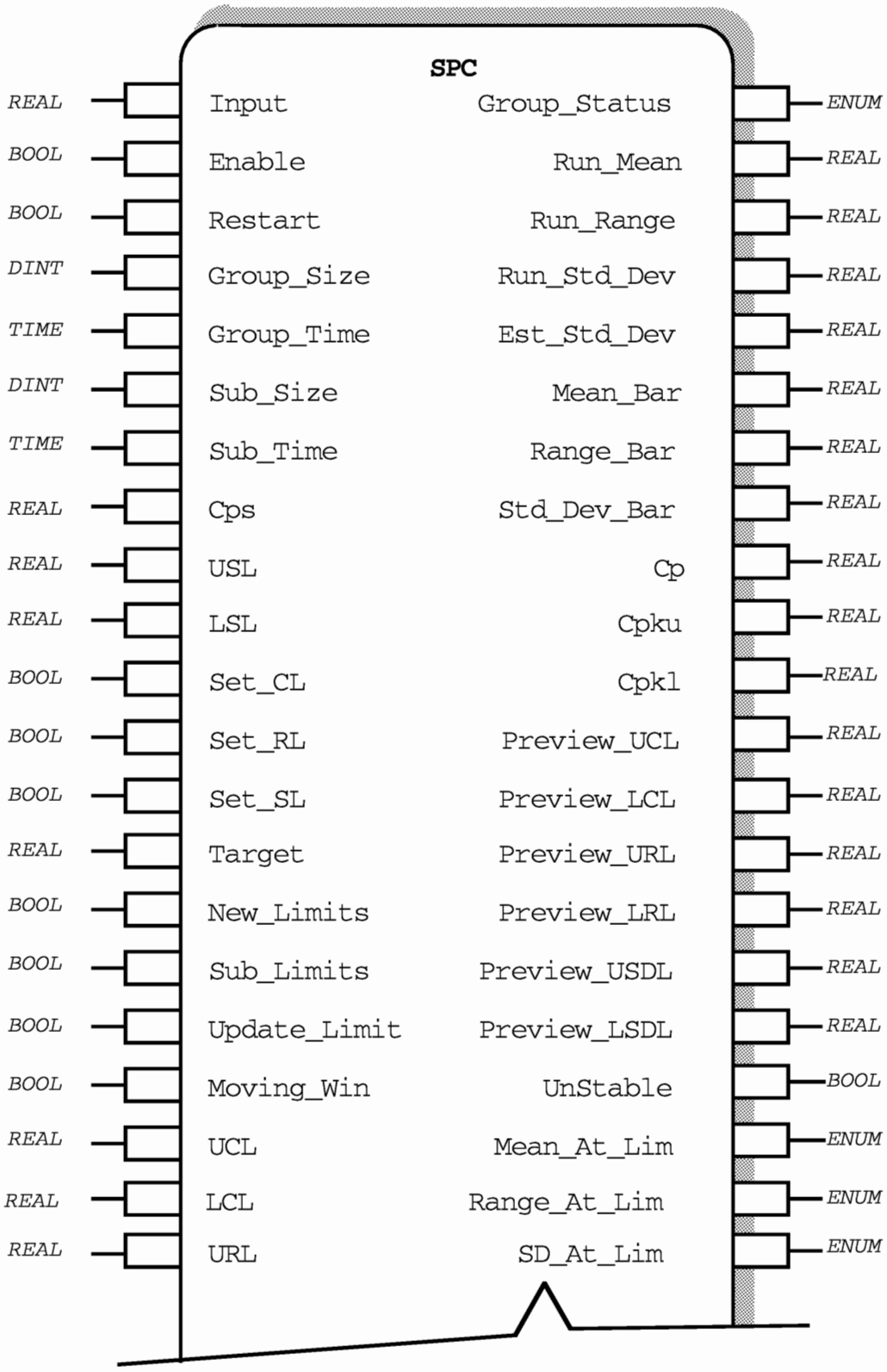

Figure 21-5 SPC Function Block Diagram

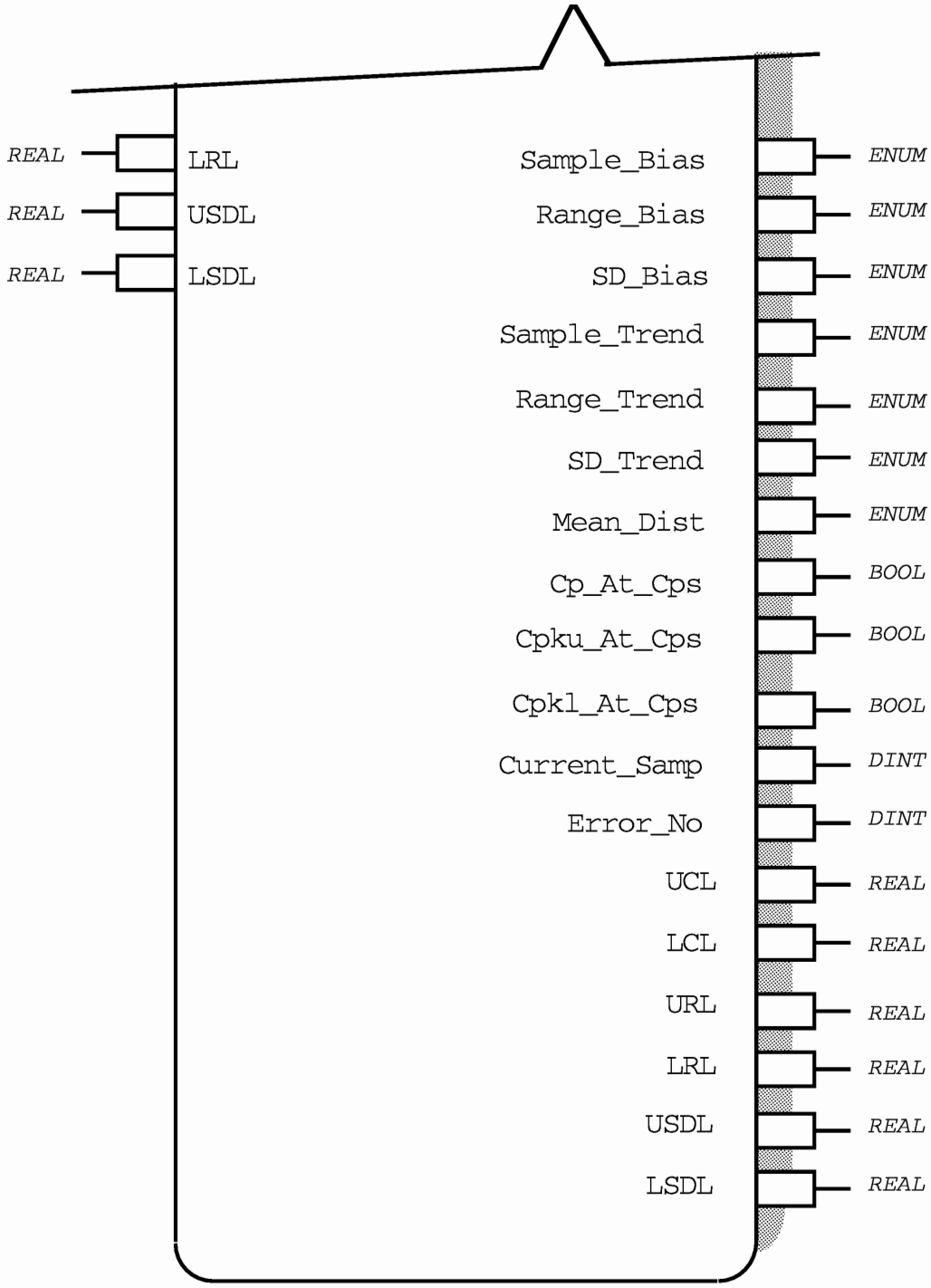

Figure 21-5 SPC Function Block Diagram (Cont)

