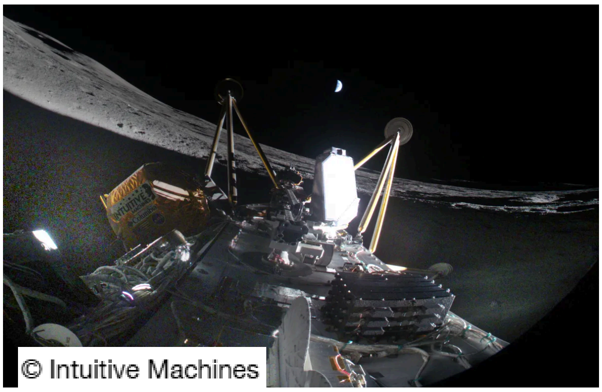
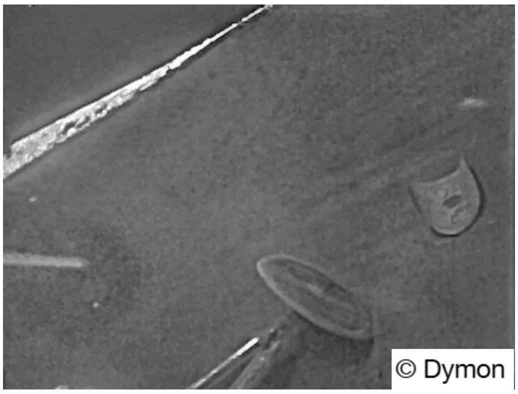
Analysis of the Data from the YAOKI Rover on the IM-2 Mission to the Lunar South Pole





(left) Processed image from the IM-2 lander camera from within the dark crater near the South Pole of the Moon, showing the Earth above the horizon. © Intuitive Machines
(right) Processed image from the YAOKI rover camera showing the broken lander leg and bright sunlit rim of the crater. © Dymon

Summary: This document provides the analysis of the data from the YAOKI rover on Intuitive Machine's IM-2 mission. It focuses on correlating the telecommands sent and the telemetry received to showcase the technology validation of the Dymon rover and the JAOPS command and control system in the lunar surface environment. This analysis emphasizes lessons learned to be implemented in the next missions.

Prepared by: Louis Burtz
Approved by: Alejandro Sela, Shinichiro Nakajima
Release date: 2026/06/05

Contributors	
Name	**Email**
Louis Burtz	ljburtz@jaops.com
Alejandro Sela	asela@jaops.com
Shinji Miyaki	shinji.miyaki@jaops.com

Document Change Log		
Version	Date	Comments
1.0	2025/05/30	Initial Release
2.0	2025/06/05	For Public Release
2.1	2025/06/26	Included Review from Open Lunar

Table of Contents

1 Introduction

The Dymon YAOKI rover aboard the Intuitive Machines IM-2 mission landed on the South Pole region of the Moon on March 6th, 2025. The YAOKI rover is a 500 gram two-wheeled rover sporting a camera and sensors to demonstrate micro-rover technology for lunar applications, including collecting in-situ data on the lunar surface environment and providing third person perspective imagery of the lander and other surface assets.

Unfortunately, the IM-2 lander touched down into a dark crater and came to rest on its side. Due to the lack of solar power generation and the cold environment, the mission duration was reduced to just 12 hours. Due to the extreme cold and lack of power availability from the lander the rover deployer did not open and YAOKI could not drop down to the lunar surface.

