# The MTP gain equation

An analysis of the MTPBin VB code and an alternative approach

Tom Baltzer

In reverse engineering the Visual Basic (VB) code which implements the retrieval algorithm (known as MTPBin) for the Microwave Temperature Profiler (MTP), the gain equation has proven to be significantly challenging. Gain is used to convert MTP counts into a temperature (TA indexed by channel number and scan angle) in this equation from the function MTPbin::TBcalculation:

1. *TA(i, j) = OAT + (C(i, j) - C(i, LocHor)) / g(i)*

Where:

* C() is an array of raw counts from the instrument indexed by channel number (i) and angle (LocHor is the angle associated with the horizontal scan).
* OAT is the outside air temperature (provided in the Iwgadts line) in degrees kelvin.
* g() is an array of per channel gain values (for converting counts to temperatures)

*It is curious that OAT which is a measurement of the ambient outside air temperature is being adjusted by a value that ostensibly is a brightness temperature.*

g() in equation (1) is initialized to a value at the beginning of a flight and then is adjusted on a scan by scan basis in an iterative fashion. The latter piece is described first.

# Scan to Scan Adjustment to the per channel gain: g(i)

g() in equation (1) is determined iteratively for each MTP scan using the following equation:

Where:

* i is channel number
* Wtg is a constant = 0.1
* Gnd is calculated as follows:

*Gnd(i) = dND(i) / (Cnd0(i) \* (1# + Cnd1(i) \* dT + Cnd2(i) \* dT ^ 2))*

Where:

* i is channel number
* dND(i) is the difference in counts between the scan of the target with the noise diode turned on, and with it turned off. AKA Noise Diode Deflection.
* Cnd0, Cnd1 and Cnd2 are values whose values are established through the TB Fit process. They are found on the gain tab as shown in figure 1
* dT = Tnd – TrefND (which is irrelevant in our case because Cnd1 andCnd2 are always 0.00)

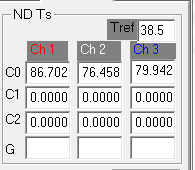


Figure 1: Cnd array values

Thus for the NCAR MTP instrument the equation simplifies to:

*Gnd(i) = dND(i) / (Cnd0(i))*

This implies that Cnd is a representation of the difference in “temperature” of the target with the noise diode on and with the noise diode off.

Looking at comments about Cnd0 in the code nets the following:

* “Noise Diode fit parameter”
* “Noise Diode Noise Temperatures”
* “Noise Diode Temperature Fit – Offset”

**Most notable about the gain adjustment is that it in no way uses Outside Air Temperature (OAT). It has been our understanding that OAT is a key component in the gain determination.**

# Initialization of the per channel gain: g(i)

Initialization of g(i) is described with the following bit of code:

' First time thru; give running avg gains: g(1), g(2), g(3) gain equation value

If g(1) = 0 Then g(1) = Geqn(1): g(2) = Geqn(2): g(3) = Geqn(3)

Code Segment 1: Initializing gain

The VB code used to calculate Geqn has myriad options for determining how to go about the calculation. For this reason, I’ve chosen to assume that certain elements of the options are set in a particular way (which is consistent with how the software is used in post-production). The key to this is how the user interface is configured on the Gain tab in the Gain Equation section of the window (shown in Figure 2).

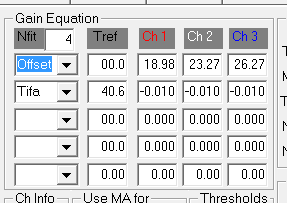


Figure 2: The Gain Equation portion of the Gain Tab in MTPBin

In particular the settings of Offset and Tifa on the left hand side of Figure 2 create a particular path for calculating Geqn as follows.

From: MTPbin::GainGE:

For i = 1 To Channels

Geqn(i) = GEC(i, 1)

For j = 2 To Nfit

Geqn(i) = Geqn(i) + FP(j) \* GEC(i, j)

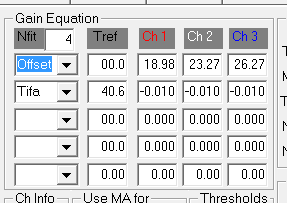
Next j

Next i

Code Segment 2: Geqn Calculation

The GEC array comes from the values shown in figure 2 and repeated here in figure 3. Also shown in Figure 3 is the GOF array which is used below.