Functional Description

The **SPC** function block is designed to perform the following tasks:

* Compute and store the appropriate indicators (i.e. **Run_Mean, Run_Range, Run_Std_Dev, Mean_Bar, Range_Bar, Std_Dev_Bar, Est_Std_Dev**) for a continuous measurement (Input);

* Compute the upper and lower control and range limits (**UCL, LCL, URL, LRL**) associated with the underlying process once **Group_Size** of subgroups are collected or be able to use predefined values for these limits;
* Compute intermediate values of control limits **Preview_LCL, Preview_UCL**, range limits **Preview_LRL, Preview_URL** and standard deviation limits (**Preview_LSDL, Preview_USDL**) at 25%, 50% and 75% of the full group;
* Update the control limits on demand based on the data in a moving window of **Group_Size** data points;
* Examine whether the underlying process is under statistical control (i.e.sample mean, range and standard deviations are within the bounds, do not have a bias away from the calculated mean of sample means and do not drift, and that the distribution of X bar is roughly normal i.e. 2/3 of the values fall within the middle third of the limits);
* Compute the process capabilities namely the overall process capability plus upper and lower process capabilities (**Cp, Cpku, Cpkl**) and check whether these are within the required specification.
* Compute charts with moving window averages, ranges and standard deviations.
* Examine the statistical stability of the process via a retrospective check on the previous data points in the case where the limits are computed from measured data points.

There are two basic modes of operation for the **SPC** function block; data gathering and internal computation of the control and range limits, or examination of statistics of the data against the specified limits.

In the both modes Input and Enable must be set by the user. The variables **Group_Size, Group_Time, Sub_Size** and **Sub_Time** are copied and checked either at the first call or immediately after a restart. The values should be either cold started or set and then the function block should be restarted. **Cps, USL** and **LSL** are specification limits and if not set correctly will result in the function block flagging errors. It will however continue working but skips computation of the process capability indices.

If the values of the control and range limits are preset then **Set_CL, Set_RL** and **Set_SL** must be set. This inhibits update of **LCL** etc. by the function block and tests for out of specification samples are performed with the preset values. The test for bias and distribution for X is performed against the value of Target which is typically set to (**UCL + LCL**)/2}. Range and standard deviation charts are performed with (**URL + LRL**)/2} and (**USDL + LSDL**)/2} respectively.

If the limits are to be computed internally then **New_Limits** and **Sub_Limits** must be set so that the preview values are updated at regular intervals and that the computed values of **LCL** etc. are transferred. **Update_Limit** can be set to enforce

a continual update of the control limits, and process capabilities based on a moving window of the size **Group_Size**.

Typical Applications

On-going charts

When the limits of the charts are fixed and the sampling strategy has been decided upon then:

Wire the Input to the variable required for charting.

Preload the values of **UCL, LCL, Target, URL, LRL, USDL and LSDL.**

Preload **Sub_Size, Group_Size, Sub_Time** and **Group_Time** in the case of **SPC** function block.

In the case of **SPC** function block set the **Enable** input whenever the sampling has to begin. In the case of **SPC_Event** function block set **Enable** whenever a single sample has to be taken. The user program has to clear the **Enable** input before preparing for another sample.

Set **Set_CL, Set_RL, Set_SL.**

The block (**SPC** or **SPC_Event**) will continue to perform the necessary checks for chart keeping and flag any errors.

Note that if resetting is required (e.g. to change the sampling strategy) change the parameters required and a rising edge on the **Restart** parameter will restart the function block with these new values.

Computing the appropriate limits

For the case where the limits must be computed then:

Wire the Input to the variable required for charting.

Ensure **Set_CL, Set_SL, Set_RL** are clear.

Preload **Sub_Size, Group_Size, Sub_Time** and **Group_Time** in the case of **SPC** function block.

In the case of **SPC** function block set the Enable input whenever the sampling has to begin. In the case of **SPC_Event** function block set Enable whenever a single sample has to be taken. The user program has to clear the Enable input before preparing for another sample.

Set **New_Limits.**

Set **Sub_Limits** if it is desired to monitor the computed control limits before they are used. Preview values can be used for this.

The block (**SPC** or **SPC_Event**) will gather the data of the required size based on the sampling strategy and compute the limits at the end of the final subgroup. The

block will then scan the historical data for standard stability checks. If there are no out of control conditions the **UnStable** flag remains clear otherwise it will be set.

The block (**SPC** or **SPC_Event)** will continue to perform the necessary checks for updating the Shewhart control charts and flag any errors.

It is the responsibility of the user program to change or ignore the computed limits.

If it is required to change the sampling process or to restart the data collection operation, set the inputs of the function block to appropriate values (e.g. change **Sub_Time and Group_Time**) and set the **Restart** input of the function block. The user program has to clear the **Restart** parameter.

Updating the limits

If it is required to update the limits from their current value to some new set based on the incoming data then **Update_Limit** flag can be set.

This mode should not be confused with the moving window subgrouping as described in figure 21-3.

Limits can also of course be modified at any time by changing the values of **UCL** etc. externally.

Preset and Computed Limits

It is possible to use on-going values for charting and compute new values at the same time based on a specified set of data points. To do this the inputs of the function block are set to those required for on-going charts and also set the **Sub_Limits** boolean. This allows the Preview values to be computed alongside the usual charting which is going on. If there are large discrepancies the user can make judgement as to which limits are appropriate for the process.

Function Block Attributes

Type: 68 10

Class:.....STATISTICAL

DefaultTask: .....Task_2

Short List: .....Input, Enable Cp Group_Status

Parameter Descriptions

Input Parameters

Input (IN)

This is a measurement to be monitored by the function block. It may be an actual measurement or an inferred one from a formula. The value must be continuous and not on / off.

Enable (EN)

This is a switch which (de)activates the operation of the function block. When it is set to 1 (* Run *) the function block is fully functional. When set to 0 (* Reset *) only a cold start consisting of initialisation of some internal arrays and variables is performed. If at any stage after initialisation **Enable** is set to **Reset** the function block enters a hold state and is reactivated on setting Enable to Run .

Restart (RE)

Whenever **Restart** is changed from a 0 (* No *) to a 1 (* Yes *) (i.e. a rising edge) a cold start is performed. A falling edge has no influence on the function block. A restart can only be performed if the function block is enabled.

Group _Size (GS)

This parameter defines the number of subgroups in a group. It can not be set less than 1 or more than 125. This parameter is only read during the initial cold start or a restart. Changing the parameter during the operation of the function block has no influence.

Group_Time (GT)

This parameter defines the time interval between the beginning of two consecutive subgroups. It should be no smaller than **Sub_Time** x **Sub_Size.**

Sub_Size (SS)

This parameter defines the number of consecutive samples taken in each subgroup. It can not be less than 2 or more than 10.

Sub _Time (ST)

This parameter defines the sample interval between two consecutive samples.The lower limit is the task interval. The granularity of time is also the task intrerval.

Cps (CPS)

This parameter is the minimum acceptable process capability. A value of 1.33 is a typical minimum.

USL (USL)

This parameter is the Upper Specification Limit for the **Input** variable. It should be chosen compatible with **LSL** .

LSL (LSL)

This parameter is the Lower Specification Limit for the **Input** variable. It should be chosen compatible with **USL** .