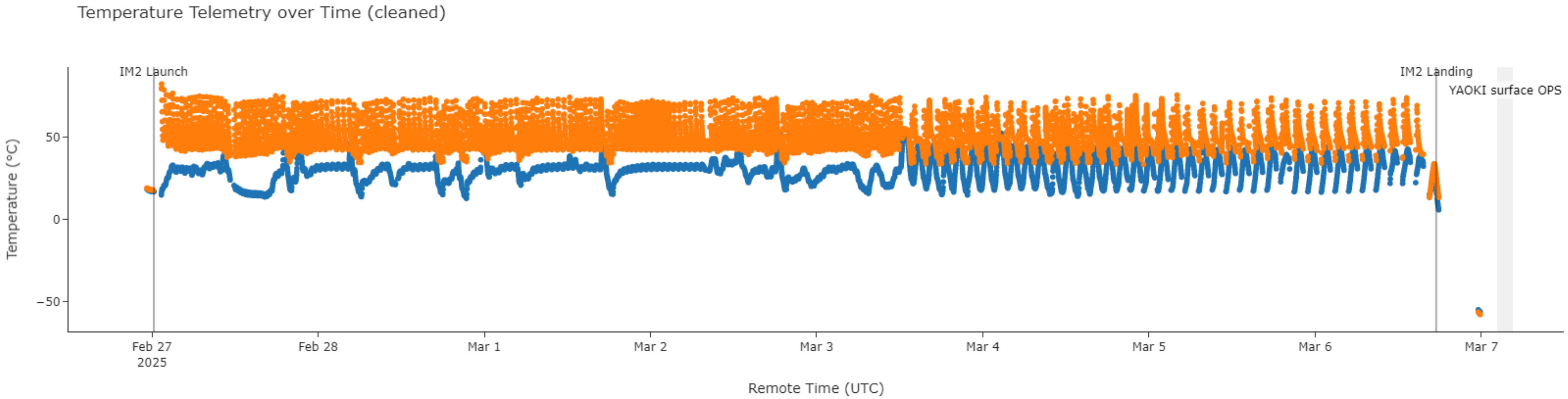
Nevertheless, the YAOKI rover was operated for ~2 hours and YAOKI operators carefully tested all possible subsystems and command configurations to raise the technology readiness of the YAOKI platform. In total, 73 commands were sent, 28 images were downlinked, and more than 2 hours of continuous telemetry data were acquired.

This document focuses on correlating the telecommands sent and the telemetry received to showcase the technology validation of the Dymon rover and the JAOPS command and control system in the lunar surface environment. First the data received during launch, cruise and landing is analyzed. Then the main focus is on the surface operations, with a deep dive into the telecommands, the temperature data, the radio data, the camera data, the accelerometer data and the status bits data. Throughout the document, the emphasis is on lessons learned to be implemented in the next missions.

All analysis is made by accessing the Yamcs database via its Python API. The data analysis jupyter notebooks are made available as part of this public data release.

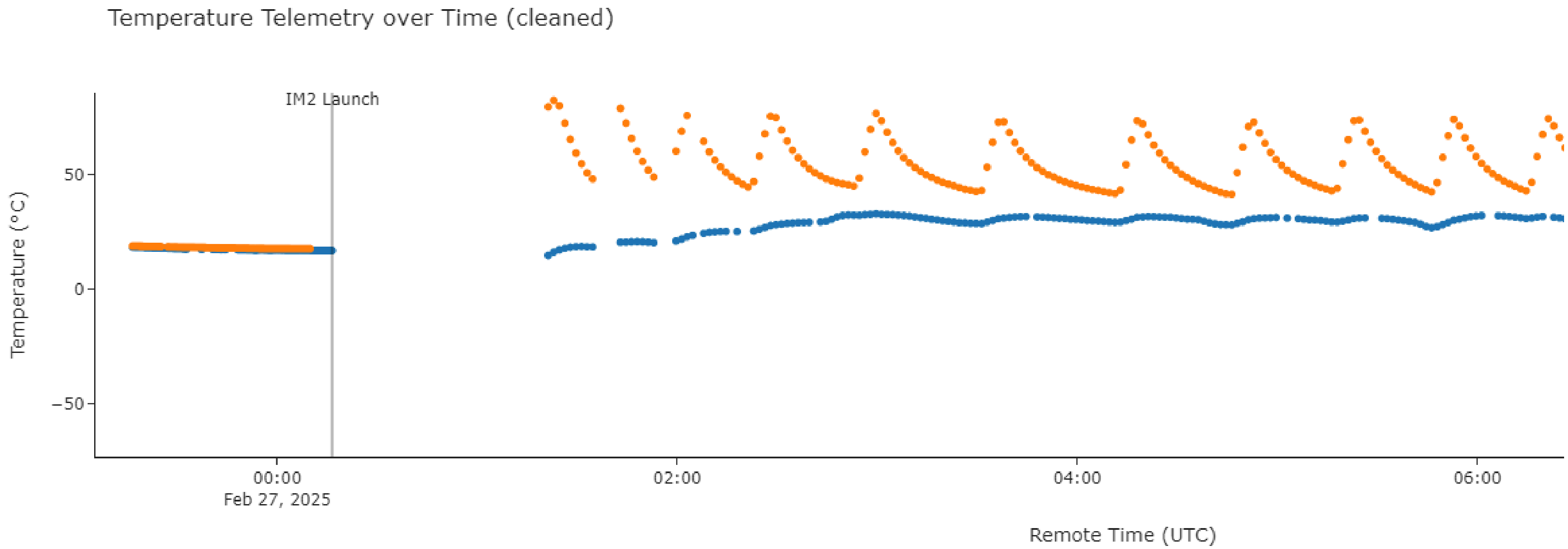
2 Launch, Cruise and Landing

During launch, cruise and landing mission phases, the YAOKI rover is powered OFF. Intuitive Machines provides a few lander-side telemetry items that are relevant to understand the environment of the YAOKI rover within its deployer. The figure below shows the temperature data from two temperature sensors during the 8 day cruise from the Earth to the Moon and the 12 hours on the lunar surface.
Blue is the temperature measured outside the deployer, near the burnwire deployment mechanism.
Orange is the temperature measured outside the deployer, near the lander mounting bracket.