GEC matrix



GOF array

Figure 3: GEC values as used in code segment

I continue to struggle to find how these values (GOF and GEC) are calculated. I think it must be in the initial setup of a flight. They are found in the .REF file for the flight, so determining when that gets written is key to determining their initialization values.

So ultimately Geqn is calculated as follows:

For example, using the values from Figure 3, Geqn(1) = FP(2)\*-0.010 + 18.98

Next we need to know the value of FP(2).

Where:

* GOF(2) is the second element of the GOF array shown in Figure 3 and seems to be consistently 40.6 from flight to flight and project to project.

And Tifa is set to a value also known as Tmix (the temperature of the mixer). Tmix is calculated on a per scan basis using values from the set of Platinum wire line values to MTPBin from the MTP controller software. There are 8 platinum wire values sent to MTP bin which I will label as Pt0 through Pt7 for the purpose of the following equations.

Where:

* AA = -244.3364635
* Bb = 0.462418
* cC = 0.0000588
* DD = -0.000000013

And r is calculated using platinum wire values as follows:

Comments from the code give some clue as to the meaning of the platinum wire values:

' "Rref 350 " 'R350

' "Target 1 " 'Ttlo

' "Target 2 " 'Tthi

' "Window " 'Twin

' "Mixer " 'Tmix

' "Dblr Amp " 'Tamp

' "Noise D. " 'Tnd

' "Rref 600 " 'R600

' R(0) = 350 'rref low

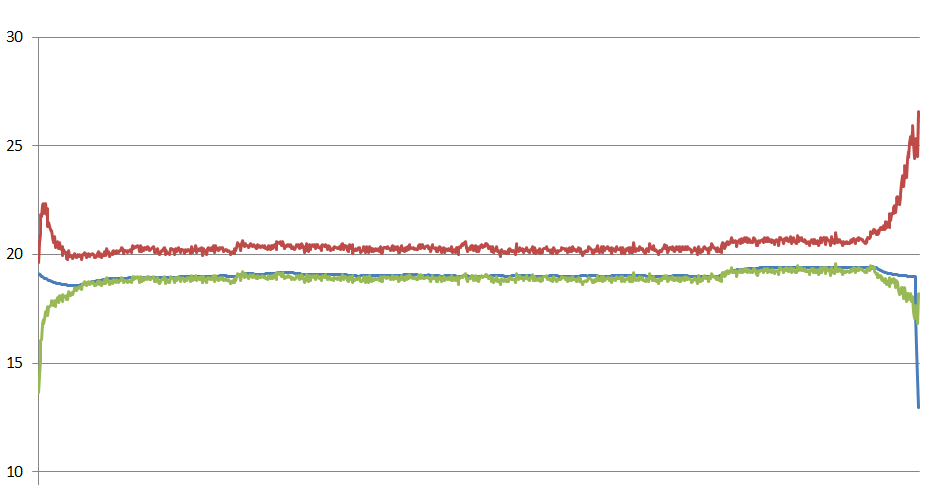
' R(7) = 600 'rref hi

# Implementation of the Gain equation using OAT

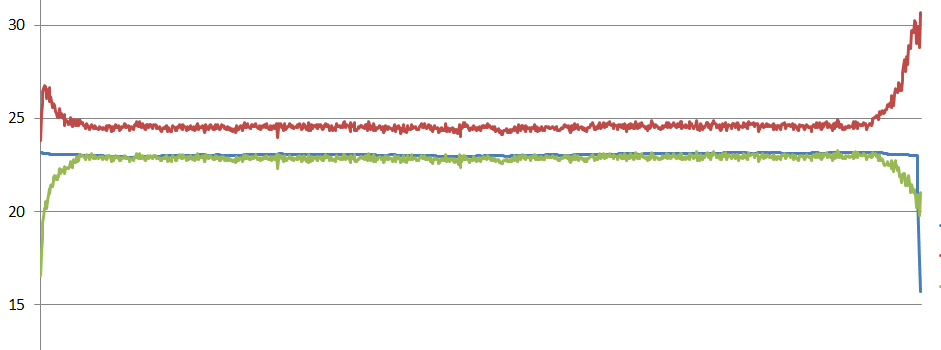
Our understanding of the gain for the MTP is that the simple difference of the MTP channel counts for the target scan minus the channel counts for the scan parallel to flight divided by the quantity of the target temperature minus the outside air temperature gives the gain. So I tried that approach on an MPEX flight and found that I was consistently getting a value about 20% greater than the gain value provided by MTPbin at flight level.

