NSTX CENTER STACK UPGRADE

Digital Coil Protection System
Algorithm Requirements Document

NSTX-SRD-13-170, Revision 0

October 27, 2012

Prepared By: ________________________________

Ronald E. Hatcher

Reviewed By: ________________________________

Charles Neumeyer

Reviewed By: ________________________________

Paul Sichta

Approved By: ________________________________

Alfred von Halle

NSTX_CSU DCPS SOFTWARE REQUIREMENTS DOCUMENT

RECORD OF CHANGES

REVISION	DATE	DESCRIPTION
0		First Issue

NSTX_CSU DCPS SOFTWARE REQUIREMENTS DOCUMENT

TABLE OF CONTENTS

1 Table of Contents

2 REFERENCES

1. General Requirements Document – NSTX_CSU-RQMTS-GRD, Rev. 4
2. Coil Protection System Requirements Document – NSTX_CSU-RQMT-CPS-159, Rev. 1
3. Digital Coil Protection System Software Requirements Document – NSTX-SRD-13-163--1

3 INTRODUCTION

The NSTX Center Stack Upgrade experimental device (NSTX-U) has an operating space that is both larger and more complex than that of the original NSTX. The mechanical integrity of some components can be compromised both by the instantaneous values of combinations of magnet currents and resulting magnetic fields and as a consequence of the time histories thereof. An upgrade to the existing protection systems and methodology is required to allow for both safe and effective use of the expanded operating space. Per the Coil Protection System (CPS) requirements document [2], a Digital Coil Protection System (DCPS) is a required subsystem of the CPS.

At its core, the DCPS consists of one or more computer systems executing protection algorithms within the context of a sampled-data system. The protection algorithms are used to calculate approximations to one or more physical quantities (forces, stresses, temperatures, moments, etc.)

defined by the analysis as limit variables (LVs). Collectively, the LVs define a safe operating envelope for the NSTX-U device. The algorithms are executed (in double precision) at each time step and their output compared to limiting values. In addition, the DCPS uses predicted future values of the LVs to provide protection in a "first derivative" sense. A software solution is chosen to allow for maximum flexibility and for ease maintenance, modification, and expansion.

4 DATA CALIBRATION

The analog "data" inputs to the DCPS system are digitized representations of NSTX-U field coil and plasma currents. The data are sampled at a rate of 5 kHz (Δt_{samp} = 0.2 ms) and typically span 0 – 10 VDC for uni-polar signals or -10 – 10 VDC for bi-polar signals. To convert (scale) the input data to engineering units suitable for algorithm calculation, the DCPS system will use a calibration factor table. The entries of the calibration factor table are multipliers that when applied to the input data convert them to DCPS internal units (e.g., a 50,000 Ampere current transducer might require a 5000 A/V calibration factor). As other NSTX-U systems (e.g., PCS, PSRTC, etc.) require these same calibration factors, the DCPS should, where feasible, use existing facilities for generating, accessing, and maintaining data calibration factors.

Also, as part of calibration, each analog input data stream will require offset subtraction. A simple moving average of 100 samples taken prior to the start of a pulse will constitute the signal offset. The offset will be subtracted from all samples taken during the pulse prior to the application of the calibration factor. An error is generated if the offset exceeds (in magnitude) exceeds a pre-set limiting value.

5 ALGORITHM CLASSES

Most, if not all, of the quantities of interest to the DCPS are functions of the poloidal and toroidal field currents and the plasma current. In most cases the function primarily depends on a weighted sum of products of currents. This allows the many individual algorithms for the different physical quantities to be "re-cast" into a small number of standard algorithm classes. Limiting the algorithm classes to a small number (for the DCPS XX) will simplify the software design, implementation, and maintenance.

It is conceivable that in the future additional algorithms would be required that do not fit within the pre-defined classes. In that case, a decision will be made to either create a new class which admits the algorithms or to leave the algorithm as an exception to the general rule.