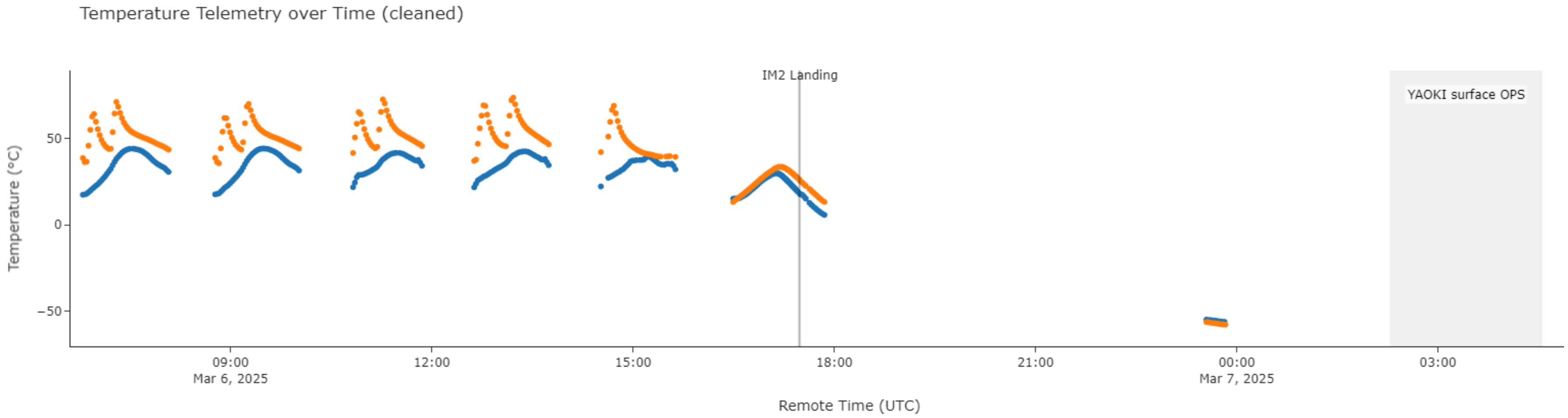


Note the different temperature environments during the different phases of flight:

- Before Feb 27 : room temperature, lander is integrated within the rocket fairing, on the launch pad
- Feb 27th to March 3rd: cruise from Earth to the Moon, temperatures depend on the orientation of the lander w.r.t the Sun.
- March 4 to March 6th: lunar orbit: note the ~2 hour cycles as the lander goes from sunlight to eclipse behind the Moon
- March 7th: landing and surface operations: the temperature drops to -60 degC in the dark crater



The graph above shows the YAOKI rover temperature environment around launch. At first, the temperature is controlled on the ground (around 20 degC), then the rover is exposed to the initial environment in space after separation from the rocket. Note the periodical oscillations, most probably due to the slow rotation of the lander leading to the YAOKI rover being in the sunlight (temperatures up to +80 degC) and in the shadow (temperatures down to 30 degC)



