10115-J-NPG Revision 1, "DCPP Units 1 & 2 Process Protection System Replacement Controller Transfer Function Specification" (LAR Reference 120)

Specification No.: 10115-J-NPG, Rev. 1

Title: Process Protection System Controller Transfer Functions Design Input Specification

Project: Diablo Canyon Power Plant, Units 1 & 2 Date 03/09/2011

Department/Group: Engineering Projects/Instrumentation and Controls

Structure, System or Component: Process Protection System (System 38)

Type or Purpose of Specification: Design of Controller Transfer Functions for the Process Protection System (PPS)

No. of Sheets: 31 (inc. cover)

Nuclear Safety Related: Yes X No ___

10CFR21 Applies: Yes X No ___

Graded Quality: Yes ___ No X

	Signature	Section	Date
PREPARER:	R Lint	PSSA	03/08/2011
VERIFIER:	J Hefler	PSSA	03/08/2011
COORD SECT:	J Reinholdt	PSSA	03/08/2011
SEISMIC APPROVAL:	N/A	N/A	N/A
ENVIRON. APPROVAL:	N/A	N/A	N/A
LEAD MGR APPROVAL:	R Klimczak	DE	03/09/2011

Enter the professional engineer's (PE) full name and registration number, or stamp, or seal and expiration date in this space.

REGISTERED PROFESSIONAL ENGINEER
JOHN W. HEFLER
No. E 9995
Expires 9/30/2011
ELECTRICAL
STATE OF CALIFORNIA

RECORDS OF REVISIONS

Revision Number	Date	Reasons for Revision	Revised By	Verified By	Coord	APPROVAL SQ	APPROVAL EQ	APPROVAL PE	APPROVAL MGR
0	3/9/11	Initial Issue for Use	RAL	JWH	JPR	N/A	N/A	JWH	RLK
1	6/15/11	See Revision Summary	[illegible]	[illegible]	[illegible]	N/A	N/A	[illegible]	RLK

6/15/11

Revision Number	Affected Pages	Reason for Revision
0	All	Initial issue
1	Section 2.4.1	Revised to clarify Lead/Lag Filter equation.
	Section 2.6.1	Revised to clarify Rate/Lag Filter equation.
	Section 2.14.2	Added new section to define Thot Estimate Compensation Algorithm.

CONTENTS

1 INTRODUCTION 1

1.1 PURPOSE 1
1.2 SCOPE 1
1.3 ACRONYMS 1
1.4 REFERENCES 2

2 CONTROLLER TRANSFER FUNCTION REQUIREMENTS 3

2.1 OVERVIEW 3
2.2 TRANSFER FUNCTION: INPUT SCALING 3
2.3 TRANSFER FUNCTION: RTD RESISTANCE TO TEMPERATURE CONVERSION 4
2.4 TRANSFER FUNCTION: LEAD/LAG FILTER 4
2.5 TRANSFER FUNCTION: LAG FILTER 5
2.6 TRANSFER FUNCTION: RATE/LAG FILTER 6
2.7 TRANSFER FUNCTION: DTTA TAVG CALCULATION 7
2.8 TRANSFER FUNCTION: NORMALIZED POWER (P_B) CALCULATION 7
2.9 TRANSFER FUNCTION: DTTA DELTA-T CALCULATION 8
2.10 TRANSFER FUNCTION: OVERTEMPERATURE DELTA-T (OTDT) SETPOINT CALCULATION 8
2.11 TRANSFER FUNCTION: OVERPOWER DELTA-T (OPDT) SETPOINT CALCULATION 10
2.12 TRANSFER FUNCTION: SENSOR QUALITY ALGORITHM 2 (SQA2) 12
2.13 TRANSFER FUNCTION: SENSOR QUALITY ALGORITHM 3A AND 3B (SQA3A/SQA3B) 14
2.14 TRANSFER FUNCTION: T_{HOT} STREAMING FACTOR CALCULATION 26
2.15 TRANSFER FUNCTION: STEAMFLOW COMPENSATION 27
2.16 TRANSFER FUNCTION: STEAM GENERATOR LOW-LOW LEVEL TRIP TIME DELAY 28

1 Introduction

1.1 Purpose

The purpose of this Specification is to provide vendors with details necessary to develop the appropriate algorithms to implement the Controller Transfer Functions specified in the Process Protection System (PPS) Functional Requirements Specification (FRS) [Reference 1.4.1.1].

1.2 Scope

The PPS FRS [Reference 1.4.1.1] identifies the requirements that must be implemented by the PPS and is written in terms that are hardware independent. This specification supplements the PPS FRS by providing details for developing algorithms for use by a digital system to implement the PPS FRS specified controller transfer functions. All transfer functions specified in the PPS FRS (with the exception of bistable comparators) are included in this specification with the appropriate PPS FRS requirement section identified for traceability purposes.

1.3 Acronyms

1.3.1 Acronyms

ACRONYM	DEFINITION
CFR	Code of Federal Regulations
DCPP	Diablo Canyon Power Plant
DTTA	Delta-T / Tavg
FRS	Functional Requirements Specification
OPDT	Overpower Delta-T
OPSP	Overpower Setpoint
OPTR	Overpower Turbine Runback
OTDT	Overtemperature Delta-T
OTTR	Overtemperature Turbine Runback
PG&E (PGE)	Pacific Gas & Electric Company
PLS	Precautions, Limitations, and Setpoints (document)
PPS	Process Protection System
RCS	Reactor Coolant System

ACRONYM	DEFINITION
RTD	Resistance Temperature Detector
RTP	Rated Thermal Power
SQA	Sensor Quality Algorithm