Set _CL (SCL)

This parameter when set to 1 (* Yes *) will fix **LCL** and **UCL** to their assigned values before execution of the function block. A value of 0 (* No *) implies that the function block may change the value of control limits upon execution of the function block. When Set is 1 (* Yes *) tests for violating the limits and drift of sample mean are performed. The test for sample bias is performed against the **Target** .

Set _RL (SRL)

This parameter when set to 1 (* Yes *) will fix **LRL** and **URL** to their assigned values before execution of the function block. A value of 0 (* No *) implies that the function block may change the value of range limits upon execution of the function block. When Set is 1 (* Yes *) tests for violating the limits and drift of sample range are performed. The test for range bias is performed against (**URL** + **LRL**)/2 .

Set _SL (SSL)

This parameter when set to 1 (* Yes *) will fix **LSDL** and **USDL** to their assigned values before execution of the function block. A value of 0 (* No *) implies that the function block may change the value of standard deviation limits upon execution of the function block. When **Set_SL** is 1 (* Yes *) tests for violating the limits and drift of sample range are performed. The test for standard deviation bias is performed against (**USDL** + **LSDL**)/2 .

Target (T)

This is the target value for X When **Set_CL** is set then the tests for bias are performed against this value.It is typically set to (**UCL** + **LCL**)/2 .

New _Limits (NL)

When set to 1 (* Yes *) this parameter causes full group calculation of control and range limits to **LCL , UCL , LRL** and **URL .** If **SetCL, Set_RL** or **Set_SL** are set then the corresponding limits are *not* updated.

Sub _Limits (SL)

When set to 1 (* Yes *) this parameter causes partial group calculation of control and range limits. These are updated at 25%, 50%, 75% and 100% of the Group_Size.

Update _Limit (UL)

When set to 1 (* Yes *) this parameter cause recalculation of control and range limits based on a sliding window of data of the size of **Group_Size** after the group full is reached. When set to 0 (* No *) the update stops and it resumes from the same state as it was previously when set to 1 (* Yes *) again.

Moving_Win (MW)

When set this implies that the subgroups are to be formed from consecutive samples as opposed to traditional method of grouping.

Input/Output Parameters

UCL (UCL)

This parameter is the upper control limit. It is either set by the user or computed by the function block. The computed value is only transferred under the control of **Set_CL New_Limits** and **Update_Limit** parameters.

LCL (LCL)

This parameter is the lower control limit. It is either set by the user or computed by the function block. The computed value is only transferred under the control of **Set_CL, New_Limits** and **Update_Limit** parameters.

URL (URL)

This parameter is the upper range limit. It is either set by the user or computed by the function block. The computed value is only transferred under the control of **Set_CL, New_Limits** and **Update_Limit** parameters.

LRL (LRL)

This parameter is the lower range limit. It is either set by the user or computed by the function block. The computed value is only transferred under the control of **Set_CL, New_Limits** and **Update_Limit** parameters.

USDL (USD)

This parameter is the upper standard deviation limit. It is either set by the user or computed by the function block. The computed value is only transferred under the control of **Set_CL, New_Limits** and **Update_Limit** parameters.

LSDL (LSD)

This parameter is the lower standard deviation limit. It is either set by the user or computed by the function block. The computed value is only transferred under the control of **Set_CL, New_Limits** and **Update_Limit** parameters.

Output Parameters

Group_Status (GRP)

This parameter acts as an indicator that the required number of data points are collected for a complete group and the indicators **Mean_Bar Range_Bar Est_Std_Dev** and the control and range limits are computed based on the assimilated data. It is set to NotFull , Pending and Full.Pending is set when the function block is computing the limits and performing a retrospective check on the data. The retrospective check is only performed when the function block is required to compute the limits.

Run _Mean (RM) [Also referred as ($\overline{X}$ or X bar]

This parameter is the computed mean of the last subgroup of data. It is updated every time a subgroup is complete.

Run _Range (RR)

This parameter is the computed range of the last subgroup of data. It is updated every time a subgroup is complete.

Run _Std_Dev (RSD)

This parameter is the computed standard deviation of the last subgroup. It is updated every time a subgroup is complete.

Est _Std_Dev (ESD)

This parameter is the estimated standard deviation of the population where the measurements are taken from. It is updated when the group is full and anytime after if **Update_Limit** is set.

Mean _Bar (MB) [Also referred to as ($\overline{\overline{X}}$ or X bar bar]

This parameter is the computed mean of the subgroup means. It is evaluated when the group is full and anytime after if **Update_Limit** is set.

Range _Bar (RB)

This parameter is the computed mean of the subgroup ranges. It is evaluated when the group is full and anytime after if **Update_Limit** is set.

Std _Dev_Bar (STD)

This parameter is the computed mean of the subgroup standard deviations. It is evaluated when the group is full and anytime after if **Update_Limit** is set.

Cp (CP)

This parameter is the computed process capability index. It is evaluated when the group is full or anytime after when **Update_Limit** is set.

Cpku (CPU)

This parameter is the computed upper process capability index. It is evaluated when the group is full or anytime after when **Update_Limit** is set.

Cpkl (CPL)

This parameter is the computed lower process capability index. It is evaluated when the group is full or anytime after when **Update_Limit** is set.

Preview _UCL (PUC)

This parameter is the computed upper control limit. It is evaluted at 25%, 50% , 75% , and 100% of the **Group_Size** if **Sub_Limit** is set.

Preview _LCL (PLC)

This parameter is the computed lower control limit. It is evaluted at 25%, 50% , 75% , and 100% of the **Group_Size** if **Sub_Limit** is set.

Preview_URL (PUR)

This parameter is the computed upper range limit. It is evaluted at 25%, 50% , 75% , and 100% of the **Group_Size** if **Sub_Limit** is set.

Preview_LRL (PLR)

This parameter is the computed lower range limit. It is evaluted at 25%, 50% , 75% , and 100% of the **Group_Size** if **Sub_Limit** is set.

Preview _USDL (PUS)

This parameter is the computed upper standard deviation limit. It is evaluted at 25%, 50% , 75% , and 100% of the **Group_Size** if **Sub_Limit** is set.

Preview _LSDL (PLS)

This parameter is the computed lower standard deviation limit. It is evaluted at 25%, 50% , 75% , and 100% of the **Group_Size** if **Sub_Limit** is set.

UnStable (US)

On completion of the group the block checks the measurements retrospectively and if it detects out of control conditions it sets the **UnStable** flag. This is only performed if the block is required to compute the limits from that set of measurements.

Mean _At_Lim (MAL)

This parameter is set to 1 (* Low *) when the subgroup mean is equal to or smaller than the lower control limit. It is set to 2 (* High *) when the subgroup mean is equal or larger than the upper control limit. This test is performed if either the group is full or **Set_CL** is set.

Range _At_Lim (RAL)

This parameter is set to 1 (* Low *) when the subgroup range is equal to or smaller than the lower range limit. It is set to 2 (* High *) when the subgroup range is equal or larger than the upper range limit. This test is performed if either the group is full or **Set_RL** is set.

SD_At_Lim (SAL)

This parameter is set to 1 (* Low *) when the subgroup standard deviation is equal to or smaller than the lower standard deviation limit. It is set to 2 (* High *) when the subgroup standard deviation is equal or larger than the upper limit. This test is performed if either the group is full or **Set_SL** is set.

SampleBias (SAB)

This parameter is set to1 (* Low *) when 7 consecutive running means are below mean of means. It is set to 2 (* High *) when 7 consecutive running means are above the mean of means.

Range_Bias (RAB)

This parameter is set to 1 (* Low *) when 7 consecutive running ranges are below mean of ranges. It is set to 2 (* High *) when 7 consecutive running ranges are above the mean of ranges.

