

Mission Control System based on Open MCT framework

MCSs in Physics and MINI-EUSO

Claudio Moroni

Relatore: Prof. Mario E. Bertaina (UniTo, Dipartimento di Fisica)

Correlatore: Ing. Rosario Messineo (ALTEC SpA)

July 12, 2019



Introduction

Nowadays many space-based physical experiments (like the upcoming MINI-EUSO), need realtime telemetry , thus becoming necessary for a physicist to understand how telemetry data i.e. measurement values are transmitted to Earth. A scarce knowledge in this sector could also lead to poor experiment optimization, insufficient communications between the engineering and physical teams of the mission or data loss in the worst case. Luckily there are many MCS solutions out there to fulfill everyone's needs. We explored and combined a bunch of these possibilities during the stage experience, and then turned to physical applications of them during the thesis period.

So the last part will be devoted to a data analysis fit into JEM-EUSO, the experiment this thesis experience ultimately showed to be well-fit for.

Mission Control Systems

The software we are looking for are Mission Control Systems (MCS), programs able to process and display telemetry, and even send commands to the payloads in some cases.

We investigated Open MCT, an open source, web-application based MCS.

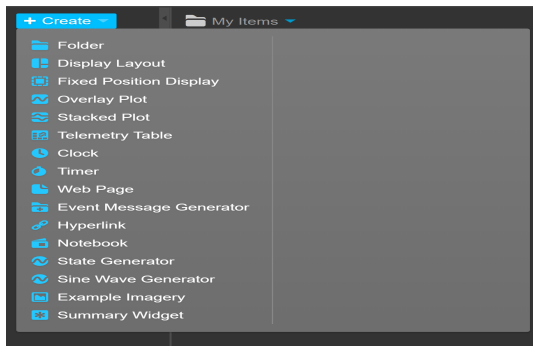


This tool allows to visualize historical and realtime data. Data could be ordinary floats (variously formatted), strings or even images (that could come from a optical payload on a rover).

Figure 1: Open MCT

Open MCT-2

Open MCT features some default functionalities which are pretty self-explanatory (like plot environments, general layouts in which drag-and-dropping simpler items, activity and timelines management, signal generators...).



Unfortunately, in order to do anything more complex (like displaying data), it is necessary to code. Unlike the previous Angular-based version, the newer release comes with user-friendly JavaScript internal API.

Figure 2: Default OpenMCT

Extending Open MCT

To provide user-cases and test our comprehension of the tool, an attempt to *extend* default plugins has been performed. The first success was obtained with the *function generator*, a plugin that aimed to improve the already existing *State* and *SineWave* generators.

The screenshot shows a dialog box titled "Create a New DataGenerator". At the top left, it says "All fields marked * are required". The dialog is split into two main sections: "PROPERTIES" and "LOCATION".

PROPERTIES:

- Title: DataGenerator (with a green checkmark)
- Notes: This is a sample signal Generator (with a green checkmark and a green 'G' icon to the right)
- function: x*Math (with a green checkmark)
- start: 0 (with a green checkmark and a small square icon to the right)
- stop: * 100 (with a yellow asterisk)
- interval: * -10000 (with a yellow asterisk)
- realInterval: * 1 (with a yellow asterisk)

LOCATION:

Save In: A dropdown menu showing "Style Guide Home" and "My Items".

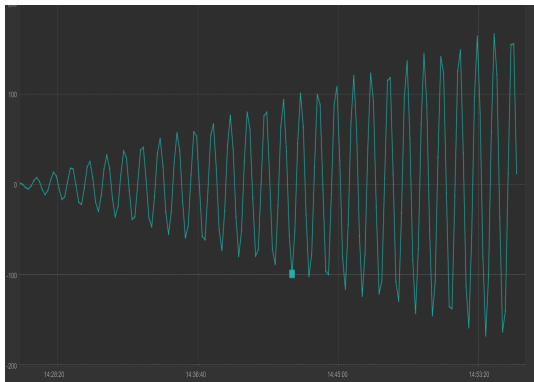
At the bottom right, there are "OK" and "Cancel" buttons.

Upon creation, the plugin shows a form in which one may set the function he/she wants to generate, its domain and machine precision.

