ANEXO II

FÓRMULAS PREDETERMINADAS DE MODEL

A continuación se listan las fórmulas que Model nos proporciona para realizar cálculos con los vectores. Las fórmulas pueden combinarse entre ellas para obtener una nueva fórmula que sea de nuestro interés.

Todas las fórmulas realizan sus cálculos utilizando doble precisión de punto flotante, incluso con funciones Booleanas, en las cuales el cero es Falso y distinto de cero Verdadero.

Para introducir valores del tipo Bad Value debe utilizarse la palabra reservada BAD en lugar del valor –9999.

El rango de valores permitidos para las fórmulas está comprendido entre -10^20 y 10^20.

@Log(value) Natural logarithm of a positive real value

@Log10(value) Common logarithm of a positive real value

@Exp(value) Natural antilogarithm of a value

@Sqrt(value) Square root of a positive real value

@Abs(value) Absolute value of a value

@Interp(value,x1, y1, x2, y2[, x3, y3…]) Piece-wise linear interpolation: (x,y) pairs define function

@InterpBad(vector) Interpolate over bad values in a vector

@Cav(vector, nsamples) Central average of a vector. nsamples must be odd

@EFilt(vector, factor) Exponential filter: out(1) = in(1),
out(i) = out(i-1) * factor + in(i) * (1 - factor)

@Diff(vector) Difference a vector: out(1) = 0,
out(i) = in(i) - in(i-1)

@Integ(vector, bias) Integrate a vector: out(1) = in(1) + bias,
out(i) = in(i) + out(i-1)

@Shift(vector, nsamples) Shift a vector. Negative nsamples means shift earlier.

@VaLim(vector, vallow, valhigh) Mark samples outside range vallow and valhigh as BAD

@MkBad(vector, nfirst1, nlast1[, nfirst2, nlast2…]) Mark sample ranges as BAD. nfirst and nlast are sample numbers

@LoClp(vector, valclip, valnew) Set values less than valclip to valnew

@HiClp(vector, valclip, valnew) Set values greater than valclip to valnew

@Max(value1[, value2, value3 …]) Maximum of a set of values

@Min(value1[, value2, value3 …]) Minimum of a set of values

@IF(test, value1, value2) value1 if test is "true" or value2 if test is "false"

@EQ(value1, value2) "true" if value1 is equal to value2

@NE(value1, value2) "true" if value1 is not equal to value2

@GT(value1, value2) "true" if value1 is greater than value2

@GE(value1, value2) "true" if value1 is greater than or equal to value2

@LT(value1, value2) "true" if value1 is less than value2

@LE(value1, value2) "true" if value1 is less than or equal to value2

@OR(value1, value2[, value3…]) "true" if any value in a list of values is true

@AND(value1, value2[, value3…]) "true" if all values in a list of values are true

@NOT(value) "true" if value is "false"

@BADVAL(value) "true" if value is a BAD value

@PCT2(Temp<degF>, Pres<psig>, Rpres<psig>, B, C, Bias<degF>, LogInd<1=ln, 2=log10>) Pressure Compensated Temperature

@PFPCT(Temp<degF>, Pres<psig>, StdP<psig>, WatK) Petroleum Fraction Pressure Compensated Temperature

@REFPCT(Temp<degF>, Pres<psig>, StdP<psig>, Bias<degF>) Refinery Pressure Compensated Temperature

@SELECT(N, V1, V2, V3, …)x Returns Vn based on value of N

@MAPRANGE(X, L1, I1, L2, I2, …, Ln, In, In+1) Returns range indicator In if X <= Ln