SD _Bias (SDB)

This parameter is set to 1 (* Low *) when 7 consecutive running standard deviations are below mean of standard deviations. It is set to 2 (* High *) when 7 consecutive running standard deviations are above the mean of standard deviations.

Sample _Trend (SAT)

This parameter is set to 1 (* Low *) when 7 consecutive running means are drifting downwards. It is set to 2 (* High *) when 7 consecutive running means are drifting upwards.

Range_Trend (RAT)

This parameter is set to 1 (* Low *) when 7 consecutive running ranges are drifting downwards. It is set to 2 (* High *) when 7 consecutive running ranges are drifting upwards.

SD_Trend (SDT)

This parameter is set to 1 (* Low *) when 7 consecutive running standard deviations are drifting downwards. It is set to 2 (* High *) when 7 consecutive running standard deviations are drifting upwards.

Mean _Dist (MD)

This is set if out of 25 subgroups more than 92% or less than 40% of the subgroup means fall inside the middle third of the distribution. When more than 92% are in the middle third the flag is set to **Lev_92** and if below 40% present it is set to **Lev_40**.

Cp _At_Cps (CAC)

This parameter is set to 1 (* On *) when the computed process capability is equal to or less than the specified limit. This test is performed when the group is full.

Cpku _At_Cps (CUC)

This parameter is set to 1 (* On *) when the computed upper process capability is equal to or less than the specified limit. This test is performed when the group is full.

Cpkl_At_Cps (CLC)

This parameter is set to 1 (* On *) when the computed lower process capability is equal to or less than the specified limit. This test is performed when the group is full.

Current _Samp (CS)

This parameter is the sample number since the last cold start or restart of the function block.

Error_No (ERN)

This parameter is the error number indicating inconsistency in the input to the function block or a computed parameter inside the function block.

0 ..... Normal;

1 ..... Error in timing specification;

10 ..... **Sub_Size** too big or too small;

100 ..... **Group _Size** too big or too small;

1000 ..... Specification limits **USL, LSL** incompatible;

10000 ..... Calculated standard deviation too small for subsequent calculations.

Any number in between indicates a combination of these errors. The function block will continue to function partially for the last two errors.

Parameter Attributes

Name	Mnem-onic	Type	Cold Start	Wire able	Read Access	Write Access	Type Specific Information	
Input	**IN**	**REAL**	0	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Variable
Enable	**EN**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Restart	**RE**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Group_Size	**GS**	**DINT**	10	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 125 Fixed as 1 Variable
Group_Time	**GT**	**TIME**	15m	Yes	Oper	Oper	None	
Sub_Size	**SS**	**DINT**	5	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 10 Fixed as 2 Variable
Sub_Time	**ST**	**TIME**	5s	Yes	Oper	Oper	None	
Cps	**CPS**	**REAL**	1.333	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 None
USL	**USL**	**REAL**	0	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
LSL	**LSL**	**REAL**	0	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Set_CL	**SCL**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Set_RL	**SRL**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Set_SL	**SSL**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Target	**T**	**REAL**	0	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input

Table 21-1 SPC Input Parameter Attributes

Name	Mnem-onic	Type	Cold Start	Wire able	Read Access	Write Access	Type Specific Information	
New_Limits	**NL**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Sub_Limits	**SL**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Update_Limit	**UL**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Moving_Win	**MW**	**BOOL**	No (0)	Yes	Oper	Oper	Senses	Fixed as Yes (1) Fixed as No (0)

Table 21-1 SPC Input Parameter Attributes (Cont)

Name	Mnem-onic	Type	Cold Start	Wire able	Read Access	Write Access	Type Specific Information	
UCL	UCL	REAL	0		Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
LCL	LCL	REAL	0		Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
URL	URL	REAL	0		Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
LRL	LRL	REAL	0		Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
USDL	USD	REAL	0		Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
LSDL	LSD	REAL	0		Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input

Table 21-2 SPC Input /Output Parameter Attributes

Name	Mnem-onic	Type	Cold Start	Read Access	Type Specific Information	
Group_Status	**GRP**	**ENUM**	NotFull(0)	Oper	Strings	NotFull (0) Pending (1) Full (2)
Run_Mean	**RM**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Run_Range	**RR**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Run_Std_Dev	**RSD**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Est_Std_Dev	**ESD**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Mean_Bar	**MB**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Range_Bar	**RB**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Std_Dev_Bar	**STD**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Cp	**CP**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 None
Cpku	**CPU**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 None
Cpkl	**CPL**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 None
Preview_UCL	**PUC**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input

Table 21-3 SPC Output Parameter Attributes

Name	Mnem-onic	Type	Cold Start	Read Access	Type Specific Information	
Preview_LCL	**PLC**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Preview_URL	**PUR**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Preview_LRL	**PLR**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Preview_USDL	**PUS**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
Preview_LSDL	**PLS**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
UnStable	**US**	**BOOL**	No (0)	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Mean_At_Lim	**MAL**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)
Range_At_Lim	**RAL**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)
SD_At_Lim	**SAL**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)
Sample_Bias	**SAB**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)
Range_Bias	**RAB**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)
SD_Bias	**SDB**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)
Sample_Trend	**SAT**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)

Table 21-3 SPC Output Parameter Attributes (Cont)

Name	Mnem-onic	Type	Cold Start	Read Access	Type Specific Information	
Range_Trend	**RAT**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)
SD_Trend	**SDT**	**ENUM**	OK (0)	Oper	Strings	OK (0) High (1) Low (2)
Mean_Dist	**MD**	**ENUM**	OK (0)	Oper	Strings	OK (0) 92_Lev (1) 40_Lev (2)
Cp_At_Cps	**CAC**	**BOOL**	No (0)	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Cpku_At_Cps	**CUC**	**BOOL**	No (0)	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Cpkl_At_Cps	**CLC**	**BOOL**	No (0)	Oper	Senses	Fixed as Yes (1) Fixed as No (0)
Current_Samp	**CS**	**DINT**	0	Oper	High Limit Low Limit Units	Fixed as 2147483647 Fixed as 0 Variable
Error_No	**ERN**	**DINT**	0	Oper	High Limit Low Limit Units	Fixed as 11111 Fixed as 0 Variable

Table 21-3 SPC Output Parameter Attributes (Cont)