Figure 3: Function Generator Form

Historical Function Generator

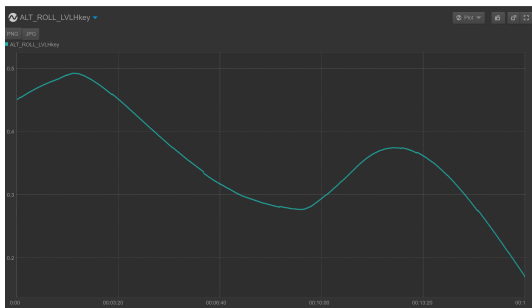
The first version of the Function Generator emulates historical telemetry: after the form is filled, all the pairs are generated and displayed. This simulates an access to a satellite data archive.



The result could be embedded in an *Overlay Plot* to get access to more advanced features. Now let's show realtime capabilities.

Figure 4: Historical Data Generator

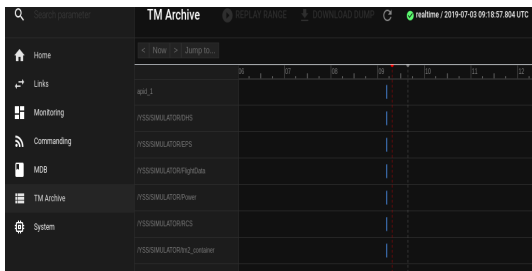
A further user-case confirmed Open MCT capabilities to retrieve data from archives. ISS position and attitude (with respect to LVLH and GTOD reference frames) data were given as .csv. A little code written by us interfaced Open MCT with the data:



This is the plot of an attitude parameter. This is one of the sets of data upon which the aforementioned data analysis has been operated.

Figure 5: ISS

Satellites telemetry data are usually generated in CCSDS. It is a standard for space operations. CCSDS is raw binary, meaning it has to be *interpreted*. This is accomplished by reading the CCSDS along with some tables, called *MIBs*. Unfortunately, Open MCT has no tool for doing so. One has to rely on a third-party plugin that interfaces Open MCT visualization capabilities with another MCS, Yamcs:

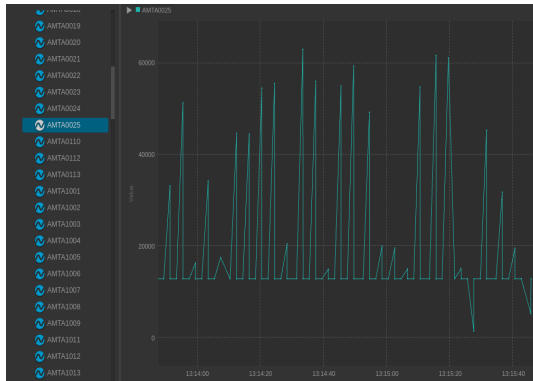


Yamcs is a heavier MCS compared to Open MCT. It is infact more complete, but perhaps lacks of the visualization capabilities of the latter.

Figure 6: Yamcs

The Chain

The CCSDS data is divided in *frames* that sent one by one to Yamcs using a small Java script. Yamcs has then been configured to process such data and distribute them to the server, to which Open MCT was connected as a client.



The resulting visualization. The parameter belongs to a dataset relative to IXV, a controlled reentry module for astronauts. The outcome of the stage activities is a report (50 pages) explaining how to technically achieve what has been shown.

