Allen Telescope Array

PAX box power level autotune

Janusz S. Kulpa

SETI Institute
189 Bernardo Ave, Suite 200
Mountain View, CA 94043
jkulpa@seti.org

February 14, 2020

Contents

1 General

This document describes the `ataautotune` and `atagetdetdbm` software: usage, processing scheme explanation, and data structures involved.

Signals from ATA antennas is converted from FR on coaxial cable to analog fiber signal in the *Pax Box*, and then converted back to RF in *RFCB* in the signal processing room. The RF to (optical) Fiber converter has a limited dynamic range and the input power, with 1 dB compression point at about 5 dBm[1].

The correct power level setting should ensure linearity of the signal converted to the optical signal. Since the feed and LNA are wide band, the autotune setting should not be dependent on the observation/recording frequency, but is dependent on the source being observed. This may cause some issues while observing strong transient signals and an alternative method may need to be established. Also, erroneous setting may happen if strong transient RFI is present during the *tuning* phase.

The *Pax Box* includes amplifiers (for x and y polarization), tuneable attenuators, and power detection diode. The modification of the attenuators settings is accomplished by calling `atagetpams`/`atasetpams` from obs network computer, whereas the readout of the power level is accomplished by calling `atagetdet`.

The `ataautotune` program is used to set the proper attenuators values

The `atagetdetdbm` program is used to show a detector values converted to dBm

Both programs are using calibration information from *Pax Box* measurements[2] to convert from raw detector readout to actual power level.

The power (detector) level fluctuates in time. Keep that in mind.

2 Usage

2.1 atagetdetdbm

```
Usage: Usage atagetdetdbm [options] antstrlist

Options:
  -h, --help     show this help message and exit
  -v, --verbose  prints additional information
```

[1] citation needed

[2] citation needed

The program is used to read (and convert) the *Pax box* detector values to dBm. Also returns additional information. The error messages are printed to *stderr*. *antstrlist* is a comma-separated list of antenna names, e.g. `atagetdetdbm 1a` or `atagetdetdbm 1a,2c,4j`

The example *stdout* printout (note that -v flag will give more *stdout* information)

```
atagetdetdbm 2c,1c
antenna  x pol [dBm]   sat flag x    x pol raw     y pol [dBm]   sat flag y    y pol raw     was measured
ant2c    -1.228795     0             0.378900      -39.423795    1             0.000700      0
ant1c    +1.744582     0             0.614100      -3.219506     0             0.244400      1
```

The field description as follows:

- antenna – the antenna name with “ant” prefix
- x/y pol [dBm] – the calculated “real” power at the input of RF/Optic converter
- sat flag x/y – information if the power calculation is unreliable (i.e. raw value is beyond approximation limit)
- x/y pol raw – the raw (linear, uncalibrated) readout of the detector value.
- was measured – information if the particular *Pax Box* was calibrated, or if the default data is taken

2.2 ataautotune

```
Usage: Usage ataautotune [options] antstrlist

Options:
  -h, --help            show this help message and exit
  -p POWER, --power=POWER
                        autotune target power in dBm
  -r RETRY, --retry=RETRY
                        number of autotune steps
  -t TOLERANCE, --tolerance=TOLERANCE
                        tolerance goal in dBm
  -v, --verbose         prints additional information
```

The program to autotune the antenna. Autotune should be executed after pointing the antenna to the desired source. By default, program returns nothing to *stdout*. Warning and error messages may be returned to *stderr*. The warnings may include:

- bad call (non-existent antenna, spelling/type errors)
- broken detector information
- failed to tune antenna in given retry steps

3 Detailed description

3.1 Code

The `ataautotune` and `atagetdetdbm` are available from the `ATA-Utils` github repository:

https://github.com/SETIatHCRO/ATA-Utils/tree/master/ataautotune

The restricted `ATA-Utils-priv` repository is necessary (imports `ATASQL` package to communicate with the database)

https://github.com/SETIatHCRO/ATA-Utils-priv/

The software is installed on the obs network computers. The software currently uses an internal `ah` package to indirectly call `atagetpams`, `atasetpams`, and `atagetdet`

The code is written in Python. Currently tested only in Python 2.7.

3.2 Calibration polynomial

Each measured *Pax Box* has a 5th order calibration polynomial, with separate coefficients stored for x and y polarization. Calibration polynomials for single tone and noise sources[3] are stored, but only single tone **(type='cw')** is used for calculation. The mapping between antenna name and *Pax Box* serial number is also stored in the database (see 3.5) The polynomial dictionary is returned by calling `autotunecommons.getPolynomials()`.