Discussing it with Julie, she mentioned that the emissivity of the target is supposed to be unity, but that the emissivity of the atmosphere will be less, so I attempted an experiment of multiplying the outside air temperature by an emissivity value less than one before subtracting it from the target temperature. With a little trial and error I arrived at an emissivity value of 0.965 which gave the following results (Blue line is MTP gain value, Red line is gain calculated without emissivity applied, Green line is gain calculated with emissivity applied):

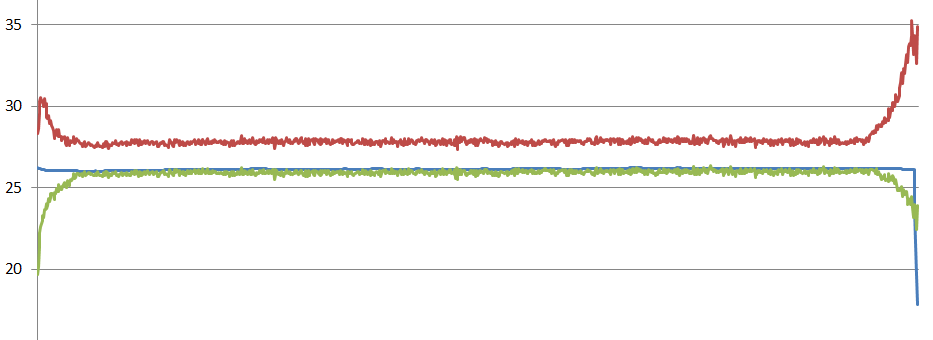
Channel 1:



Channel 2



Channel 3:

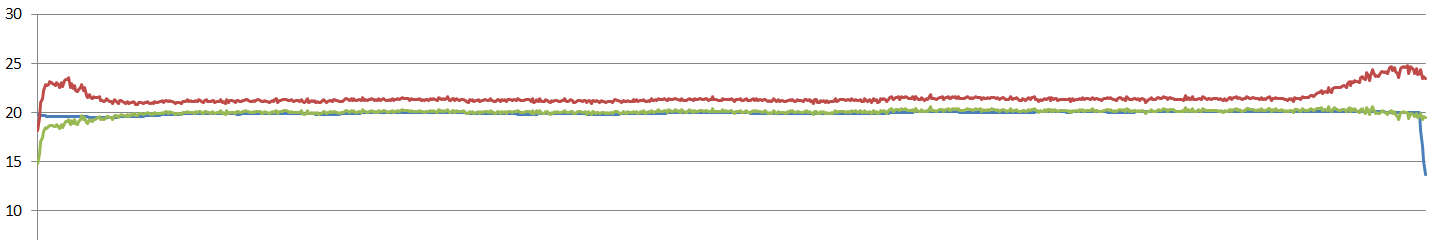


While the new gain value is noisier than the original, for most of the (MPEX) flight it compares nicely. The tails would seem to represent an adjustment to emissivity is needed based on temperature/pressure. For a first approximation I assume that emissivity changes linearly as a function of outside air temperature (which is at least half true and OAT is so readily available ☺ ).

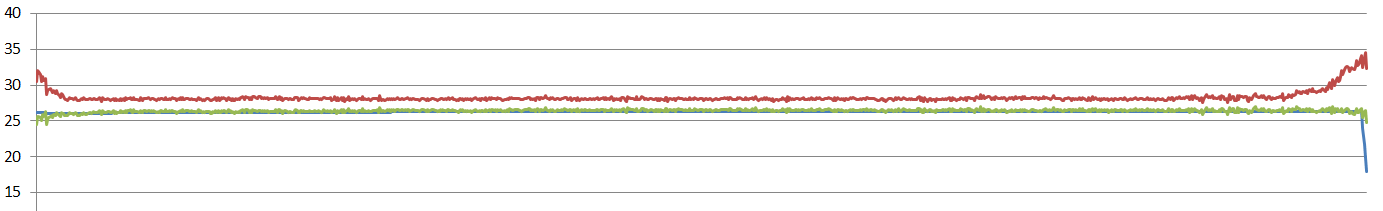
Using a HIPPO flight (where outside air temperature ranged quite a bit) I devised a linear equation to apply to the value of outside air temperature in order to determine the emissivity to apply to it. The equation is:

Using this equation I generated a new set of gain variables and then compared them to the MTPbin derived gain value for a number of flights. The following are the results with the line color representing the same values as before (blue = MTPbin gain, Red = uncorrected simple gain, Green = modulated emissivity applied to OAT in the simple gain approach).