Figure 7: Open MCT final visualization

Open MCT

Claudio Moroni

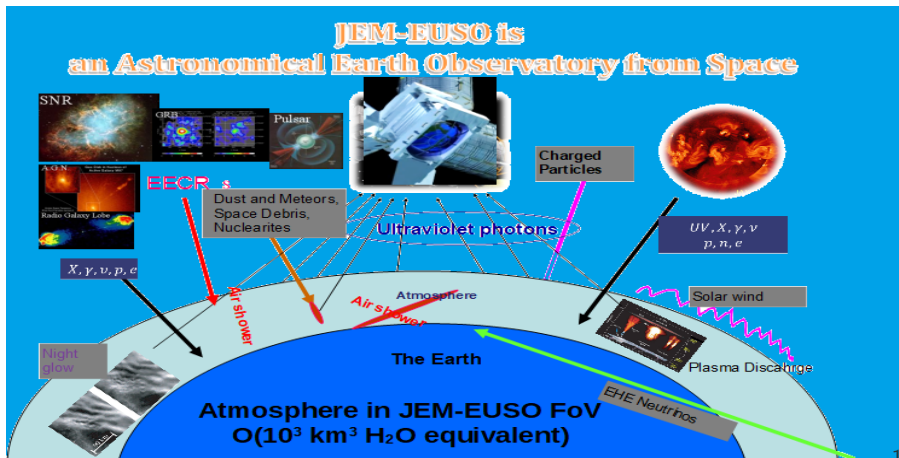
July 9, 2019

Contents

1 Prerequisites and Preamble	2
2 Command history	2
3 General Flow	2
4 Default Plugins	5
5 Extend Open MCT	6
5.1 The Bundle Way	6
5.2 The Plugin Way	6
5.2.1 Historical Function Generator Plugin	6
5.2.2 Realtime Function Generator Plugin	15
6 ISS plugin	17
7 Imagery	21
8 Time API	23
9 View API	26
10 Yames	26
10.1 Yames installation	26
10.2 IXV	27
10.3 SCOS2K	27
10.4 Yames Configuration	30
10.5 Open MCT plugin installation and Configuration	41
10.6 Yames Client	47
10.7 Open Issues	48

JEM-EUSO-1

The physical applications of what we learned has been found in the JEM-EUSO project. This program aims to put into orbit a telescope capable of detecting Ultra High Energy Cosmic Rays (UHECR) showers.



JEM-EUSO-2

This program is composed of many path-finder experiments, next one being MINI-EUSO.

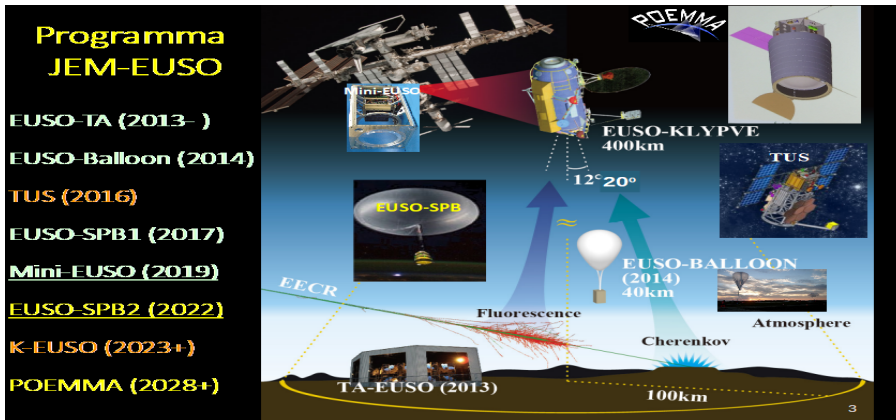
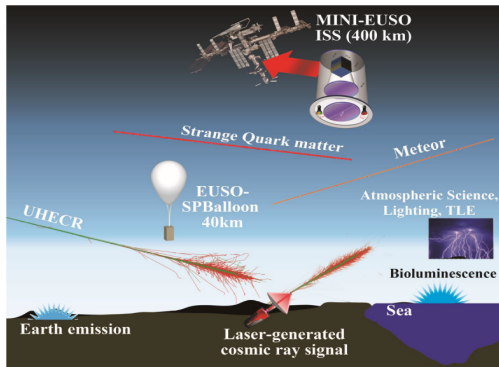


Figure 8: JEM-EUSO Program

MINI-EUSO-1

MINI-EUSO is a telescope that will measure night sky UV emission. It will be also triggered by meteors and laser-generated CR shower-like events.

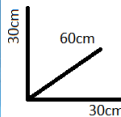
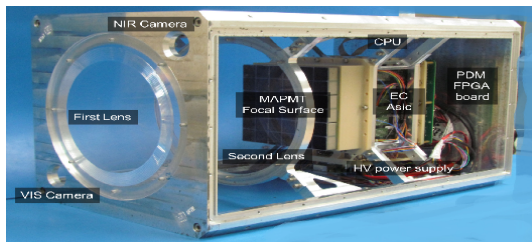


MINI-EUSO will be placed on board the ISS, on the russian module behind a UV-transparent window. Its purpose is to let us tell the origin of an observed UV peak (e.g. whether its origin is a UHECR shower or a city)

Figure 9: MINI-EUSO

MINI-EUSO-2

Its main apparatus is a 48×48 pixel matrix, each having a sight amplitude of 0.8° , giving a total of 22° (also taking into account inter-pixel photomultipliers dimensions).