SPC _EVENT FUNCTION BLOCK

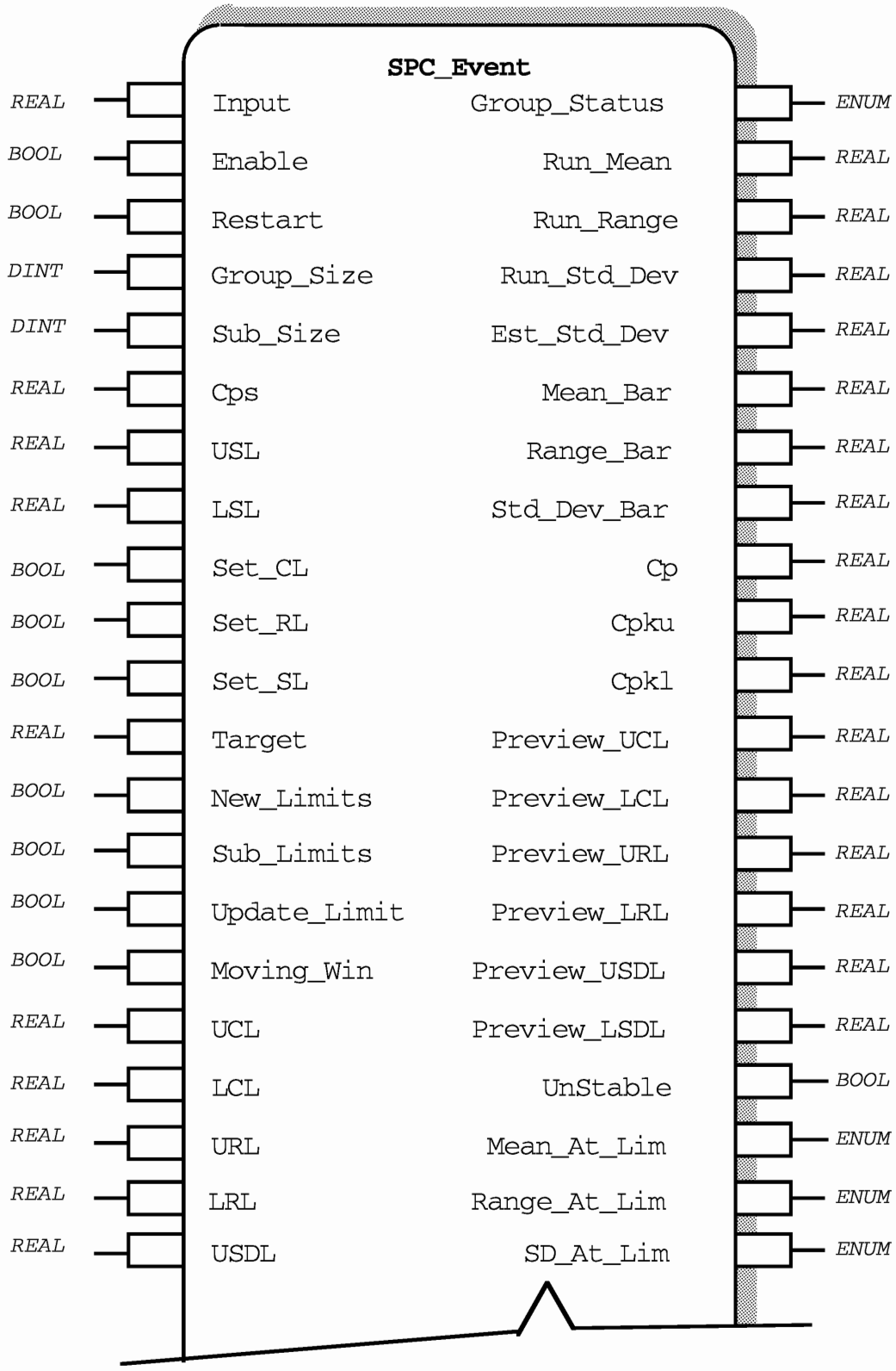

Figure 21-6 SPC _Event Function Block Diagram

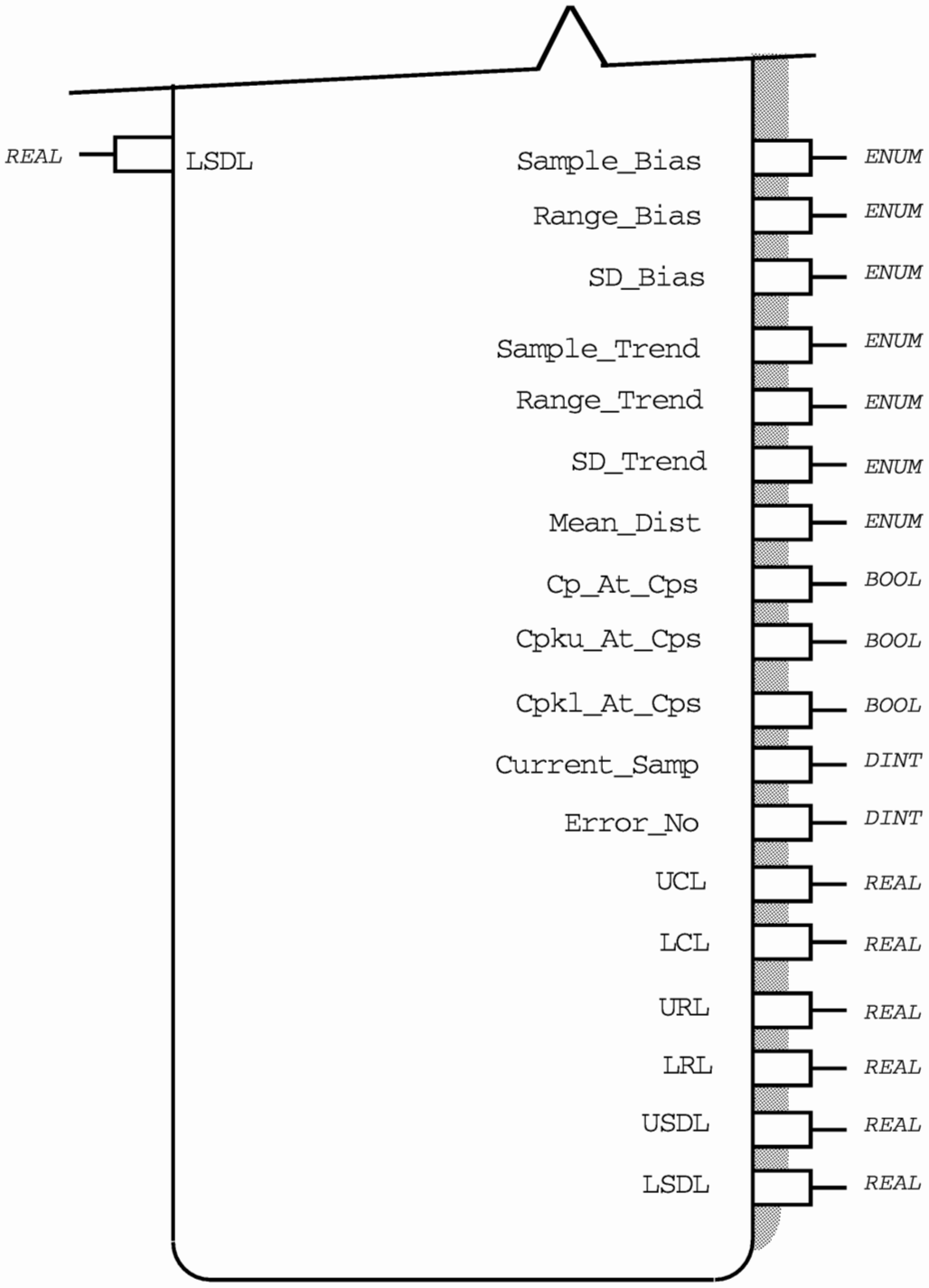

Figure 21-6 SPC _Event Function Block Diagram (Cont)

Functional Description

Overall functionality is the same as the **SPC** function block. The **SPC_Event** is used for event based sampling rather than time based. There are therefore no **Group_Time** or **Sub_Time** inputs.

A sample is taken by the **Enable** input changing from 0 to 1. This must be implemented in the User program which must also reset Enable for the next sample.

For examples see SPC function block description.

Function Block Attributes

Type: 68 30

Class:..... STATISTICAL

DefaultTask:..... Task_2

Short List:..... Input, Enable Cp Group_Status

Parameter Descriptions

Input Parameters

Input (IN)

This is a measurement to be monitored by the function block. It may be an actual measurement or an inferred one from a formula. The value must be *continuous* and not on / off.

Enable (EN)

On a transition from 0 to 1 the function block samples the **Input** and another measurement is added to the appropriate subgroup. Leaving the **Enable** signal at 1 has no other influence. The User program must reset the **Enable** ready for the next sample.

Restart (RE)

Whenever **Restart** is changed from a 0 (* No *) to a 1 (* Yes *) (i.e. a rising edge) a cold start is performed. A falling edge has no influence on the function block. A restart can only be performed if the function block is enabled.