1.4 References

1.4.1 Implementing Documents (Use Latest Revision)

1.4.1.1 DC 663195-44-1, DCPP Units 1 & 2, Process Protection System Functional Requirements Specification (Altran Solutions Corporation Document 08-0015-SP-001)

1.4.1.2 Technical Specifications, DCPP Units 1 and 2, Appendix A to License Nos. DPR-80 and DPR-82, as amended

1.4.1.3 DC 663229 – 47, Precautions Limits and Setpoints Document (PLS)

1.4.1.4 PG&E IDAP CF2.ID9, Software Quality Assurance Plan, Software Development

1.4.1.5 DCPP Maintenance Scaling Calculation (Unit 1) SC-I-36-M.2

1.4.1.6 DCPP Maintenance Scaling Calculation (Unit 2) SC-I-36-M.2

1.4.1.7 DCPP Maintenance Scaling Calculation (Unit 1) SC-I-36-M.5

1.4.1.8 DCPP Maintenance Scaling Calculation (Unit2) SC-I-36-M.5

2 Controller Transfer Function Requirements

2.1 Overview

The PPS FRS [Reference 1.4.1.1] specifies that controller transfer functions be provided to handle various control actions necessary to implement protection channel functions. The methods to be utilized to implement the specified controller transfer functions are included in this specification.

The methods defined in this section are acceptable. Any alternate method proposed by a particular vendor must be presented with proof of equivalence for acceptance for use by PG&E. Any such acceptance by PG&E will result in a revision to this specification to document the acceptability for use of the methodology proposed by the vendor.

Certain controller transfer functions that satisfy Technical Specification [Reference 1.4.1.2] requirements must/shall be implemented as shown in this specification. This requirement will be specifically identified where that function is described in this specification.

2.2 Transfer Function: Input Scaling

2.2.1 Input Scaling Implementation

Scaling shall be implemented as follows:

$$N = m * X + b$$

Where:

N	= input scaling value (engineering units)
M	= gain (constant)
X	= transmitter input (engineering units)
b	= offset (constant)

2.2.2 Input Scaling capability shall be provided for the following PPS functions:

a) All analog inputs [PPS FRS 3.2.1.13.2]
   i. m (gain) shall be set to one (1) unless specific scaling requirements are specified in the PPS FRS
   ii. b (offset) shall be set to zero (0) unless specific scaling requirements are specified in the PPS FRS

b) Reactor Coolant Flow [PPS FRS 3.2.2.13.1]
   i. Tuning constant ranges: [PPS FRS 3.2.1.14.3]
   ii. Information: gain (m) and offset (b) values are determined per Scaling Calc SC-I-36-M.2 [1.4.1.5, 1.4.1.6]

c) Steamflow [PPS FRS 3.2.9.13.2]
   i. Tuning constant ranges: [PPS FRS 3.2.1.14.3]
   ii. Information: gain (m) and offset (b) values are determined per Scaling Calc SC-I-36-M.5 [1.4.1.7, 1.4.1.8]

2.3 Transfer Function: RTD Resistance to Temperature Conversion

2.3.1 RTD Resistance to Temperature Conversion Implementation

Resistance to temperature conversion shall be implemented as follows:

$$RS(T) = a + bT + cT^2$$

Where:

RS(T) = resistance in ohms (actual measured reading)

T = temperature in degrees F

a, b, c = RTD constants from the RTD calibration curve (manually input for each RTD)

Hence:

$$T = \frac{-b + \sqrt{b^2 - 4c(a - RS(T))}}{2c}$$

2.3.2 Resistance to temperature conversion shall be provided for the following PPS functions:

a) Wide Range Reactor Coolant Temperature [PPS FRS 3.2.3.13.1]
   i. Tuning constant ranges: a, b, c values per individual RTD calibration curve.

b) DTTA (Narrow Range Hot and Cold Leg Temperatures) [PPS FRS 3.2.5.13.5]
   i. Tuning constant ranges: a, b, c values per individual RTD calibration curve.

c) Pressurizer Vapor Temperature [PPS FRS 3.2.8.13.1]
   i. Tuning constant ranges: a, b, c values per individual RTD calibration curve.

2.4 Transfer Function: Lead/Lag Filter

2.4.1 Lead/Lag Filter Implementation

Lead/Lag filters shall be implemented as follows:

Reference equation:

$$Y(n) = C1 * X(n) + C2 * X(n-1) + C3 * Y(n-1)$$

Where:

Y(n) = present output value (engineering units)

X(n) = present input value (engineering units)

X(n-1) = previous cycle input value (engineering units)

Y(n-1) = previous cycle ouput value (engineering units)

$$\text{Coefficient } C1 = \left(\frac{G * 2 * \tau_n + T}{2 * \tau_d + T}\right)$$

$$\text{Coefficient } C2 = -\left(\frac{G * 2 * \tau_n - T}{2 * \tau_d + T}\right)$$

$$\text{Coefficient } C3 = \left(\frac{2 * \tau_d - T}{2 * \tau_d + T}\right)$$

Where:

G = Gain (equal to 1 unless otherwise specified)

τ_n = user entered lead time constant (seconds)

τ_d = user entered lag time constant (seconds)

T = cycle time in seconds

To provide a unity transfer function (output = input), set lead time constant (τ_n) and lag time constant (τ_d) equal to 0.0.