In order to know the sky surface MINI-EUSO is currently looking, it is necessary to have realtime feedback of ISS altitude: that's where Open MCT comes into play! A similar issue concerns ISS oscillations.





Figure 11: Astronaut Luca Parmitano with MINI-EUSO

Question

We aimed to quantitatively estimate ISS oscillations and drag-induced changes in altitude, and demonstrate that we will be able to assess whether they should be taken into account during MINI-EUSO data analysis.



Figure 12: MINI-EUSO simulated FOV

Being in possession of a ISS attitude dataset, a data analysis has been performed.

This analysis is a proof of concept of what should be eventually done when MINI-EUSO will be working.

Anyway, Open MCT (and MCS in general), are capable of processing and realtime plotting, but little do they know about data analysis: so we used R.

The first concern was getting informations about the altitude. We had data consisting of altitude and attitude of the ISS every second.

The following plot represents the mean altitude of the ISS each day, from mid 2011 to early 2014.

Altitude Plot

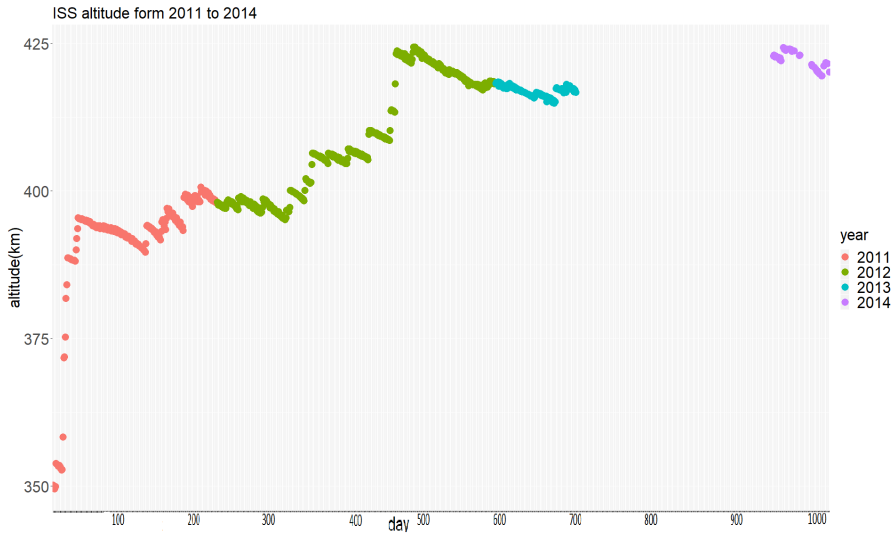


Figure 13: Altitude

Regressions-Intro

One may notice that the ISS has been gradually lift up, roughly by 75 km, over those three years.

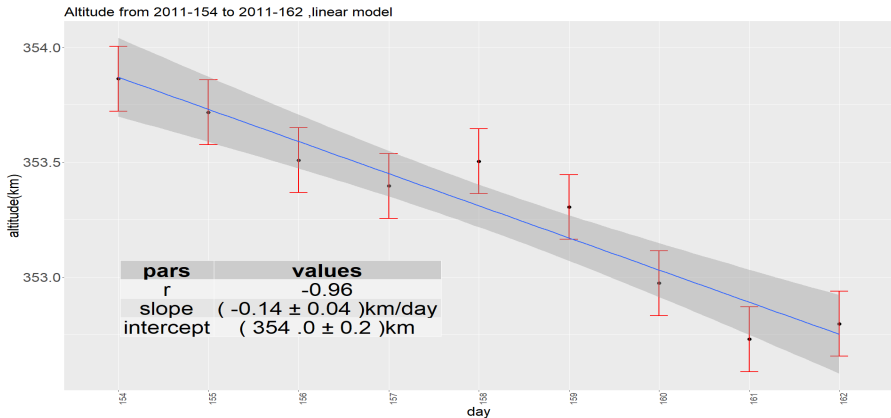
It would be interesting to know how much the ISS descends during those descending patterns, and see if we could ascribe them to atmospheric drag.