Group _Size (GS)

This parameter defines the number of subgroups in a group. It can not be set less than 1 or more than 125. This parameter is only read during the initial cold start or a restart. Changing the parameter during the operation of the function block has no influence.

Sub _Size (SS)

This parameter defines the number of consecutive samples taken in each subgroup. It can not be less than 2 or more than 10.

Cps (CPS)

This parameter is the minimum acceptable process capability. A value of1.33 is a typical minimum.

This and all the remaining parameters are the same as the standard SPC function block.

Parameter Attributes (See page 21-13).

Parameters are the same as the standard SPC function block.

HISTOGRAM FUNCTION BLOCK

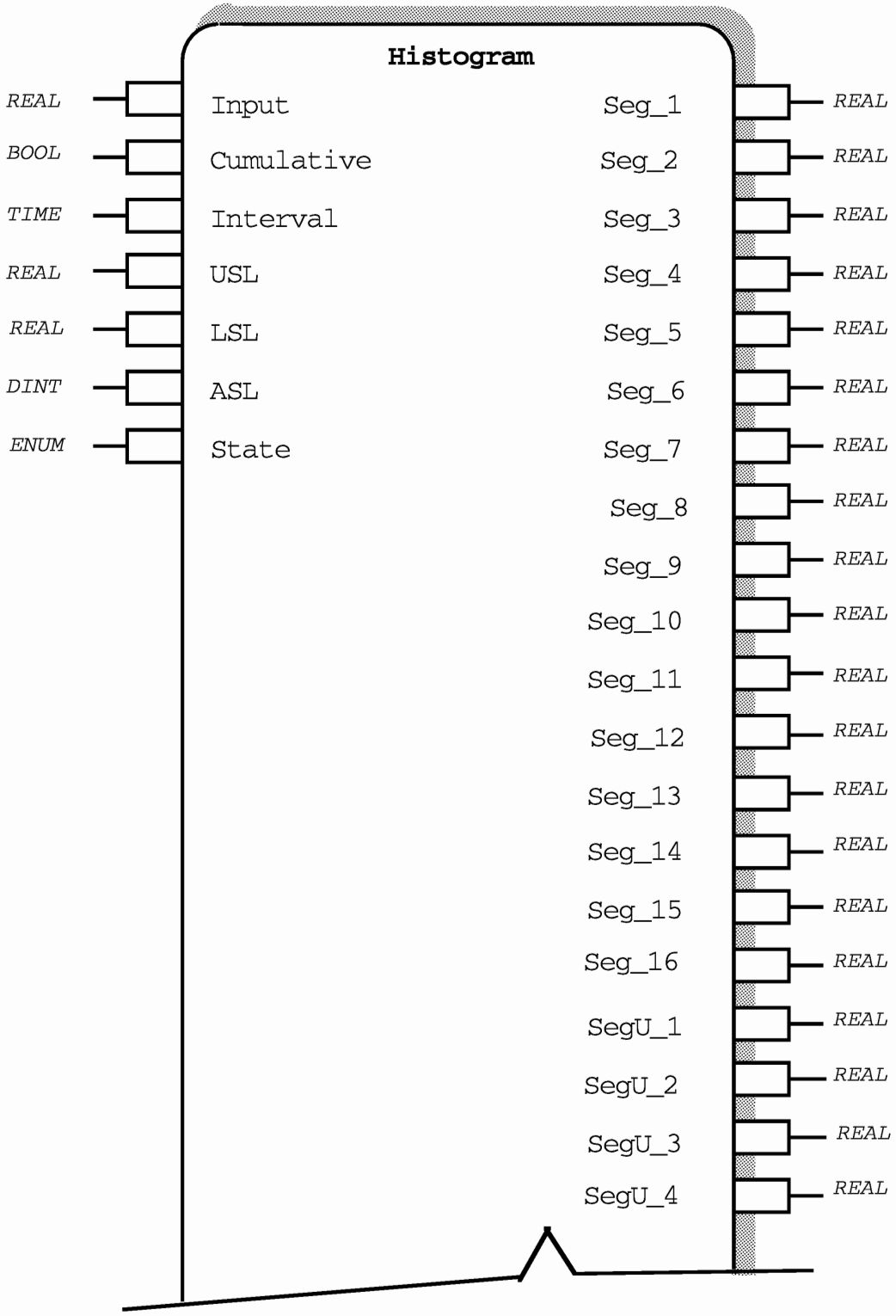

Figure 21-7 Histogram Function Block Diagram

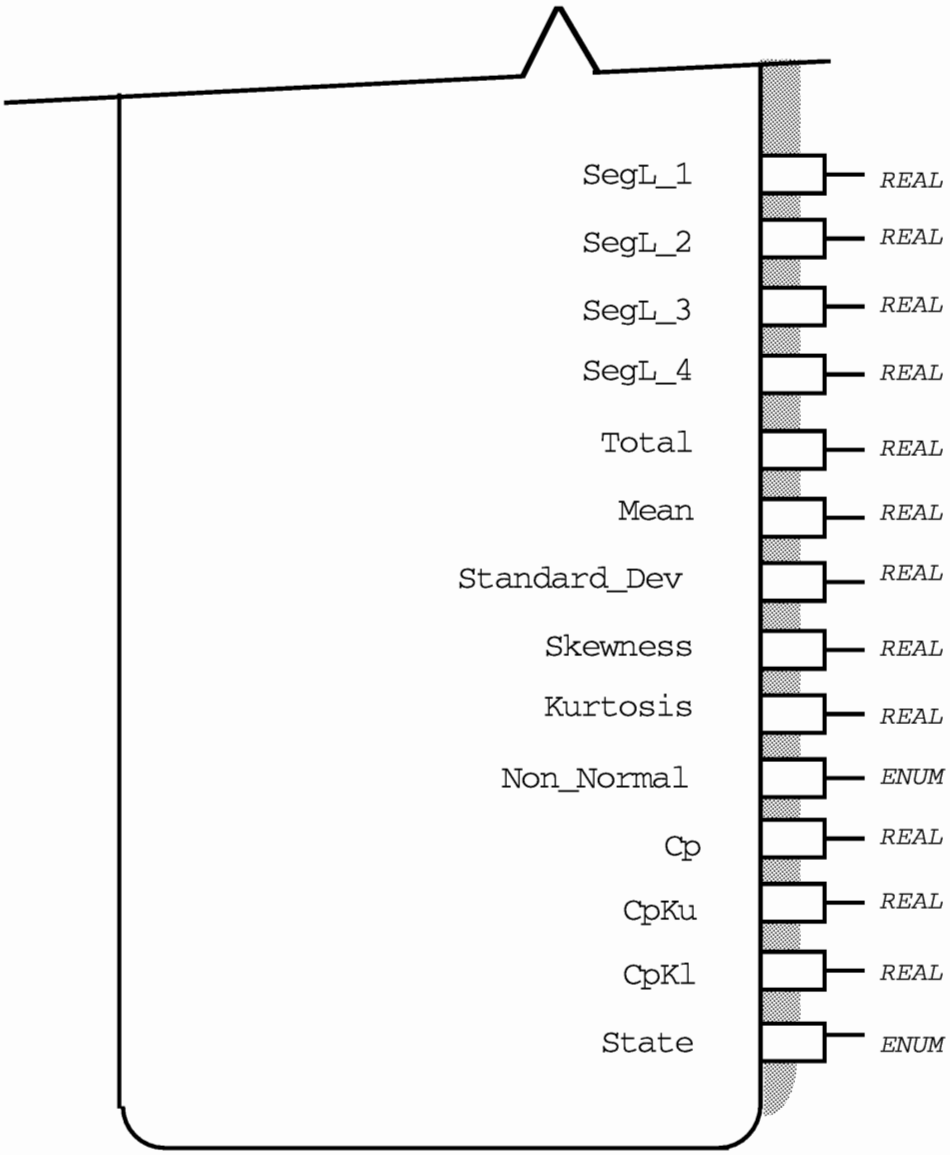

Figure 21-7 Histogram Function Block Diagram (Cont)

Functional Description

The Histogram function block computes the distribution of a REAL measurement. The user specifies the upper and lower specification limits of the variable and the function block divides this interval to 16 equispaced segments from which the distribution of the variable is computed.