5.1 CURRENT PREDICTOR

Strictly speaking, the current predictor is not a class of protection algorithm. Rather, it is the fundamental method whereby currents are generated for use by all other algorithms. When the DCPS is actively protecting the NSTX-U device, $SOP \pm \delta \leq t_{NSTX} \leq EOP \pm \gamma$ (the P state)[1], the system selects the larger, in magnitude, of the two redundant input current signals to be used by the algorithms. This "auctioneering" process is executed at each time step.

[1] The time between $SOP \pm \delta$ and $EOP \pm \gamma$ is referred to in the CPS requirements document as the pulse "P" state. The usage of "state" in this context should not be confused with the more formal usage in the DCPS Software Requirements Document.

The collection of currents makes up the current vector $\{I\}$. The current vector $\{I\}$ includes the ohmic heating (OH) coil current, the poloidal field (PF) currents, the toroidal field (TF) current (note that for some calculations it may be convenient to augment the current vector with the plasma current I_p). In addition to the current values that comprise the current vector at each time step, we also require the same projected one DCPS time step into the future, assuming a plasma disruption event. The currents are projected into the future using the method of flux conservation [2]. We call the collection of predicted future values of current $\{I'\}$.

To calculate $\{I'\}$ we first partition the system mutual inductance matrix as

$$M = \left(\begin{array}{cc} \overline{L_{coils}} & \overline{M_{pl-coils}} \\ \hline \overline{M'}_{pl-coils} & L_{pl} \end{array} \right)$$

where L_{coils} is the typical (ncoils x ncoils) matrix describing the coil system, $M_{pl\text{-}coils}$ is an (ncoils x 1) vector representing the mutual couplings between the plasma and the coils ($M'_{pl\text{-}coils}$ is its transpose), and L_{pl} is the bulk plasma inductance. The predicted change in current is

$$\Delta I = \overline{L}_{coils}^{-1} \cdot \overline{M}_{pl-coils} \cdot \overline{I}_{pl}$$

where $\overline{L}_{coils}^{-1}$ is the matrix inverse of $\overline{L}_{coils}$. Thus

$$\overline{I}' = \overline{I} + \overline{L}_{coils}^{-1} \cdot \overline{M}_{pl-coils} \cdot \overline{I}_{pl}$$

the worst case "next-step" predicted current. For the remaining algorithms, the calculations shall be performed using both the actual and predicted current vectors {I} and {I'}.

5.2 ACTION INTEGRALS

During the P state the action integral, $\int i^2(t)dt$, is computed for each coil system and compared to a pre-determined limiting value. If we denote the action integral variable for the "i^{th}" circuit at the k^{th} time step as

$$A_k^i$$

then there are two parts to consider. The first part consists of the accumulated "action" since the start of the P state "AA"

$$AA_k^i = AA_{k-1}^i + {I_k^i}^2 \Delta t$$

where Δt is the DCPS time step. The second part is an estimate of the action that would accumulate as a result of a fault and a subsequent L/R decay from the present value of the current "AD"

$$AD_k^i = \frac{{I_k^i}^2 \tau}{2}$$

where τ is the L/R time constant of the circuit based on the self inductance and 20°C resistance of the circuit. The total action integral at any instant of time for the i^{th} circuit is then

$$A_k^i = AA_k^i + AD_k^i$$

or

$$A_k^i = AA_{k-1} + {I_k^i}^2 \Delta t + \frac{{I_k^i}^2 \tau}{2}$$

Note that mutual coupling between the coils and circuit resistance temperature effects are ignored in this approximation for the initial version of the DCPS.