2.4.2 Lead/Lag Filters shall be provided for the following functions:

a) DTTA Tavg [PPS FRS 3.2.5.13.2]
   - i. $Tavg_{LEAD/LAG}$ = Y (Lead/Lag filter output value per Section 2.4.1)
   - ii. $\left(T_{avg} - T^{\circ}_{avg}\right)$ = X (Lead/Lag filter input per Section 2.4.1)
   - iii. Tuning constant ranges: [PPS FRS 3.2.5.14.7].

b) DTTA Delta-T [PPS FRS 3.2.5.13.2]
   - i. $\Delta T_{LEAD/LAG}$ = Y (Lead/Lag filter output value per Section 2.4.1)
   - ii. Calculated ΔT = X (Lead/Lag filter input value per Section 2.4.1)
   - iii. Tuning constant ranges: [PPS FRS 3.2.5.14.7].

c) Pressurizer Pressure reactor trip compensation [PPS FRS 3.2.7.13.1]
   - i. Tuning constant ranges: [PPS FRS 3.2.7.14.6].

d) Steamline Pressure [PPS FRS 3.2.10.13.1]
   - i. Tuning constant ranges: [PPS FRS 3.2.10.14.4]

2.5 Transfer Function: Lag Filter

2.5.1 Lag Filter Implementation

The Lag Filter shall be implemented using the same format as the Lead/Lag Filter described in Section 2.4 with the lead time constant (τ_n) set to 0.0.

2.5.2 Lag filters shall be provided for the following PPS functions:

The following functions require Lag Filters for other than process noise filtering purposes:

a) Low pass (Lag) filtering capability shall be provided for all analog inputs [PPS FRS 3.2.1.12.2]

   i. Lag time constant shall be tunable as specified in the referenced PPS FRS section.

   ii. Lag filter (time constant) shall be adjusted as necessary to provide attenuation of process noise.

   iii. Lag time constant shall be set to 0.0 if not required for process noise attenuation.

   iv. Adjustment of the Lag time time constant will be per administrative procedure.

b) DTTA Narrow Range Tcold [PPS FRS 3.2.5.13.1]

   i. Each DTTA Tcold input shall be provided with a Lag Filter.

   ii. Tuning constant ranges: [PPS FRS 3.2.5.14.7]

c) DTTA Narrow Range Thot [PPS FRS 3.2.5.13.1]

   i. Each DTTA Thot input shall be provided with a Lag Filter.

   ii. Tuning constant ranges: [PPS FRS 3.2.5.14.7]

d) Calculated Thot Streaming Factor [PPS FRS 3.2.5.13.1]

   i. Each calculated Thot streaming factor output shall be provided with a Lag Filter.

   ii. Tuning constant ranges: [PPS FRS 3.2.5.14.7]

2.6 Transfer Function: Rate/Lag Filter

2.6.1 Rate/Lag Filter shall be implemented as follows:

Reference equation:

$$Y(n) = C1 * (X(n) - X(n-1)) + C2 * Y(n-1)$$

Where:

$Y(n)$ = present output value (engineering units)

$X(n)$ = present input value (engineering units)

$X(n-1)$ = previous cycle input value (engineering units)

$Y(n-1)$ = previous cycle output value (engineering units)

$$\text{Coefficient } C1 = \left(\frac{G * 2 * \tau_n}{2 * \tau_d + T}\right)$$

$$\text{Coefficient } C2 = \left(\frac{2 * \tau_d - T}{2 * \tau_d + T}\right)$$

Where:

- G = Gain (equal to 1 unless otherwise specified)
- τ_n = user entered rate time constant (seconds)
- τ_d = user entered lag time constant (seconds)
- T = cycle time in seconds

2.6.2 Rate/Lag filters shall be provided for the following functions:

a) Tavg input to DTTA OPSP calculation [PPS FRS 3.2.5.13.3]
   i. Calculated Tavg = X (Rate/Lag filter input value per Section 2.6.1).
   ii. Tuning constant ranges: [PPS FRS 3.2.5.14.7]

b) Steamline Pressure [PPS FRS 3.2.10.13.2]
   i. Gain for steamline pressure Rate/Lag shall be = -1
   ii. Tuning constant ranges: [PPS FRS 3.2.10.14.4]

2.7 Transfer Function: DTTA Tavg Calculation

2.7.1 DTTA Tavg Calculation Algorithm [PPS FRS 3.2.5.13.4]

DTTA Tavg shall be calculated as follows:

$$T_{avg} = \frac{T^f{}_{havg} + T^f{}_{cavg}}{2.0}$$

Where:

- Tavg = calculated loop average temperature (°F)
- T^fhavg = calculated loop average hot leg temperature (°F)
- T^fcavg = calculated loop average cold leg temperature (°F)

Note: T^fhavg and T^fcavg values are determined by the SQA3A(B) [Section 2.13] and SQA2 [Section 2.12] algorithms.

2.8 Transfer Function: Normalized Power (P_B) Calculation

2.8.1 Normalized Power calculation algorithm [PPS FRS 3.2.5.13.11]:

Normalized Power shall be calculated as follows:

$$P_B = \frac{T^f_{havg} - T^f_{cavg}}{\Delta T^\circ}$$

Where:

P_B = normalized power (unitless)

$T^f havg$ = calculated loop average hot leg temperature (°F)

$T^f cavg$ = calculated loop average cold leg temperature (°F)

ΔT° = user entered tuning constant representing the loop specific DT at rated thermal power (°F)

P_B = 0.0 when calculated PB ≤ 0.0

P_B = 1.5 when calculated PB ≥ 1.5

Note: $T^f havg$ and $T^f cavg$ values are determined by the SQA3A(B) [Section 2.13] and SQA2 [Section 2.12] algorithms.

2.9 Transfer Function: DTTA Delta-T Calculation

2.9.1 DTTA Delta-T Calculation Algorithm [PPS FRS 3.2.5.13.4]

DTTA Delta-T shall be calculated as follows:

$$\Delta T = P_B * 100$$

Where:

ΔT = reactor power equivalent of loop differential temperature (equivalent reactor power units)

P_B = normalized power (unitless)

2.10 Transfer Function: Overtemperature Delta-T (OTDT) Setpoint Calculation

2.10.1 DTTA OTDT Setpoint Calculation Algorithm [PPS FRS 3.2.5.13.6]

Note: Tech Spec requirement – no change to algorithm allowed.