The polynomial allow for transition from a logarithmic raw detector values ($10\log_{10}(raw\_det\_value)$) to the RF/Optic converter input power). Only a certain range of the detector values can be accurately mapped to the power values[4]. If the detector value is beyond the conversion range, it should be saturated on the limit before using the polynomial coefficients. This may be accomplished by calling `autotunecommons.getLimittedPower()` function.

As presented in Fig. 1, the transition function is strongly linear

[3] cite Alex's MEMO

[4] polynomial should be used for interpolation, not extrapolation

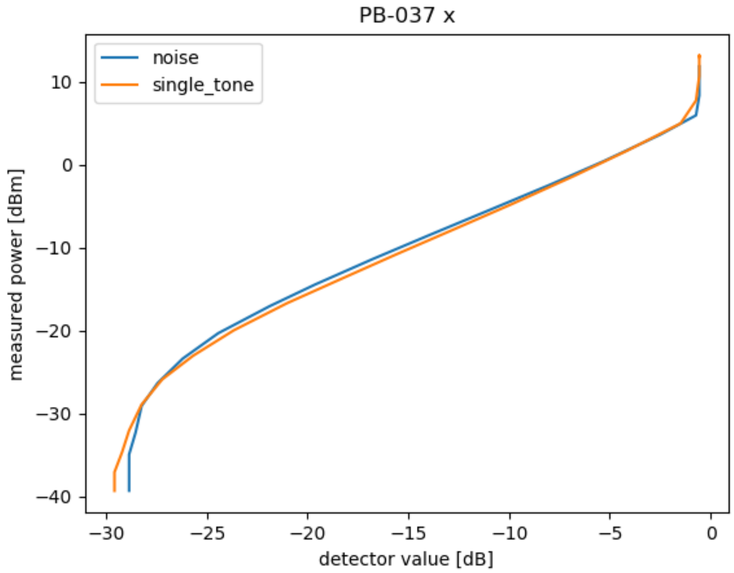


Figure 1: Measurement data for PB-037 pol x

3.3 Calibration polynomial generation

The calibration polynomial is calculated using `ATA-Utils/PamMeas/process_data.py`

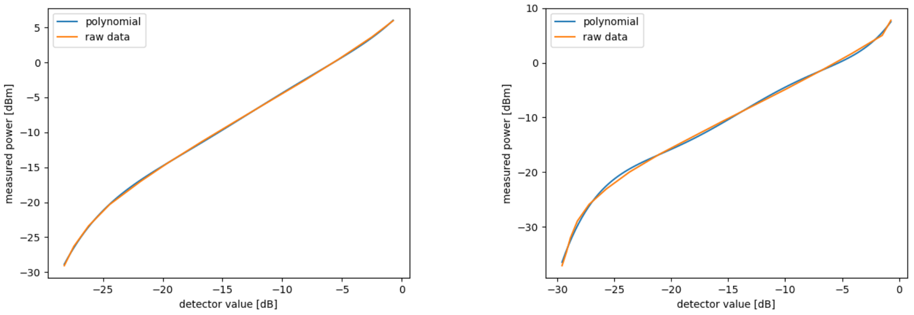
The software reads text files in the measurement directory. The part of filenames is hardcoded. There has to be PAXBOXSNx-a.txt for cw measurement data, PAXBOXSNx-b.txt for single tone data (y-a.txt and y-b.txt for y polarization). The text file is a new-line separated list of 21 detection values (linear scale) and 21 measurement values (dBm). Single dash '-' is permitted for detection values and means the saturation value. It was a "quick hack" for fast copying of data from pax box measurement report. Also, hard-coded is a saturation value of detector 0.88.

Some upper values of detector may be discarded (as they defaults to saturation value) to alleviate the data bias for high power. As of Feb 14, 2020, also the lower power data points may be discarded, it we believe that would bias the polynomial estimation.

After polynomial estimation, the data is uploaded to the database

Figure 2 shows the polynomial fit withing the valid power ranges of Pax Box PB-037

Moreover, for given polarization, between -20 and -3 dB detector values, the 1st order polynomial is estimated. If the estimation between noise and single tone polynomials differs in p0 by more than 1 dB and p1 0.1, both



(a) Noise source (b) Single tone source

Figure 2: Polynomial fit for PB-037, pol x

polynomials are marked as incoherent (flag iscoherent is set to 0)

3.4 Ataautotune execution

In ataautotune call, the polynomials for each requested antennas are fetched. If Pax Box data is not available, the default data (virtual PB-000) is returned.