5.3 FORCES AND MOMENTS

In the P state the radial force, F_r, the vertical force, F_z, and certain moments (torques), T, will be computed for selected coils and combinations thereof. The force and moment algorithms will not only be calculated for both the present and predicted future values of the current ({I} and {I'}) but also for multiple plasma models. The formula for each of these quantities can be expressed as a quadratic form in the current. Thus, for the ith coil circuit, the equation is

$$X_i = I_i \sum_j CX_{i,j} I_j$$

where X is the "force-like" quantity, CX is the influence matrix for the quantity, and j is a summation index that ranges over the coils included in the summation. The elements of the influence matrix

$$CX_{i,j}$$

are the contribution to the force-like quantity for coil current combination

$$I_i * I_j.$$

For example, the equation for the radial force on the ith coil is

$$Fr_i = I_i \sum_j CFr_{i,j} I_j.$$

In this equation Fr is the "force-like" quantity and the $CFr_{i,j}$ are the radial force influence matrix elements. Force (F_r and F_z) and moment (torque) influence matrices have been determined for the NSTX-U coil set for various simple plasma models. To use the influence matrix technique correctly, one must always be careful to use the appropriate influence matrix for the plasma model assumed, in all calculations.

5.4 DERIVED LIMIT VARIABLES

Another class of limit variables is the so called “derived limit variables.” Derived limit variables are those that can be defined as weighted sums of the basic limit variable types described above. There are two types of derived limit variables.

5.4.1 Derived Type I

A Type I derived limit variable is a linear weighted sum of the basic limit variable types (currents, action integrals, forces, and torques). The generic form of the Type I derived limit variable is

$$Y_X = K_0 + \sum_i C_i^I \, \mathrm{I}_i + C_i^A \, \mathrm{A}_i + C_i^{Fr} \, \mathrm{Fr}_i + C_i^{Fz} \, Fz_i + C_i^T \, \mathrm{T}_i$$

where “i” ranges over the coils.

5.4.2 Derived Type II

A Type II derived limit variable is a square root of a sum-of-squares of Type I derived limit variables. The general form of the Type II derived limit variable is

$$Z = \sqrt{Y_A^2 + Y_B^2 + \cdots + Y_J^2}$$

where the Y_X are Type I derived limit variables as defined above.

5.5 Other/Miscellaneous`

It is conceivable that an algorithm does not fit into one of the pre-defined classes. In these cases the algorithm will be classified as type "Other." Note that all of the current algorithms fit into one of the defined classes.

6 ALGORITHM EXAMPLES

A few simple examples should serve to illustrate how the "types" of algorithms operate in practice. For illustrative purposes we'll consider a subset of the NSTX-U coil systems (PF1AU, PF5L, and OH) and the plasma (represented here by I_p) from the configuration dated March 8, 2011 (110308). In the examples that follow, we'll use the following matrices:

$$\overline{M} = \begin{pmatrix} 2.03 & 0.153 & 2.87 & 0.00188 \\ 0.153 & 12.2 & 1.56 & 0.0397 \\ 2.87 & 1.56 & 36.8 & 0.0437 \\ 0.00188 & 0.0397 & 0.0437 & 0.00112 \end{pmatrix} mH$$

$$\overline{CFr} = \begin{pmatrix} 1179 & 60.1 & -266.5 & 2.5 \\ 2.5 & 12.2 & -51.6 & -1.8 \\ 5328.7 & 1438.1 & 3.4894 & 80.7 \\ 0.5 & 9.1 & -1.9 & 0.2 \end{pmatrix} \frac{lbf}{kA^2}$$

$$\overline{CFz} = \begin{pmatrix} 0 & -8.1 & -154.1 & -0.6 \\ 8.1 & 0 & 19.8 & 2.6 \\ 154.1 & -19.8 & 0 & 0 \\ 0.6 & -2.6 & 0 & 0 \end{pmatrix} \frac{lbf}{kA^2}$$

The ordering of [PF1AL, PF5L, OH, Ip] will be maintained throughout the subsequent calculations. We also need to define some simple current waveforms for use as input to the calculations. The values here can be assumed to be the output of the auctioneering process that selects the largest magnitude current from the redundant input signals.