DTTA OTDT Setpoint shall be calculated as follows:

$$OTDT_{setpoint} = \Delta T^\circ * \left[K_1 - K_2 * T_{avgLEAD/LAG} + K_3 * (P - P^\circ) - f_1(\Delta I)\right]$$

Where:

ΔT° = loop specific indicated ΔT at rated thermal power (expressed in equivalent reactor power units)

T_{avg} = measured Tavg signal (°F)

T°_{avg} = nominal Tavg at rated thermal power (°F)

P = pressurizer pressure (psig)

P^o = nominal RCS operating pressure (psig)

s = Laplace transform operator (sec^{-1})

K_1 = user entered tuning constant (unitless)

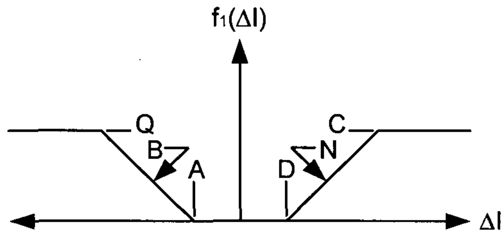
K_2 = user entered tuning constant (/°F)

K_3 = user entered tuning constant (/psig)

$f_1(\Delta I)$ = flux imbalance as shown below (% of rated thermal power)

$Tavg_{LEAD/LAG}$ = see Section 2.4.2

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].



Where:

- ΔI = the difference between the upper and lower calibrated ion chamber current readings
- A = breakpoint (user entered tuning constant)
- B = slope (user entered tuning constant)
- Q = limit (user entered tuning constant)
- D = breakpoint (user entered tuning constant)
- N = slope (user entered tuning constant)
- C = limit (user entered tuning constant)

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.10.2 The input to the OTDT temperature comparator shall be:

$$\Delta T_{LEAD/LAG} - OTDT_{setpoint}$$

$\Delta T_{LEAD/LAG}$ = see Section 2.4.2

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.10.3 An OTDT Reactor Trip shall occur when:

$$\Delta T_{LEAD/LAG} \succ OTDT_{setpoint}$$

$\Delta T_{LEAD/LAG}$ = see Section 2.4.2

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.10.4 An OTDT Turbine Runback shall occur when:

$$\Delta T_{LEAD/LAG} - OTDT_{setpoint} \succ OTTR_{setpoint}$$

$\Delta T_{LEAD/LAG}$ = see Section 2.4.2

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.11 Transfer Function: Overpower Delta-T (OPDT) Setpoint Calculation

2.11.1 DTTA OPDT Setpoint Calculation Algorithm [PPS FRS 3.2.5.13.7]

Note: Tech Spec requirement – no change to algorithm allowed.

DTTA OPDT Setpoint shall be calculated as follows:

$$OPDT_{setpoint} = \Delta T^{\circ} * \left[K_4 - K_5 * T_{avgRATE/LAG} - K_6 * \left(T_{avg} - T'_{avg}\right) - f_2(\Delta I)\right]$$

Where:

ΔT^{o} = loop specific indicated ΔT at rated thermal power (expressed in equivalent reactor power units)

T_{avg} = measured Tavg signal (°F)

T'_{avg} = nominal loop specific indicated Tavg at rated thermal power (°F)

s = Laplace transform operator (sec-1)

K_4 = user entered tuning constant (unitless)

K_5 = user entered tuning constant (/°F)

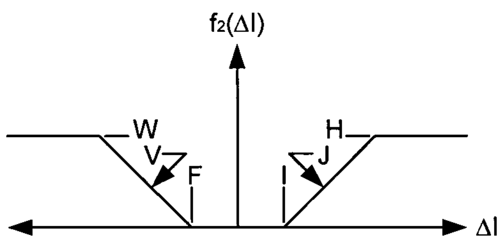
K_6 = user entered tuning constant (/°F)

$f_2(\Delta I)$ = flux imbalance as shown below (% of rated thermal power)

$Tavg_{RATE/LAG}$ = see Section 2.6.2

Note: $f_2(\Delta I)$ shall be 0% of rated thermal power for all ΔI at DCPP.

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].



Where:

ΔI = the difference between the upper and lower calibrated ion chamber current readings

F = breakpoint (user entered tuning constant)

V = slope (user entered tuning constant)

W = limit (user entered tuning constant)

I = breakpoint (user entered tuning constant)

J = slope (user entered tuning constant)

H = limit (user entered tuning constant)

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.11.2 The input to the OPDT temperature comparator shall be:

$$\Delta T_{LEAD/LAG} - OPDT_{setpoint}$$

$\Delta T_{LEAD/LAG}$ = see Section 2.4.2

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.11.3 An OPDT Reactor Trip shall occur when:

$$\Delta T_{LEAD/LAG} \succ OPDT_{setpoint}$$

$\Delta T_{LEAD/LAG}$ = see Section 2.4.2

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.11.4 An OPDT Turbine Runback shall occur when:

$$\Delta T_{LEAD/LAG} - OPDT_{setpoint} \succ OPTR_{setpoint}$$

$\Delta T_{LEAD/LAG}$ = see Section 2.4.2

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.12 Transfer Function: Sensor Quality Algorithm 2 (SQA2)

2.12.1 SQA2 Algorithm [PPS FRS 3.2.5.13.8]

The SQA2 algorithm is performed to determine the average temperature of a DTTA loop cold leg channel (T^fcavg).