The function block has a 50 sample memory. Every time the specification limits change all of the information except for the last 50 samples are cleared.

The function block computes the running average, standard deviation, skewness and pointedness of the measurement starting from the the time the sampling began to the current measurement. It is possible to use the variable **ASL** (Asymptotic Sample Length) to shorten this window to *approximately* ASL samples. The

approximation is because the window is an exponential one with an effective time constant of ASL samples. This allows tracking of slowly varying means. The histogram part of the function block is *not* affected by the choice of **ASL**. Typical values of **ASL** range between 100 to 1000 samples or so.

The function block provides two other pieces of information:

It gives an indication of whether the distribution of the sampled measurements is either too skew, too flat or too pointed to be from a normal (Gaussian) distribution.

An estimate of process capability index from the histogram data is also given. The estimate though crude, gives quite good indication of actual process capability index

It is possible to use the function block for regular sampling on the basis of time or on demand. It is also possible to reinitialise the block at any point during the execution.

When triggered the function block recompiles the new outputs over 4 executions i.e. 4 x Task Interval

In order to obtain a histogram of a REAL variable

Wire the variable to Input.

Either set the Interval to a desired length of time (it *must* be at least 5 times the task interval) and set the **State** to **SmpCont** or set the **State** to **Demand** every time a sample is required. In the latter case the State reverts to OK once the calculations have been performed.

Preload **USL** and **LSL**. Every time these values are changed the only the last 50 samples are remembered.

If *continuous* forgetting is required it is possible to do so by setting **ASL** (Asymptotic Sample Length) to the required value (say 100 samples). This means that the effective window of the data for the running Mean, Standard deviation etc. is 100 samples. The choice of **ASL** does not affect the frequency values in the segments.

Note that when **Cumulative** is set the function block generates the cumulative (actual count) of the frequencies in each segment. Otherwise the relative frequency (i.e. actual count divided by the total number of samples taken) is given.

Function Block Attributes

Type: 68 20

Class:..... STATISTICAL

DefaultTask:..... Task_2

Short List:..... Input Cumulative Interval State

Parameter Descriptions

Input Parameters

Input (IN)

This is the input which is either regularly sampled or sampled on demand depending on the value of state.

Cumulative (C)

Set to Yes (0) each segment contains the cumulative number of samples and the total of all segments = Total.

Set to No (1) each segment contains the relative number of samples (cumulative/Total) and the total of all segments = 1.00.

Interval (INT)

This is the interval of time between samples from the measurement in the continuous sample mode. The minimum value of the **Interval** is 5 times the task interval. If the task interval is set greater than one second then the minimum interval is the task interval.

USL (USL)

This is the upper specification level for the measurement. Changing **USL** during the operation of the function block causes recalculation of the segements and the distribution using the last 50 samples of data which is kept in a cicular buffer internally.

LSL (LSL)

This is the lower specification level for the measurement. Changing **LSL** during the operation of the function block causes recalculation of the segements and the distribution using the last 50 samples of data which is kept in a cicular buffer internally.There are 16 equispaced segments between **USL** and **LSL** and 4 segments either side outside the limits.

ASL (ASL)

This is the asymptotic sample length used for exponential forgetting of data in the histogram. The effective window of data used is **ASL** samples long. Forgetting only affects the values of mean, standard deviation, skewness and kurtosis.

Input/Output Parameters

State (S)

This is an enumerated type and can take values of:

0	Disable
1	SmpCont
2	Demand
3	Pending
4	OK
5	Init

If **State** is set to **SampCont** the block samples the **Input** at regular intervals set by **Interval.** If **State** is set to **Demand** then on completion of the calculation the block changes **State** to **OK** ready for the next sample. During the calculation the block will set **State** to **Pending** and no sampling of the measurement can take place until the calculations are over. If the user sets the **State** to **Pending** when it is not the block will revert back to **OK** and vice versa. On setting the **State** to **Init** the function block reinitialises the histogram generation.

The calculation is spread over 4 execution cycles i.e. 4 x Task interval.

Output Parameters

Seg (S1)

Frequency in the segment 1 in the histogram.

Seg (S2)

Frequency in the segment 2 in the histogram.

etc.

Seg (S16)

Frequency in the segment 16 in the histogram.

SegU_1 (SU1)

Frequency of all measurements falling outside **USL** and below **USL** + (**USL-LSL**)/16.

SegU_2 (SU2)

Frequency of all measurements falling outside **USL** + (**USL-LSL**)/16 and below **USL** + 2*(**USL-LSL**)/16.

SegU_3 (SU3)

Frequency of all measurements falling outside **USL** + 2*(**USL-LSL**)/16 and below **USL** + 3*(**USL-LSL**)/16.

SegU_4 (SU4)

Frequency of all measurements falling outside **USL** + 3*(**USL-LSL**)/16.

SegL_1 (SL1)

Frequency of all measurements falling outside **LSL** - 3*(**USL-LSL**)/16.

SegL_2 (SL2)

Frequency of all measurements falling above **LSL** - 3*(**USL-LSL**)/16 and below **LSL** - 2*(**USL-LSL**)/16.

SegL_3 (SL3)

Frequency of all measurements falling above **LSL** - 2*(**USL-LSL**)/16 and below **LSL** - (**USL-LSL**)/16.

SegL (SL4)

Frequency of all measurements falling above **LSL** - (**USL-LSL**)/16 and below **LSL**.

Total (TOT)

The total number of data points gathered.

Mean (M)

The running average of the **Input** sequence.

Standard _Dev (STD)

The running standard deviation of the **Input** sequence.

Skewness (SK)

The running skewness of the **Input** sequence.

Kurtosis (K)

The running kurtosis of the **Input** sequence.

Non _Normal (NN)

Set to **Normal** if there is no positive indication of non-normality, set to **Skew** if there is an indication of skewness, set to **Kurtic** if there is indication of high or low kurtosis, and set to S if there is an indication of abnormal skewness and kurtosis.

Cp (CP)

Estimated process capability index based on the segments in the histogram.

CpKu (CKU)

Estimated upper process capability index based on the segments in the histogram.

CpKl (CKL)

Estimated lower process capability index based on the segments in the histogram.