Time [ms]	I_{PF1AU} [kA]	I_{PF5L} [kA]	I_{OH} [kA]	I_P [MA]
t_0	-7.0	34.0	-18.0	1.0
$t_0 + 0.2$	-7.0	34.0	-17.6	1.0
$t_0 + 0.4$	-6.8	34.0	-17.0	1.1
$t_0 + 0.6$	-6.6	34.0	-16.6	1.1
$t_0 + 0.8$	-6.4	33.8	-16.2	1.2
$t_0 + 1.0$	-6.4	33.8	-15.6	1.3

Table 1- Current values for example calculations.

6.1 Current Prediction

The current prediction algorithm is used to approximate the effect of a plasma current quench on the PF currents after the quench. It is a “worst case” algorithm that is meant to envelope the expected current excursions following the current quench. To illustrate the algorithm we’ll assume that we're at the last time point in the time series above (t_0 + 1.0ms). The current prediction algorithm assumes that the plasma current disappears in one time step (for us 0.2 ms). The calculation uses a partitioned version of the total inductance matrix. Recalling the partitioning suggested in Section 4.1 and the matrices defined above

$$\overline{L}^{-1}{}_{coils} = \begin{pmatrix} 0.5537 & -0.0014 & -0.0431 \\ -0.0014 & 0.0824 & -0.0034 \\ -0.0431 & -0.0034 & 0.0307 \end{pmatrix} \cdot 1000 \cdot \frac{1}{mH}$$

$$\overline{M}_{pl-coils} = \begin{pmatrix} 0.00188 \\ 0.0397 \\ 0.0437 \end{pmatrix} mH$$

$$\overline{I}_{coils}^{initial} = \begin{pmatrix} -6.4 \\ 33.8 \\ -15.6 \end{pmatrix} kA$$

and

$$I_{pl}^{initial} = 1.3MA$$

So the change ΔI in the currents due to a current quench would be

$$\overline{L}_{coils}^{-1} \cdot \overline{M}_{pl-coils} \cdot \overline{I}_{pl} = \begin{pmatrix} 0.5537 & -0.0014 & -0.0431 \\ -0.0014 & 0.0824 & -0.0034 \\ -0.0431 & -0.0034 & 0.0307 \end{pmatrix} \cdot 1000 \cdot \begin{pmatrix} 0.00188 \\ 0.0397 \\ 0.0437 \end{pmatrix} \cdot \frac{1}{1000} \cdot 1.3MA$$

$$= \begin{pmatrix} -1.17 \\ 4.05 \\ 1.46 \end{pmatrix} kA$$

The predicted currents are then

$$\overline{I}_{coils}^{final} = \overline{I}_{coils}^{initial} + \Delta I = \begin{pmatrix} -6.4 \\ 33.8 \\ -15.6 \end{pmatrix} kA + \begin{pmatrix} -1.17 \\ 4.05 \\ 1.46 \end{pmatrix} kA = \begin{pmatrix} -7.57 \\ 37.86 \\ -14.14 \end{pmatrix} kA = \overline{I}'$$

Note that this calculation is only meant illustrative of the method and does not represent a realistic scenario. Executing the current prediction algorithm for the entries in Table 1 results in

Time [ms]	I_{PF1AU} [kA]	I'_{PF1AU} [kA]	I_{PF5L} [kA]	I'_{PF5L} [kA]	I_{OH} [kA]	I'_{OH} [kA]
t_0	-7.0	-7.9	34.0	37.1	-18.0	-16.9
$t_0 + 0.2$	-7.0	-7.9	34.0	37.1	-17.6	-16.5
$t_0 + 0.4$	-6.8	-7.8	34.0	37.4	-17.0	-15.8
$t_0 + 0.6$	-6.6	-7.6	34.0	37.4	-16.6	-15.4
$t_0 + 0.8$	-6.4	-7.5	33.8	37.8	-16.2	-14.8
$t_0 + 1.0$	-6.4	-7.6	33.8	37.9	-15.6	-14.1

Table 2 - Predicted current values using flux conservation method.