RETRY trials of the pam settings are performed.

- raw detector values and pam values are fetched for each antenna
- for each untuned antenna
  - real power is calculated based on calibration polynomial and raw detection values.
  - power delta is calculated as a difference between target and current power
  - if power delta for both polarization is within a tolerance value, nothing is done and the antenna is removed from untuned list
  - if power delta for one polarization is within a tolerance value, and second polarization is marked as having broken detector, nothing is done and the antenna is removed from untuned list
  - otherwise the new pam settings are calculated as old pam + 0.9 * power delta
  - the new pam value is saturated between 0 and 63 dB and quantize (0.5 dB). If this is a first iteration and the detector value was “very low”, the saturation is between 10 and 63 dB

- if one polarization is marked as having a broken detector, the pax value from other polarization is copied
- pam values are sent to Pax Box
- if this is first iteration and at least one of detectors was "very low"
  * raw detection values are fetched for current antenna
  * if the detection value for respective polarization is still "very low" (despite change of pam settings), we mark the ant-pol as having a broken detector. The pam value of the other polarization is copied and applied. Warning is issued to stderr
- if both ant-pols for given antenna are marked as having broken detector, the default[5] pam settings are being applied. Antenna is then removed from untuned antenna list. Warning is issued to stderr

3.5 Database

The ataautotune uses two tables from the database (currently using obs-network database). The first table includes measurement values read from the pax bos measurement documents, source type, pax box serial number, polarization, polynomial coefficients and the range of polynomial data validity. That is, if the detector value (converted to dB) is not within [`lowdet`,`highdet`] range, the polynomial interpolation can't be reliably executed.

The iscoherent flag is set to 0 if the polynomials from different data source differs *"significantly"*

Note, meas0-20 means that there are 21 fields, meas0, meas1 . . . meas20

The "dummy" value was added for PB-000 as a default value.

```
pbmeas table:
+------------+--------------+------+-----+-------------------+-------+
| Field      | Type         | Null | Key | Default           | Extra |
+------------+--------------+------+-----+-------------------+-------+
| ts         | datetime     | YES  |     | CURRENT_TIMESTAMP |       |
| pax_box_sn | varchar(16)  | NO   | PRI | NULL              |       |
| pol        | varchar(2)   | NO   | PRI | NULL              |       |
| type       | varchar(3)   | NO   | PRI | NULL              |       |
| iscoherent | int(11)      | NO   |     | NULL              |       |
| meas0-20   | double       | YES  |     | NULL              |       |
| det0-20    | double       | YES  |     | NULL              |       |
| lowdet     | double       | NO   |     | -32               |       |
| highdet    | double       | NO   |     | -0.6              |       |
| p0         | double       | NO   |     | 0                 |       |
| p1         | double       | NO   |     | 1                 |       |
```

[5] values given by atasetpams without specifying the settings. It falls back to database that was created at least 20y ago

```
| p2-5          | double        | NO   |      | 0                   |         |
+---------------+---------------+------+------+---------------------+---------+
```

The second table feed_parts is used to match the antenna name with pax box number (with the assumption that Pax Boxes may be moved between antennas).

```
feed_parts table:
+-----------------+---------------+------+-----+-----------+---------+
| Field           | Type          | Null | Key | Default   | Extra   |
+-----------------+---------------+------+-----+-----------+---------+
| ant             | varchar(3)    | NO   | PRI | NULL      |         |
| control_box_sn  | varchar(16)   | YES  | UNI | NULL      |         |
| rim_box_box_sn  | varchar(16)   | YES  | UNI | NULL      |         |
| drive_box_sn    | varchar(16)   | YES  | UNI | NULL      |         |
| feed_sn         | varchar(16)   | YES  | UNI | NULL      |         |
| pax_box_sn      | varchar(16)   | YES  | UNI | NULL      |         |
| rfcb_sn         | varchar(16)   | YES  | UNI | NULL      |         |
| note            | varchar(256)  | YES  |     | NULL      |         |
+-----------------+---------------+------+-----+-----------+---------+
```

4 TODO

Probably more reliable (but slower) approach is to read N detector values (e.g. 3) and calculate average. That should mitigate power fluctuation while calculating the pam settings

Retire feed_parts table as a part of an effort to migrate array health/ata history do the database.