The SQA2 Algorithm shall be implemented per Steps “a” and “b” as follows:

a) Determine the SQA2 Case and perform the associated “Action” per the following Truth Table:

SQA2 Algorithm Case Determination Truth Table

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
2 Good Inputs (pass diagnostics and not RFS)						
1	T^f_{c1}	X			X	Perform Consistency Check 1
	T^f_{c2}	X			X	
1 Good Input (pass diagnostics and not RFS)						
2	T^f_{c1}	X			X	T^fcavg = T^f_{c1}
	T^f_{c2}	X		X		
3	T^f_{c1}	X			X	T^fcavg = T^f_{c1}
	T^f_{c2}		X		X	Alarms = TRBL
4	T^f_{c1}	X			X	T^fcavg = T^f_{c1}

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	T^f_{c2}		X	X		
5	T^f_{c1}	X		X		$T^f cavg = T^f_{c2}$
	T^f_{c2}	X			X	
6	T^f_{c1}		X		X	$T^f cavg = T^f_{c2}$
	T^f_{c2}	X			X	Alarms = TRBL
7	T^f_{c1}		X	X		$T^f cavg = T^f_{c2}$
	T^f_{c2}	X			X	
No Good Inputs (fail diagnostics and/or RFS)						
8	T^f_{c1}	X		X		$T^f cavg = T^f_{c1}$ or T^f_{c2} (see Note 1 at end of Table)
	T^f_{c2}	X		X		Alarms = RTD FAILURE (no Tc available)
9	T^f_{c1}	X		X		$T^f cavg = T^f_{c2}$
	T^f_{c2}		X		X	Alarms = RTD FAILURE (no Tc available)
10	T^f_{c1}	X		X		$T^f cavg = T^f_{c1}$ or T^f_{c2} (see Note 1 at end of Table)
	T^f_{c2}		X	X		Alarms = RTD FAILURE (no Tc available)
11	T^f_{c1}		X		X	$T^f cavg = T^f_{c1}$
	T^f_{c2}	X		X		Alarms = RTD FAILURE (no Tc available)
12	T^f_{c1}		X		X	$T^f cavg = (T^f_{c1} + T^f_{c2})/2$
	T^f_{c2}		X		X	Alarms = RTD FAILURE (no Tc available)
13	T^f_{c1}		X		X	$T^f cavg = T^f_{c1}$
	T^f_{c2}		X	X		Alarms = RTD FAILURE (no Tc available)
14	T^f_{c1}		X	X		$T^f cavg = T^f_{c1}$ or T^f_{c2} (see Note 1 at end of Table)
	T^f_{c2}	X		X		Alarms = RTD FAILURE (no Tc available)

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
15	T^f_{c1}		X	X		$T^f cavg = T^f_{c2}$
	T^f_{c2}		X		X	Alarms = RTD FAILURE (no Tc available)
16	T^f_{c1}		X	X		$T^f cavg = T^f_{c1}$ or T^f_{c2} (see Note 1 at end of Table)
	T^f_{c2}		X	X		Alarms = RTD FAILURE (no Tc available)

Where:

$T^f cavg$ = average cold leg temp (°F)

T^f_{c1} = filtered cold leg temperature from thermowell RTD "1" (°F)

T^f_{c2} = filtered cold leg temperature from thermowell RTD "2" (°F)

Note 1: When administratively removed from service, T^f_{c1} or T^f_{c2} shall retain its last value while in service. The value of $T^f cavg$ will be from the last $T^f c$ (1 or 2) removed from service when both are RFS.

b) Perform the following SQA2 DELTAC Consistency Checks as directed by SQA2 Case Determination Truth Table:

Consistency Check 1 Truth Table

Condition	≤ DELTAC	> DELTAC	Action
$\lvert T^f_{c1}-T^f_{c2}\rvert$	X		$T^f cavg = (T^f_{c1} + T^f_{c2})/2$
$\lvert T^f_{c1}-T^f_{c2}\rvert$		X	$T^f cavg = (T^f_{c1} + T^f_{c2})/2$, Alarms = RTD FAILURE

Where:

DELTAC = user entered tuning constant (°F)

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

2.13 Transfer Function: Sensor Quality Algorithm 3A and 3B (SQA3A/SQA3B)

2.13.1 SQA3A/SQA3B Algorithms [PPS FRS 3.2.5.13.9]

The SQA3A and SQA3B algorithms are performed to determine the average temperature of a DTTA loop hot leg channel ($T^f havg$). Each determines a value and the two values (T^f_{havgA} and T^f_{havgB}) are combined to determine the $T^f havg$ for the DTTA channel.

The SQA3A algorithm shall be used to determine the average hot leg temperature for the DTTA channel as developed from the three (3) "A" Thot RTDS (one per thermowell).

The SQA3B algorithm shall be used to determine the average hot leg temperature for the DTTA channel as developed from the three (3) "B" Thot RTDS (one per thermowell).