6.2 Action

Action integrals, $\int i^2(t)dt$, are used to estimate conductor temperature rise in the coils. Two action integral estimates are required: an estimate of the action up to the current time point, and an estimate of the additional action that would be accumulated if a fault were to occur and the current were to decay exponentially from the present state. We call the two components

$$AA_k = AA_{k-1} + I_k^2 \Delta t$$

and

$$AD_k = \frac{I_k^2 \tau}{2}$$

with the total estimated action at any time step "k" being

$$A_k = AA_k + AD_k$$

where Δt is the system time step and τ is the L/R time constant of the circuit under consideration. As an example, consider the PF_{1AU} coils system. If we assume that the circuit resistance is 8.93 mΩ (τ = 0.257 s), the time step is 0.2 ms, and the initial action is 25 kA^2s, then using the values in the table above we see that

$$AA_0 = 25kA^2s$$

the accumulated contribution is

$$AA_1 = AA_0 + (-7.0kA)^2 \cdot 0.0002s = 25kA^2s + 0.0098kA^2s = 25.0098kA^2s$$

the estimated decay portion is

$$AD_1 = \frac{(-7.0kA)^2 \cdot 0.257s}{2} = 6.30kA^2s$$

so the estimate of the total possible action (including L/R decay) at this time i

$$A_1 = AA_1 + AD_1 = 25.0098kA^2s + 6.30kA^2s = 31.31kA^2s$$

this is the value that would be compared to the coils I^2t limit. Similarly for the next time step,

$$AA_2 = AA_1 + (-7.0kA)^2 \cdot 0.0002s = 25.0098kA^2s + 0.0098kA^2s = 25.02kA^2s$$

and

$$AD_2 = \frac{(-7.0kA)^2 \cdot 0.257s}{2} = 6.30kA^2s$$

so that

$$A_2 = AA_2 + AD_2 = 25.02kA^2s + 6.30kA^2s = 31.32kA^2s$$

If we execute the calculation for the entire I_{PF1AU} time series for both I_{PF1AU} and I'_{PF1AU} we get

Time [ms]	I_{1AU} [kA]	AA_{1AU} [kA^{2}s]	AD_{1AU} [kA^{2}s]	A_{1AU} [kA^{2}s]	I'_{1AU} [kA]	AA'_{1AU} [kA^{2}s]	AD'_{1AU} [kA^{2}s]	A'_{1AU} [kA^{2}s]
t_0	-7.0	0.0098	6.3	31.31	-8.08	0.0131	8.39	33.4
$t_0 + 0.2$	-7.0	0.0098	6.3	31.32	-8.08	0.0131	8.39	33.42
$t_0 + 0.4$	-6.8	0.0092	5.94	30.97	-7.88	0.0124	7.98	33.02
$t_0 + 0.6$	-6.6	0.0087	5.6	30.64	-7.68	0.0118	7.58	32.63
$t_0 + 0.8$	-6.4	0.0082	5.26	30.31	-7.48	0.0112	7.19	32.25
$t_0 + 1.0$	-6.4	0.0082	5.26	30.31	-7.48	0.0112	7.19	32.26

Table 3 - Action integral calculation for the PF1AU current.

Note that the total action does not strictly increase as a function of time as expected because of its dependence on the exponential decay approximation.