We will be focusing on some of those descending patterns, and fit them with straight lines and exponentials.

We immediately show an example of the resulting linearized plot.

Regressions-A posteriori

To deal with the high χ^2 , and because we weren't given the instrument errors, we decided to assign them *a posteriori*, via $\sigma_y^2 = \frac{\sum_{i=1}^N |a + bx_i - y_i|^2}{N-2}$ after evaluating linear correlation.



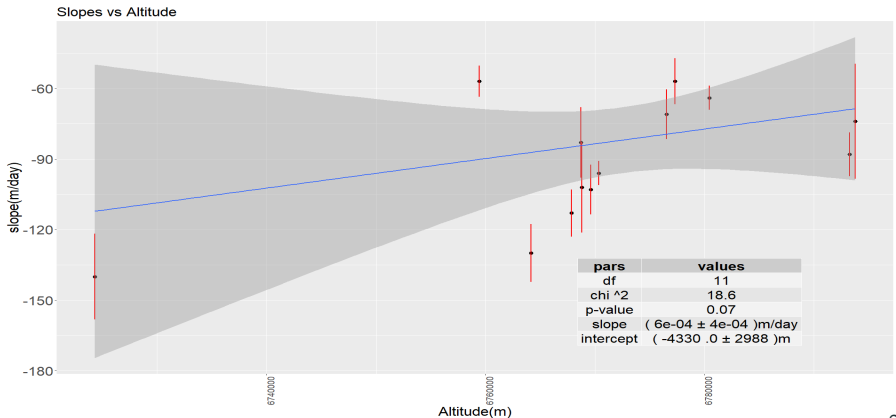
Regressions-Summary

The table with all the *a posteriori* values we retrieved:

Period	r	slope($\frac{m}{day}$)	$\sigma_{slope}(\frac{m}{day})$	intercept (m)	$\sigma_{intercept}(m)$
11-154~11-162	-0.96	-140	36	6725000	200
11-169~11-179	-0.97	-60	13	6759800	90
11-273~11-291	-0.98	-130	24	6765500	280
11-344~12-012	-0.97	-100	21	6771000	425
12-014~12-028	-0.93	-100	38	6767000	350
12-029~12-059	-0.93	-83	30	6770000	550
12-061~12-090	-0.98	-110	20	6769600	360
12-097~12-115	-0.99	-96	10	6771300	120
12-147~12-171	-0.95	-70	21	6777500	320
12-173~12-199	-0.94	-60	19	6778100	310
12-201~12-227	-0.98	-64	10	6781350	160
12-236~12-257	-0.84	-74	49	6794600	640
12-258~12-300	-0.97	-88	19	6795200	470

Slope vs Altitude

To test whether a 75 km change in altitude (from about 350km) determines a change in the drag, we fit the slope vs (mean) altitude for each period with a line, and check if the angular coefficient is compatible with zero.



Altitude Analysis Conclusions

Thus we get:

$$z = \frac{6 * 10^{-4} - 0}{0.0004} = 1.5$$

So we are not able to distinguish the slopes, thus we can take the weighted mean:

	Weighted mean	Weighted Error
slope	$-82 \frac{m}{day}$	$-5 \frac{m}{day}$

To give a rough estimate, let's say the ISS is 400km in altitude. Then a MINI-EUSO pixel would observe a