The SQA3 and SQA3B Algorithms shall be implemented per Steps "a" thru "e" as follows:

a) Determine the SQA3 Case and perform the associated "Action" per the following

Truth Table:

SQA3 Algorithm Case Determination Truth Table (Typical of SQA3A or SQA3B)

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
3 Good Inputs (pass diagnostics and not RFS)						
1	Th1est	X			X	Perform Consistency Check 1
	Th2est	X			X	
	Th3est	X			X	
2 Good Inputs (pass diagnostics and not RFS)						
2	Th1est	X			X	Perform Consistency Check 2 (Case 1)
	Th2est	X			X	
	Th3est	X		X		
3	Th1est	X			X	Perform Consistency Check 2 (Case 1) Alarms = TRBL (1 Bad Input)
	Th2est	X			X	
	Th3est		X		X	
4	Th1est	X			X	Perform Consistency Check 2 (Case 1)
	Th2est	X			X	
	Th3est		X	X		
5	Th1est	X			X	Perform Consistency Check 2 (Case 2)
	Th2est	X		X		
	Th3est	X			X	
6	Th1est	X			X	Perform Consistency Check 2 (Case 2) Alarms = TRBL (1 Bad Input)
	Th2est		X		X	
	Th3est	X			X	
7	Th1est	X			X	Perform Consistency Check 2 (Case 2)
	Th2est		X	X		
	Th3est	X			X	
8	Th1est	X		X		Perform Consistency Check 2 (Case 3)

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	Th2est	X			X	
	Th3est	X			X	
9	Th1est		X		X	Perform Consistency Check 2 (Case 3)
	Th2est	X			X	Alarms = TRBL (1 Bad Input)
	Th3est	X			X	
10	Th1est		X	X		Perform Consistency Check 2 (Case 3)
	Th2est	X			X	
	Th3est	X			X	
1 Good Input (pass diagnostics and not RFS)						
11	Th1est	X			X	T^fhavg = Th1est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
12	Th1est	X			X	T^fhavg = Th1est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
13	Th1est	X			X	T^fhavg = Th1est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
14	Th1est	X			X	T^fhavg = Th1est
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
15	Th1est	X			X	T^fhavg = Th1est
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
16	Th1est	X			X	T^fhavg = Th1est

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
17	Th1est	X			X	T'havg = Th1est
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
18	Th1est	X			X	T'havg = Th1est
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
19	Th1est	X			X	T'havg = Th1est
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
20	Th1est	X		X		T'havg = Th2est
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
21	Th1est	X		X		T'havg = Th2est
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
22	Th1est	X		X		T'havg = Th2est
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
23	Th1est		X		X	T'havg = Th2est
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
24	Th1est		X	X		T'havg = Th2est

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
25	Th1est		X	X		T^fhavg = Th2est
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
26	Th1est		X	X		T^fhavg = Th2est
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
27	Th1est		X		X	T^fhavg = Th2est
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
28	Th1est		X		X	T^fhavg = Th2est
	Th2est	X			X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
29	Th1est	X		X		T^fhavg = Th3est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
30	Th1est	X		X		T^fhavg = Th3est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
31	Th1est	X		X		T^fhavg = Th3est
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
32	Th1est		X		X	T^fhavg = Th3est

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
33	Th1est		X	X		T'havg = Th3est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
34	Th1est		X	X		T'havg = Th3est
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
35	Th1est		X	X		T'havg = Th3est
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
36	Th1est		X		X	T'havg = Th3est
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
37	Th1est		X		X	T'havg = Th3est
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X			X	
No Good Inputs (fail diagnostics and/or RFS)						
38	Th1est	X		X		T'havg = Th1est, Th2est, or Th3est (see Note 2 at end of Table)
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
39	Th1est		X	X		T'havg = Value of Last Thest RFS
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
40	Th1est		X		X	T'havg = (Th1est + Th2est + Th3est)/3

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
41	Th1est	X		X		T'havg = Th1est, Th2est, or Th3est (see Note 2 at end of Table)
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
42	Th1est	X		X		T'havg = Th3est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
43	Th1est	X		X		T'havg = Th1est, Th2est, or Th3est (see Note 2 at end of Table)
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
44	Th1est	X		X		T'havg = Th2est
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
45	Th1est		X	X		T'havg = Th1est, Th2est, or Th3est (see Note 2 at end of Table)
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
46	Th1est		X		X	T'havg = Th1est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
47	Th1est	X		X		T'havg = Th1est, Th2est, or Th3est (see Note 2 at end of Table)
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
48	Th1est	X		X		T'havg = Th3est

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
49	Th1est	X		X		T'havg = Th2est
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
50	Th1est	X		X		T'havg = (Th2est + Th3est)/2
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
51	Th1est		X	X		T'havg = Th1est, Th2est, or Th3est (see Note 2 at end of Table)
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
52	Th1est		X	X		T'havg = Th3est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
53	Th1est		X		X	T'havg = Th1est
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
54	Th1est		X		X	T'havg = (Th1est + Th3est)/2
	Th2est	X		X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
55	Th1est		X	X		T'havg = Th1est, Th2est, or Th3est (see Note 2 at end of Table)
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
56	Th1est		X	X		T'havg = Th2est

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
57	Th1est		X		X	T'havg = Th1est
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
58	Th1est		X		X	T'havg = (Th1est + Th2est)/2
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est	X		X		
59	Th1est		X	X		T'havg = Th3est
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
60	Th1est		X	X		T'havg = Th2est
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
61	Th1est		X		X	T'havg = Th1est
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		
62	Th1est		X	X		T'havg = (Th2est + Th3est)/2
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
63	Th1est		X		X	T'havg = (Th1est + Th3est)/2
	Th2est		X	X		Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X		X	
64	Th1est		X		X	T'havg = (Th1est + Th2est)/2

Case	Input	Diag - Pass	Diag - Fail	RFS - Yes	RFS - No	Action
	Th2est		X		X	Alarms = TRBL (<2 Good inputs this SQA3)
	Th3est		X	X		

Where:

$T^f havg$ = average hot leg temp for SQA3 (A or B) (°F)

Th1est = filtered Thot (T^f_{h1}) corrected for hot leg streaming (°F)

Th2est = filtered Thot (T^f_{h2}) corrected for hot leg streaming (°F)

Th3est = filtered Thot (T^f_{h3}) corrected for hot leg streaming (°F)

Note 1: These values are representative of SQA3A or SQA3B and tagnames would have "A" or "B" appended.