6.3 Forces

Force calculation is one of the most basic DCPS algorithms. The resulting forces are used for calculating many of the derived limit variables. The force calculation is accomplished using influence matrices. The force influence matrices are determined by the NSTX-U coil configuration and the shape of the assumed plasma model. In matrix notation,

$$\overline{F}_u = diag(\overline{I}) \cdot \overline{CF}_u \cdot \overline{I}$$

where F_u is the resulting force vector (the subscript u is either r or z), I is the current vector (which in this case includes the plasma current I_p), and CF_u is the appropriate force influence matrix. Using the currents (actual and predicted) from the $t_0 + 0.8$ ms time point,

$$\overline{F}_r = \begin{pmatrix} -6.4 & 0 & 0 & 0 \\ 0 & 33.8 & 0 & 0 \\ 0 & 0 & -16.2 & 0 \\ 0 & 0 & 0 & 1200 \end{pmatrix} kA \cdot \begin{pmatrix} 1179 & 60.1 & -266.5 & 2.5 \\ 2.5 & 356.3 & -51.6 & -1.8 \\ 5328.7 & 1438.1 & 34894 & 80.7 \\ 0.00188 & 0.0397 & 0.0437 & 0.00112 \end{pmatrix} \frac{lbf}{kA^2} \cdot \begin{pmatrix} -6.4 \\ 33.8 \\ -16.2 \\ 1200 \end{pmatrix} kA$$

$$\overline{F}_r = \begin{pmatrix} -11708 \\ 361120 \\ 7354522 \\ 751503 \end{pmatrix} lbf$$

To simplify the matrix notation, which simplifies the programming, the plasma current is included throughout the calculation. This results in calculated force components that are not used in any calculations and can be discarded.

Similarly

$$\overline{F}_z = \begin{pmatrix} -6.4 & 0 & 0 & 0 \\ 0 & 33.8 & 0 & 0 \\ 0 & 0 & -16.2 & 0 \\ 0 & 0 & 0 & 1200 \end{pmatrix} kA \cdot \begin{pmatrix} 0 & -8.1 & -154.1 & -0.6 \\ 8.1 & 0 & 19.8 & 2.6 \\ 154.1 & -19.8 & 0 & 0 \\ 0.6 & -2.6 & 0 & 0 \end{pmatrix} \frac{lbf}{kA^2} \cdot \begin{pmatrix} -6.4 \\ 33.8 \\ -16.2 \\ 1200 \end{pmatrix} kA$$

$$\overline{F}_z = \begin{pmatrix} -9646 \\ 91201 \\ 26798 \\ -108354 \end{pmatrix} lbf$$

At t_0 + 0.8 ms, I' is

$$I' = \begin{pmatrix} -7.5 \\ 37.8 \\ -14.8 \end{pmatrix} kA$$

$$F_r = \begin{pmatrix} -7.5 & 0 & 0 & 0 \\ 0 & 37.8 & 0 & 0 \\ 0 & 0 & -14.8 & 0 \\ 0 & 0 & 0 & 1200 \end{pmatrix} \cdot \begin{pmatrix} 1179 & 60.1 & -266.5 & 2.5 \\ 2.5 & 356.3 & -51.6 & -1.8 \\ 5328.7 & 1438.1 & 34894 & 80.7 \\ 0.00188 & 0.0397 & 0.0437 & 0.00112 \end{pmatrix} \cdot \begin{pmatrix} -7.5 \\ 37.8 \\ -14.8 \\ 1200 \end{pmatrix}$$

$$= \begin{pmatrix} -2998 \\ 454902 \\ 5997558 \\ 791435 \end{pmatrix} lbf$$

and

$$\overline{F}_z = \begin{pmatrix} -7.5 & 0 & 0 & 0 \\ 0 & 37.8 & 0 & 0 \\ 0 & 0 & -14.8 & 0 \\ 0 & 0 & 0 & 1200 \end{pmatrix} \cdot \begin{pmatrix} 0 & -8.1 & -154.1 & -0.6 \\ 8.1 & 0 & 19.8 & 2.6 \\ 154.1 & -19.8 & 0 & 0 \\ 0.6 & -2.6 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} -7.5 \\ 37.8 \\ -14.8 \\ 1200 \end{pmatrix}$$

$$= \begin{pmatrix} -9443 \\ 102704 \\ 28161 \\ -121422 \end{pmatrix} lbf$$

We can construct a table of radial and vertical forces using the currents in Tables 1 and 2 (the values previously calculated are highlighted in red).