$$(2 * 400.000km * \sin(0.4^\circ))^2 = 31.192km^2$$

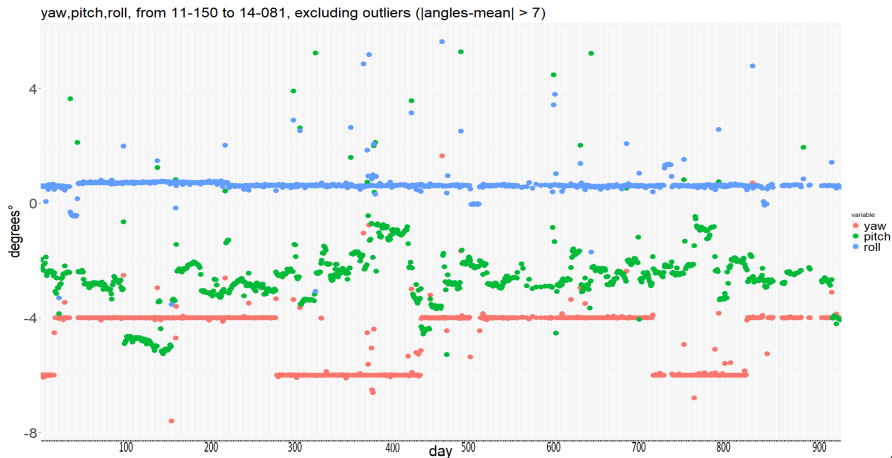
area worth surface. After one day, the new ground FOV would be

$$(2 * (400.000 - 0.082)km * \sin(0.4^\circ))^2 = (31.064 \pm 0.008)km^2$$

So the observed area would roughly reduce by $0.130km^2$ each day.

Euler angles-Intro

Concerning angles, we are interested in oscillations along the three axis: yaw, pitch and roll. Around 98% of points are in the $[-7, 7]$ interval around the mean.



Euler-Analysis

Because different experiments have different resolutions, we wrote an algorithm capable of determining the mean angle variations, taking into account only variations which time span is user-defined.

We asked it to evaluate mean variations over different periods, for the yaw, pitch and roll angles:

Table 1: YAW

Duration	mean	σ
1 sec	0.000 12°	0.000 26°
1 min	0.004°	0.003°
45min	0.11°	0.06°

Tables

Table 2: PITCH

Duration	mean	σ
1 sec	0.001 $^{\circ}$	0.005 $^{\circ}$
1 min	0.012 $^{\circ}$	0.007 $^{\circ}$
45min	0.04 $^{\circ}$	0.03 $^{\circ}$

Table 3: ROLL

Duration	mean	σ
1 sec	0.0004 $^{\circ}$	0.0007 $^{\circ}$
1 min	0.02 $^{\circ}$	0.01 $^{\circ}$
45min	0.6 $^{\circ}$	0.3 $^{\circ}$

What's next?

In conclusion, we have significant angle variations during the orbital period (which may be of interest for the debris deorbiting project).

Concerning the altitude, the qualitative result is that approximately between 350 and 425 km our analysis couldn't detect a change in drag intensity, and the quantitative result about altitude loss per day due to drag, $(82 \pm 5) \frac{m}{day}$

Many other plots, more detailed but less illustrative, have been drawn, like the following half-day, one point per second Yaw chart:

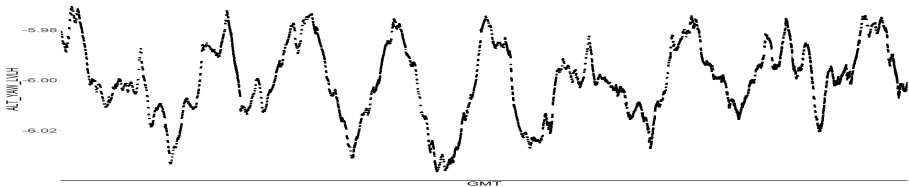


Figure 14: 2013-287

To extract more data from this behavior, one maybe needs a machine learning algorithm based on Fourier series, in order to find repeating patterns that may lead to a predictability of ISS attitude evolution in response to docking/undocking maneuvers.

Such a determination could be then implemented, as a correction factor, in the software that points the telescope and later in the one that points the laser.

BACKUP SLIDES

The Auger experiment evidenced an anisotropy in UHECR arrival direction:

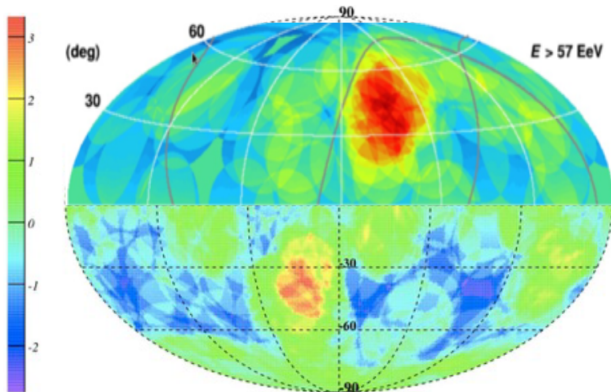


Figure 15: Auger Anisotropy detection

Observation

In the sky locations where these UHECRs excesses have been detected coming from, some star-burst galaxies were found, like M82



Which is thought to be responsible for the northern-hemisphere anomaly.