Note 2: When administratively removed from service, Th1est, Th2est, or Th3est shall retain its last value while in service. The value of $T^f havg$ will be from the last Thest (1, 2, or 3) removed from service when all are RFS.

b) Perform the following SQA3 DELTAH Consistency Checks as directed by SQA3 Case Determination Truth Table:

Consistency Check 1 Truth Table

Condition	Δ1	Δ2	Δ3	None	Action
> DELTAH				X	$T^f havg$ = (Th1est + Th2est + Th3est)/3
> DELTAH			X		Perform Consistency Check 2 (Case 1)
> DELTAH		X			Perform Consistency Check 2 (Case2)
> DELTAH	X				Perform Consistency Check 2 (Case 3)
> DELTAH		X	X		Perform Delta Comparison Check 1
> DELTAH	X		X		Perform Delta Comparison Check 2
> DELTAH	X	X			Perform Delta Comparison Check 3
> DELTAH	X	X	X		Perform Delta Comparison Check 4

Where:

DELTAH = user entered tuning constant (°F)

$Thest_{avg}$ = (Th1est + Th2est + Th3est)/3 (°F)

Δ1 = $|Thest_{avg} - Th1est|$ (°F)

Δ2 = $|Thest_{avg} - Th2est|$ (°F)

Δ3 = $|Thest_{avg} - Th3est|$ (°F)

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

c) Perform the following SQA3 DELTAH Consistency Checks as directed by the SQA3 Case Determination Truth Table or the Delta Comparison Check Truth Tables:

Consistency Check 2 Truth Table

Case	Condition	≤ DELTAH	> DELTAH	Action
1	|Th1est – Th2est|	X		T'havg = (Th1est + Th2est)/2
	|Th1est – Th2est|		X	T'havg = (Th1est + Th2est)/2, Alarms = TRBL
2	|Th1est – Th3est|	X		T'havg = (Th1est + Th3est)/2
	|Th1est – Th3est|		X	T'havg = (Th1est + Th3est)/2, Alarms = TRBL
3	|Th2est – Th3est|	X		T'havg = (Th2est + Th3est)/2
	|Th2est – Th3est|		X	T'havg = (Th2est + Th3est)/2, Alarms = TRBL

Where:

DELTAH = user entered tuning constant (^{o}F)

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

d) Perform the following SQA3 Delta Comparison Checks as directed by the SQA3 Consistency Check 1 Truth Table:

Delta Comparison Check 1 Truth Table

Condition	Δ2 > Δ3	Δ3 > Δ2	Δ2 = Δ3	Action
Compare: Δ2, Δ3	X			Perform Consistency Check 2 (Case 2)
Compare: Δ2, Δ3		X		Perform Consistency Check 2 (Case 1)
Compare: Δ2, Δ3			X	Perform Consistency Check 2 (Case 3)

Delta Comparison Check 2 Truth Table

Condition	Δ1 > Δ3	Δ3 > Δ1	Δ1 = Δ3	Action
Compare: Δ1, Δ3	X			Perform Consistency Check 2 (Case 3)
Compare: Δ1, Δ3		X		Perform Consistency Check 2 (Case 1)
Compare: Δ1, Δ3			X	Perform Consistency Check 2 (Case 2)

Delta Comparison Check 3 Truth Table

Condition	Δ1 > Δ2	Δ2 > Δ1	Δ1 = Δ2	Action
Compare: Δ1, Δ2	X			Perform Consistency Check 2 (Case 3)

Condition	Δ1 > Δ2	Δ2 > Δ1	Δ1 = Δ2	Action
Compare: Δ1, Δ2		X		Perform Consistency Check 2 (Case 2)
Compare: Δ1, Δ2			X	Perform Consistency Check 2 (Case 1)

Delta Comparison Check 4 Truth Table

Condition	Δ1 > Δ2, Δ3	Δ2 > Δ1, Δ3	Δ3 > Δ1, Δ2	Δ1 = Δ2 = Δ3	Action
Compare: Δ1, Δ2, Δ3	X				Perform Consistency Check 2 (Case 3)
Compare: Δ1, Δ2, Δ3		X			Perform Consistency Check 2 (Case 2)
Compare: Δ1, Δ2, Δ3			X		Perform Consistency Check 2 (Case 1)
Compare: Δ1, Δ2, Δ3				X	Tfhavg = (Th1est + Th2est + Th3est)/3, Alarms = TRBL

e) Determine the Loop T^fhavg (from SQA3A and SQA3B results) by performing the following Truth Table (applicable) Actions after completion of Steps "a" thru "d" above for each SQA3:

T^fhavg Calculation Truth Table (from SQA3A and SQA3B results)

Condition	T^f_{havgA} SQA3A	T^f_{havgB} SQA3B	Neither	TRBL Alarm (1 Bad Input) SQA3A	TRBL Alarm (1 Bad Input) SQA3B	Action
TRBL Alarm (< 2 Good Inputs)			X			T^fhavg = $(T^f_{havgA} + T^f_{havgB})/2$
TRBL Alarm (< 2 Good Inputs)			X	X		T^fhavg = $(T^f_{havgA} + T^f_{havgB})/2$
TRBL Alarm (< 2 Good Inputs)			X		X	T^fhavg = $(T^f_{havgA} + T^f_{havgB})/2$
TRBL Alarm (< 2 Good Inputs)			X	X	X	T^fhavg = $(T^f_{havgA} + T^f_{havgB})/2$
TRBL Alarm (< 2 Good Inputs)		X				T^fhavg = T^f_{havgA}
TRBL Alarm (< 2 Good Inputs)		X		X		T^fhavg = T^f_{havgA}