Parameter Attributes

Name	Mnem-onic	Type	Cold Start	Wire able	Read Access	Write Access	Type Specific Information	
Input	**IN**	**REAL**	0	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Variable
Cumulative	**C**	**BOOL**	Yes (0)	Yes	Oper	Oper	Senses	Fixed as No (1) Fixed as Yes (0)
Interval	**INT**	**TIME**	0ms	Yes	Oper	Oper	None	
USL	**USL**	**REAL**	0	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
LSL	**LSL**	**REAL**	0	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100,000 Same as Input
ASL	**ASL**	**DINT**	0	Yes	Oper	Oper	High Limit Low Limit Units	Fixed as 12147483647 Fixed as 2 Variable

Table 21-4 Histogram Input Parameter Attributes

Name	Mnem onic	Type	Cold Start	Wire able	Read Access	Write Access	Type Specific Information	
State	**S**	**ENUM**	Disable (0)		Oper	Oper	Strings	Disable (0) SmpCnt (1) Demand (2) Pending (3) OK (4) Init (5)

Table 21-5 Histogram Input/Output Parameter Attributes

Name	Mnem-onic	Type	Cold Start	Read Access	Type Specific Information	
Seg_1	**S1**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
Seg_2	**S2**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
etc. etc.						
Seg_16	**S16**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
SegU_1	**SU1**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
SegU_2	**SU2**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
SegU_3	**SU3**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
SegU_4	**SU4**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
SegL_1	**SL1**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
SegL_2	**SL2**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
SegL_3	**SL3**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
SegL_4	**SL4**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None

Table 21-6 Histogram Output Parameters Function Block

Name	Mnem -onic	Type	Cold Start	Read Access	Type Specific Information	
Total	**TOT**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as 0 None
Mean	**M**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100000 Same as Input
Standard_Dev	**STD**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100000 Same as Input
Skewness	**SK**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100000 None
Kurtosis	**K**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100000 None
Non_Normal	**NN**	**ENUM**	Normal (0)	Oper	Strings	Normal (0) Skew (1) Kurtic (2) S_and_K (3)
Cp	**CP**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100000 None
CpKu	**CKU**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100000 None
CpKl	**CKL**	**REAL**	0	Oper	High Limit Low Limit Units	Fixed as 100,000 Fixed as -100000 None

Table 21-6 Histogram Output Parameters Function Block (Cont)

STATISTICS FUNCTION BLOCK

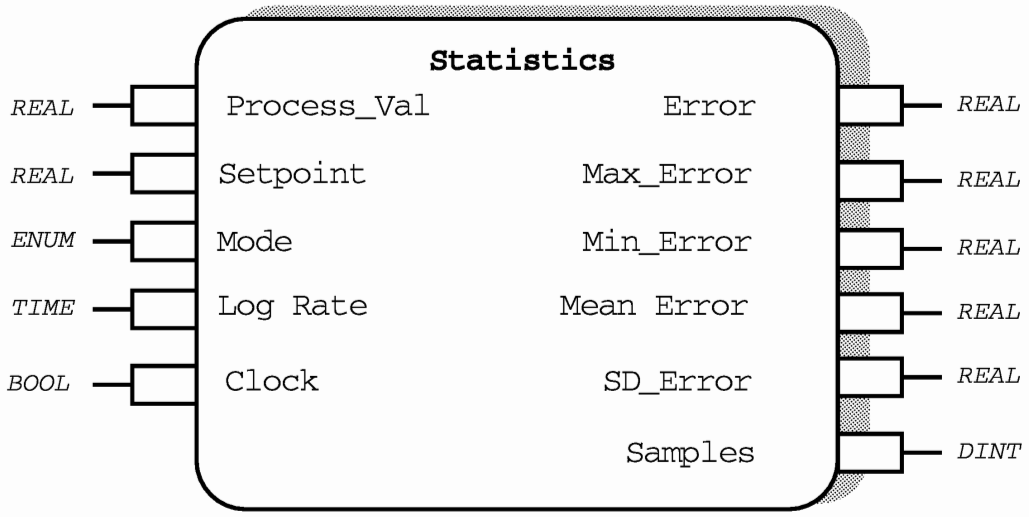

Figure 21-8 Statistics Function Block Diagram

Functional Description

This function block provides statistical information on the differences between a Process_Val and a Setpoint value. This could perhaps be applied to the Process_Val and Setpoint inputs of a control loop to provide extra information to operations staff via a user interface without implementing the full complexity of what is usually understood by the term Statistical Process Control.

Function Block Attributes

Type: 68 38

Class:..... STATISTICAL

DefaultTask:..... Task_2

Short List:..... Process_Val, Setpoint, Error, SD_Error

Memory Requirements: ..... 68 Bytes

Parameter Descriptions

Process_Val (PV)

The variable value, either direct from the plant or in some way derived that is to be monitored and compared with the **Setpoint.**

Setpoint (SP)

The essentially static value against which the **Process_Val** is examined for **Error.**

Mode (M)

The mode parameter is used to control the operation of the function block:

Reset:	All outputs are reset and statistical integrators are reset. No further computations are carried out.
Hold:	Outputs hold their current states and all internal statistical integrators are held.
RunLog:	The function block carries out all calculations but only on samples taken from the **Process_Val** and **Setpoint** at intervals determined by **Log_Rate.** The first sample is taken after an interval of **Log_Rate** from instigating **RunLog** mode.
RunClk:	The function block carries out all calculations but only on samples taken from the **Process_Val** and **Setpoint** at intervals triggered by the **Clock** input. Samples are taken on the execution cycle of the function block when the **Clock** input goes from false to true.
RunCont	The function block carries out all calculations but only on samples taken from the **Process_Val** and **Setpoint** at intervals defined by the task interval of the function block.

Log_Rate (R)

The sample rate used when the function block is in **RunLog** mode.

Clock (CLK)

The trigger used to cause a sample to be taken when in **RunClk** mode. It must be returned from On(1) to Off(0) by the user program - it is not so returned by the function block.

Error (ERR)

The difference between the most recent samples of **Process_Val** and **Setpoint.** This is not necessary the same as the difference between the current values of the inputs **Process_Val** and **Setpoint.** The obtaining of the current samples is governed by the input **Mode**.

Max_Error (MXE)

The highest value of **Error** recorded since the function block was last reset.

Min_Error (MNE)

The lowest value of **Error** recorded since the function block was last reset.

Mean_Error (MN)

The arithmetic mean of all values of **Error** recorded since the function block was last reset.

SD_Error (SDE)

The standard deviation of all values of **Error** recorded since the function block was last reset. This is calculated according to the small sample equation for standard deviation:

$$\text{SD\_Error} = \sqrt{\left(\frac{\Sigma\,(\text{Error})^2}{n-1} - \frac{(\Sigma\,\text{Error})^2}{n(n-1)}\right)}$$

Samples

The number of samples taken since the function block was last reset. This number of samples is the set of values on which the current outputs are based.

Parameter Attributes

Name	Mnemonic	Type	Cold Start	Read Access	Write Access	Type Specific Information	
Process_Val	**PV**	**REAL**	0	Oper	Oper	High Limit Low Limit	+3·402823E+38 -3·402823E+38
Setpoint	**SP**	**REAL**	0	Oper	Oper	High Limit Low Limit	+3·402823E+38 -3·402823E+38
Mode	**M**	**ENUM**	Reset (0)	Oper	Oper	See parameter List	
Log_Rate	**R**	**TIME**	1s	Oper	Oper	High Limit Low Limit	23d23h59m59s999ms 0
Clock	**CLK**	**BOOL**	Off (0)	Oper	Block	Senses	Off (0) On (1)
Error	**ERR**	**REAL**	0	Oper	Block	High Limit Low Limit	+3·402823E+38 -3·402823E+38
Max_Error	**MXE**	**REAL**	0	Oper	Block	High Limit Low Limit	+3·402823E+38 -3·402823E+38
Min_Error	**MNE**	**REAL**	0	Oper	Block	High Limit Low Limit	+3·402823E+38 -3·402823E+38
Mean_Error	**MN**	**REAL**	0	Oper	Block	High Limit Low Limit	+3·402823E+38 -3·402823E+38
SD_Error	**SDE**	**REAL**	0	Oper	Block	High Limit Low Limit	+3·402823E+38 -3·402823E+38
Samples		**DINT**	0	Oper	Block	High Limits Low Limits	2147483647 0

Table 21-7 Statistics Parameter Attributes