Figure 16: M82 starburst galaxy

Quantitative Estimate

The objective is thus to demonstrate (or falsify) that starburst galaxies are primary sources of UHECRs. In order to do it, we need to gather enough statistic.

Problem: Nowadays observations estimate a UHECRs flux which ranges from $\frac{1CR}{km^2century}$ to $\frac{1CR}{km^2millennium}$, depending on energy ($10^{19} - 10^{20} eV$).

This is a very little flux, even considering current technology.

State of the Art

One of the largest current UHECR detectors is Auger:



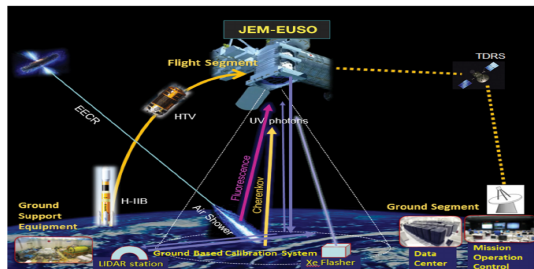
Has a surface as wide as 3000km^2 , which would account for a too little number of detected UHECRs over a reasonable span of time.

Figure 17: One of the Auger facilities

Solution

We cannot tune the time factor, but there is a way to extend the observed sky: launch the telescope into orbit and let it look down on Earth, looking for showers products .

The program that aims to develop technology to build and launch such a telescope is the JEM-EUSO program, which is an italian-japanese-american-russian collaboration.



This program is composed of many path-finder experiment, one of which is MINI-EUSO.

Figure 18: JEM-EUSO

Another Experiment

There is a concurrent project that aims to use a JEM-EUSO class telescope in combination with a laser to localize, track and de-orbit space debris with a size that varies from 1cm to 10cm.

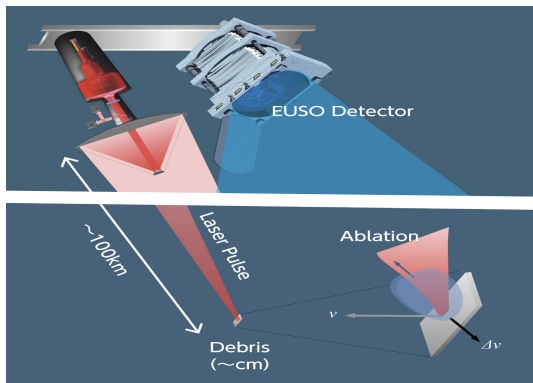
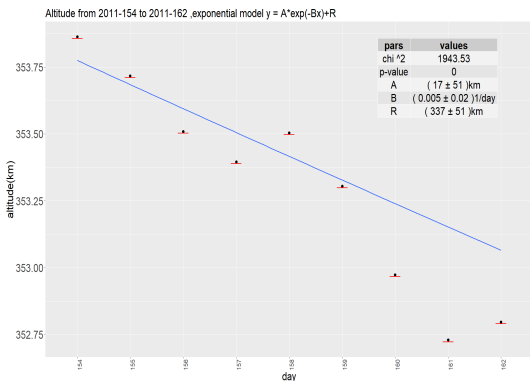


Figure 19: SpaceDebris

The telescope would detect and follow the debris for a time sufficient to the electronics to predict where the debris will be. The prediction needs to account for the time the laser takes to rotate. The laser would then de-orbit the debris via radiation pressure.

Regressions-Exponential

Assuming a residual viscous drag, we show the fit of the first descending pattern with $y = Ae^{(-Bx)} + R$. We don't take ISS shape into account, and $B = k/m$ is found from the fit rather than read from tables.



Because of the very low B , we linearized all of the other fits. High values of χ^2 with respect to low DF (7 in this case) are later taken into account.

Figure 20: Exponential Fit

Regressions-Linear

Linearizing, we obtain (the gray part is the confidence interval, and the errors are equal to the standard deviation of each day, assuming that the altitude is a daily constant):