Condition	T^f_{havgA} SQA3A	T^f_{havgB} SQA3B	Neither	TRBL Alarm (1 Bad Input) SQA3A	TRBL Alarm (1 Bad Input) SQA3B	Action
TRBL Alarm (< 2 Good Inputs)	X					$T^f havg = T^f_{havgB}$
TRBL Alarm (< 2 Good Inputs)	X				X	$T^f havg = T^f_{havgB}$
TRBL Alarm (< 2 Good inputs)	X	X				$T^f havg = (T^f_{havgA} + T^f_{havgB})/2$ Alarms = RTD FAILURE (< 2 Good Inputs SQA3A & SQA3B)

2.14 Transfer Function: T_{hot} Streaming Factor Calculation

2.14.1 Thot Streaming Factor Calculation Algorithm [PPS FRS 3.2.5.13.10]

The Thot streaming factor shall be determined per the following truth table:

Th Streaming Factor Determination Truth Table

Condition	T^f_{hjA} Good (Pass Diagnostics and Not RFS)	T^f_{hjB} Good (Pass Diagnostics and Not RFS)	Action
$P_B \geq P_{LOW}$	X	X	$S_j = [((T^f_{hjA} + T^f_{hjB}) / 2) - T^f_{havg}) / P_B]$
$P_B \geq P_{LOW}$	X		$S_j = [(Tf_{hjA} - T^f_{havg}) / P_B]$
$P_B \geq P_{LOW}$		X	$S_j = [(Tf_{hjB} - T^f_{havg}) / P_B]$
$P_B \geq P_{LOW}$			S_j = Value from previous scan
$P_B < P_{LOW}$	X	X	S_j = Value from previous scan
$P_B < P_{LOW}$	X		S_j = Value from previous scan
$P_B < P_{LOW}$		X	S_j = Value from previous scan
$P_B < P_{LOW}$			S_j = Value from previous scan

Where:

P_B = normalized power (unitless)

P_{LOW} = lower threshold value for P_B (user entered tuning constant)

T^f_{havg} = calculated average Thot in the DTTA loop (°F)

T^f_{hjA} = value of the "A" RTD in the j^{th} thermowell (°F)

T^f_{hjB} = value of the "B" RTD in the j^{th} thermowell (°F)

j = loop T_h thermowell (1, 2, or 3)

S_j = calculated streaming factor value for the j^{th} thermowell RTDs (°F)

Ranges for tuning constants: [PPS FRS 3.2.5.14.7].

S_j shall be available for updating S^o_j tuning constant. S^o_j tuning constant is used to adjust the T_{hest} value to be used in the SQA3a or SQA3B algorithm for the jth thermowell RTDs.

2.14.2 Thot Estimate Compensation Algorithm [PPS FRS 3.2.5.13.10]

The Thot streaming factor calculated for each Thot input shall be applied as a compensation factor in determining the Thot Estimate value for each Thot input as follows:

$$Thjest = Thj^f - P_B * Sj$$

Where:

Thjest = the jth filtered Th signal compensated for Thot streaming

Thjf = the jth filtered Th signal (j = 1 to 6)

P_B = normalized power

Sj = calculated streaming factor for the jth filtered Th signal

The Thot streaming factors shall be calculated in each loop cycle but shall require user action to update the streaming factors used by the Thot Estimate algorithms.

2.15 Transfer Function: Steamflow Compensation

2.15.1 Steam Density Calculation Algorithm:

SteamDensity = A * (Steam Pressure in psig) + B

Note: The steam density calculation is a best fit linearization of the steam density vs. pressure function.

Where:

A = steamflow tuning constant (user entered)

B = steamflow tuning constant (user entered)

Ranges for tuning constants: [PPS FRS 3.2.9.14.1].

2.15.2 Non-compensated Steamflow Calculation Algorithm:

$$SF = \frac{(SteamDensity) * (SFDP - SFDP_{min})}{(SteamDensity_{ref}) * (SFDP_{max} - SFDP_{min})} \text{ for } SF \geq SF_{min}$$

Where:

SF = non-compensated steamflow

SF = 0 for SF < SF_{min}

SF_{min} = user entered tuning constant

SFDP = steamflow transmitter DP signal (% of full load DP)

max = operator adjustable maximum value of SFC and SFDP signal ranges

min = operator adjustable minimum value of SFC and SFDP signal ranges

$SteamDensity_{ref}$ = user entered tuning constant derived from:

"A * (Rated Steam Pressure @ Full Load) + B"

(A and B from Section 2.15.1)

Ranges for tuning constants: [PPS FRS 3.2.9.14.1].

2.15.3 Steamflow Compensation Algorithm [PPS FRS 3.2.9.13.1]

The Steamflow Compensation Algorithm shall be implemented as follows:

$$SFC = (SFC_{max} - SFC_{min}) * (SF)^{1/2} + SFC_{min}$$

Where:

SFC = compensated steamflow (million pounds per hour)

2.16 Transfer Function: Steam Generator Low-Low Level Trip Time Delay

2.16.1 Steam Generator Low-Low Level Trip Time Delay Algorithm [PPS FRS 3.2.11.13.3]

Note: Tech Spec requirement – no change to algorithm allowed.

The Steam Generator Low-Low Level Trip Time Delay Algorithm shall be implemented as follows:

$$TD = A(PL)^3 + B(PL)^2 + C(PL) + D$$

Where:

TD = allowable time delay (seconds) with PL ≤ 50% RTP

TD = 0 with PL > 50% RTP

PL = RCS Loop ΔT Equivalent to power (% RTP)

A = constant (unitless)

B = constant (unitless)

C = constant (unitless)

D = constant (unitless)

Ranges for tuning constants: [PPS FRS 3.2.11.14.3].

Note: the formula shown is functionally equivalent to the format as presented in the Technical specification.