Time [ms]	Fr_{1AU} [lbf]	Fr_{5L} [lbf]	Fr_{OH} [lbf]
t_0	-7764	381135	9645009
$t_0 + 0.2$	-7018	380433	9185022
$t_0 + 0.4$	-9049	373223	8360736
$t_0 + 0.6$	-9636	372539	7914626
$t_0 + 0.8$	-11708	361120	7354522
$t_0 + 1.0$	-12299	353935	6629693

Table 4 - Radial force, F_r, calculation for {I}.

Time [ms]	Fz_{1AU} [lbf]	Fz_{5L} [lbf]	Fz_{OH} [lbf]
t_0	-13317	72970	31512
$t_0 + 0.2$	-12885	73238	30812
$t_0 + 0.4$	-11482	82395	29237
$t_0 + 0.6$	-10738	82718	28037
$t_0 + 0.8$	-9646	91201	26798
$t_0 + 1.0$	-8672	100249	25806

Table 5 - Vertical force, F_z, calculation for {I}.

Time [ms]	Fr_{1AU} [lbf]	Fr_{5L} [lbf]	Fr_{OH} [lbf]
t_0	464	454680	8412620
$t_0 + 0.2$	-1009	456019	9190965

$t_0 + 0.4$	-283	453451	7115915
$t_0 + 0.6$	-1258	452698	6704406
$t_0 + 0.8$	-2998	454902	5997558
$t_0 + 1.0$	-2688	449195	5261239

Table 6 - Radial force, F_r, calculation for $\{I'\}$.

Time [ms]	Fz_{1AU} [lbf]	Fz_{5L} [lbf]	Fz_{OH} [lbf]
t_0	-13492	80159	32965
$t_0 + 0.2$	-14344	79646	34331
$t_0 + 0.4$	-11514	91219	30670
$t_0 + 0.6$	-10750	91575	29419
$t_0 + 0.8$	-9443	102704	28161
$t_0 + 1.0$	-8289	113165	27075

Table 7 - Vertical force, F_z, calculation for $\{ I' \}$.

6.4 Force Combinations (Derived Type I)

As an example of a Type I derived limit variable, consider combination forces. There are many such combinations that are of interest for the DCPS. Some examples from the Design Point Spreadsheet are:

$$Y_0 = 1 \cdot Fz_{1CL} + 1 \cdot Fz_{2L}$$

$$Y_1 = 1 \cdot Fz_{1AU} + (-1) \cdot Fz_{1BU}$$

$$Y_2 = 1 \cdot Fz_{3U} + 1 \cdot Fz_{4U} + 1 \cdot Fz_{5U}$$

we can construct an example using the forces calculated above

$$Y_3 = 1 \cdot Fr_{1AU} + 1 \cdot Fr_{OH}$$

If we consider the radial forces at each time to be a row vector we can formulate the calculation using matrix notation. Using data from Table 4 at t_0 + 0.8 ms

$$\overline{F}_r = (-11708 \quad 361120 \quad 7354522)\ lbf$$

and

$$\overline{C}_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

then

$$Y_3 = \overline{F}_r \cdot \overline{C}_3 = (-11708 \quad 361120 \quad 7354522) \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} lbf = 7342814\ lbf$$

This force combination would be calculated for both $\{I\}$ and $\{I'\}$ at each time step.

7 Algorithm Database

The algorithm database is used to store the algorithm data. The database will most likely be constructed using MDSPlus, but other solutions are acceptable. The algorithm database contains all of the information required by the DCPS to execute the algorithm.