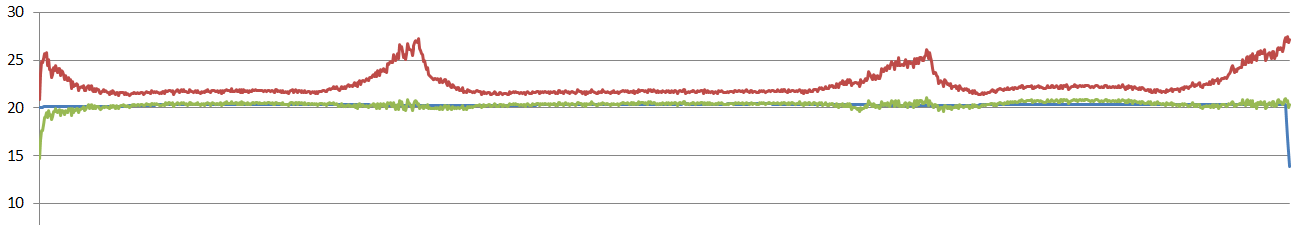
DEEPWAVE RF08:



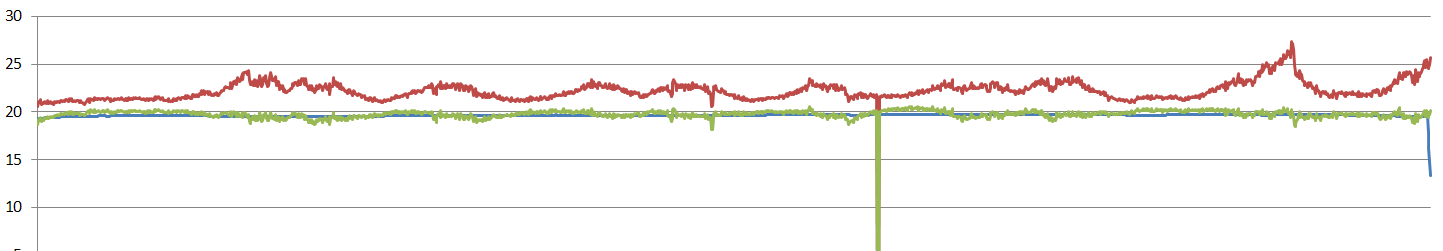
MPEX RF09:



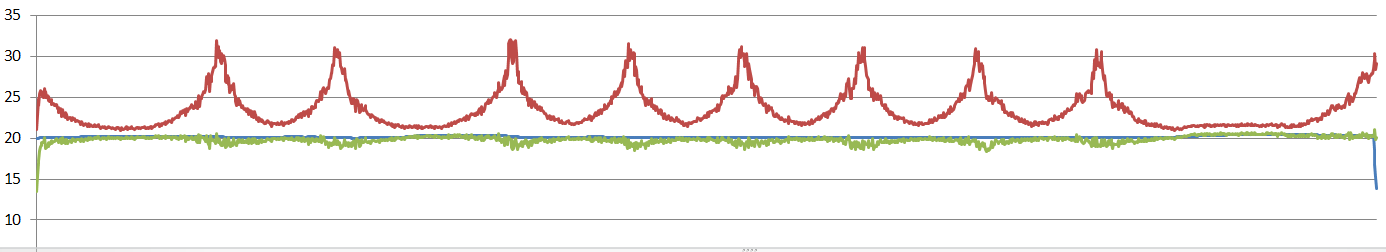
HIPPO-5 RMMA to ANC



HIPPO-5 ANC to Pole and back (note: spike is a few bad OAT values):



HIPPO-5 Hawaii to Rarotonga:



So until such time as we have a better grasp on calculating the gain using the approach found in MTPBin, this should be a reasonable substitute.