1. Algorithm number – a unique numeric algorithm identifier.
2. Algorithm name – a unique identifier used to differentiate between algorithms. The algorithm name should be descriptive.
3. Description – a few sentences describing the algorithm.
4. Number of inputs – number of algorithm inputs.
5. Input names – names of the input variables.
6. Input units – units of the algorithm input variables.
7. Input scale factor – multiplicative scale factor(s) for the input variables.
8. Number of outputs – number of algorithm outputs.
9. Output names – names of the algorithm output variables.
10. Output units – units of algorithm output variables.
11. Output scale factor - multiplicative scale factor(s) for the output variables.
12. Dependencies – other algorithms (algorithm number) whose outputs are required as input.
13. Class – class of algorithm.
14. Limit variable maximum – scalar or array of maximum allowable values of the algorithm output variable(s).
15. Limit variable minimum – scalar or array of minimum allowable values of the algorithm output variable(s).
16. Data –data (scalars, vectors, matrices, etc.) required to execute the algorithm.

17. Enable bit – bit indicating whether the algorithm is executed (normally all algorithms are turned on, except possibly during algorithm development).
18. Active bit – when set, this bit which allows the algorithm to effect the DCPS L1 fault output. Allows for algorithms to be enabled, but not protecting, during development to minimize nuisance trips.
19. Executed bit – status bit that is set each time the algorithm is completed (and cleared prior to the start of each time step). This bit can be used by the system and other algorithms to assess algorithm execution progress.

8 ALGORITHM EXECUTION

The DCPS executes each algorithm at each time step (200 μs) between the times of EOP and SOP. The DCPS will execute each algorithm not only using the input current vector $\{I\}$ but also using the "next-time-step" predicted current vector $\{I^{'}\}$. The system will be able to test and deploy new algorithms using combinations of the "Enable" and "Active" bits. There is a minimal set of DCPS algorithms (these will initially be the "Day 0" algorithms) that must be both active and protecting to provide effective system protection. The DCPS shall check to ensure that these algorithms are enabled and active prior to the beginning of each shot attempt (SOP).

8.1 Algorithm Queues and Precedence

To allow for effective use of the multi-processor computer system, the DCPS will utilize algorithm queues. An algorithm queue will nominally be associated with a processor or thread. The arrangement of algorithms in the processing queues should optimize processor throughput while maintaining proper algorithm processing precedence (i.e., a dependent algorithm cannot precede an algorithm from which it requires input data). Reconfiguration of the algorithm queues is accomplished with the DCPS in maintenance mode. The DCPS shall check for proper algorithm queue arrangement prior to the beginning of each shot attempt (SOP).

8.2 Algorithm Outputs

The results of algorithm execution, the algorithm outputs, will be saved and archived by the DCPS system. Each algorithm output will be saved in the MDSPlus data system in a form suitable for viewing and data analysis. The outputs will be saved using unique, descriptive identifiers. In addition, it is conceivable that the user might want to view intermediate results of algorithm calculations. For example, the user might want to look at one term in a radial force calculation,

$$T_r^{1,2} = I_1 \times CFr_{1,2} \times I_2$$

or perhaps a couple of terms from a Type I derived limit variable,

$$T_{4,5} = C_{PF5}^{Fr} \mathrm{Fr}_{PF5} + C_{PF4}^{Fr} Fr_{PF4}$$

or just simple arithmetic functions based on the algorithm inputs and constant data

$$T_A = K_1 I_{PF4} + K_2 I_{PF5}$$

9 ALGORITHM LIST

The following is a list of proposed DCPS algorithms. The type of the algorithm is noted along with whether or not the algorithm is required for Day 0 operation. For algorithm types we will use the following numbering scheme:

0. Current selection
1. "Next time step" current prediction
2. Action integral computation
3. Influence matrix "force-like" quantity computation
4. Type I derived quantity
5. Type II derived quantity

ALGORITHM NAME	ALGORITHM TYPE	DAY 0
Current selection	0	✓
Next step current prediction	1	✓
Action integrals	2	✓
Radial/Vertical Force calculation	3	✓

Force combinations	4	✓