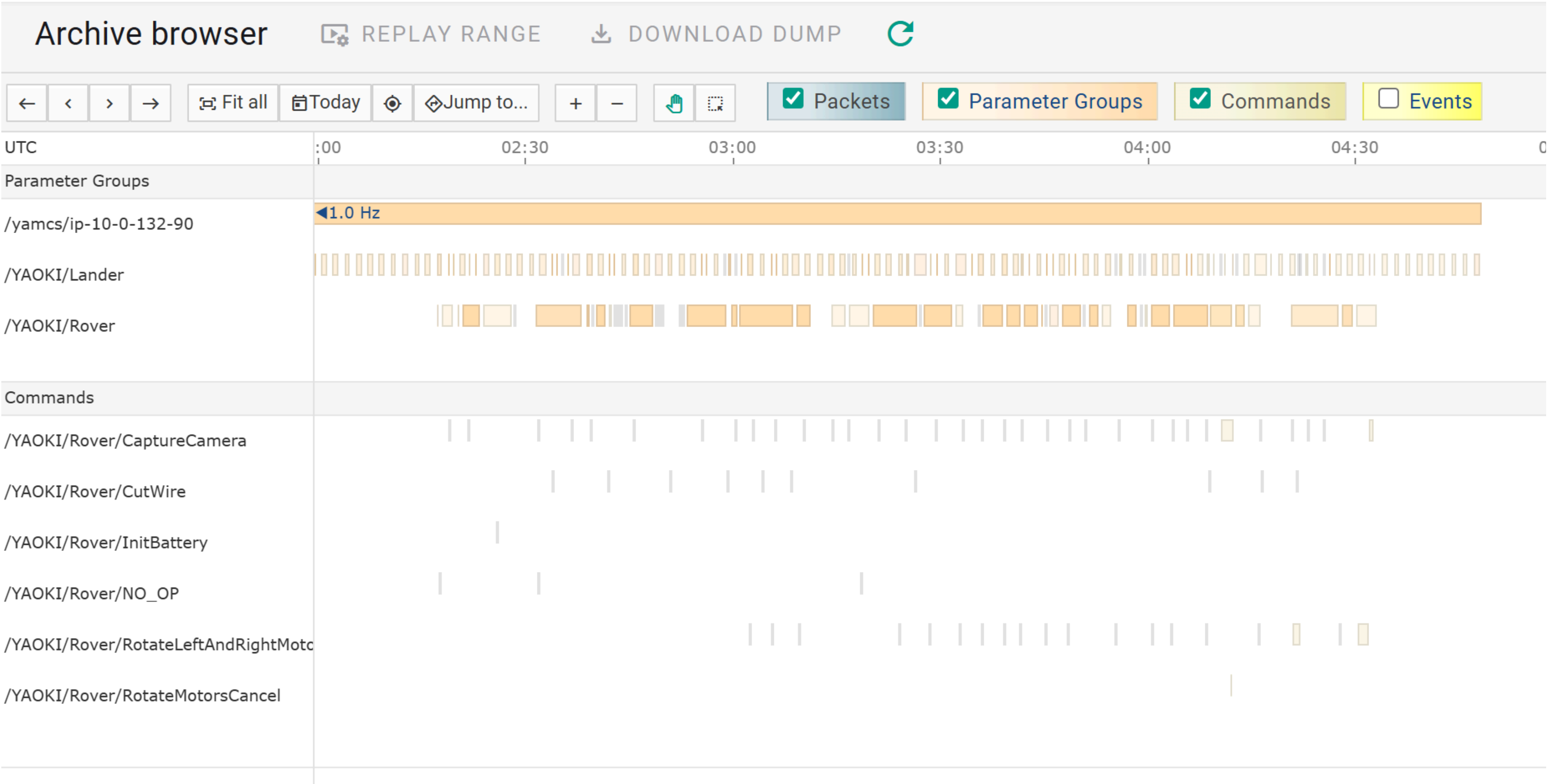
The graph above shows the temperature environment during the last five orbits around the Moon, then the landing, then surface operations. Note that soon after landing, the thermal system of the lander was powered OFF to conserve electrical power onboard the lander (no data). A little before midnight UTC on March 7th, the lander thermal subsystem was briefly powered ON again to assess the thermal environment: temperatures in the dark crater are still dropping, now around -60 degC.

The YAOKI operations started at 02:17 am on March 7th: the temperatures can be estimated to be around -70 degC at the sensor locations outside the YAOKI deployer. (see section 3.2 for rover surface operations temperature analysis)

This space environment data contributes to raising the technology readiness level of the rover and provides insights for the design of the next missions.

3 YAOKI Surface Operations

The Yamcs archive browser provides a convenient overview of all the telemetry received from and all the telecommands sent to the YAOKI rover:



The following times (in UTC timezone) are given for the key events of the YAOKI surface operations:

IM2 Landing	2025/03/06 17:28:50
YAOKI first housekeeping telemetry received	2025/03/07 02:17:15
YAOKI first telecommand (NO_OP command)	2025-03-07 02:17:51
YAOKI last telecommand (CaptureCamera command)	2025-03-07 04:32:40
YAOKI last image packet received (Intuitive Machines turned off the YAOKI communications link)	2025/03/07 04:33:04
IM2 lander last telemetry received	2025/03/07 05:42:56

YAOKI lunar surface operation duration: 2 hours 15 minutes 49 seconds

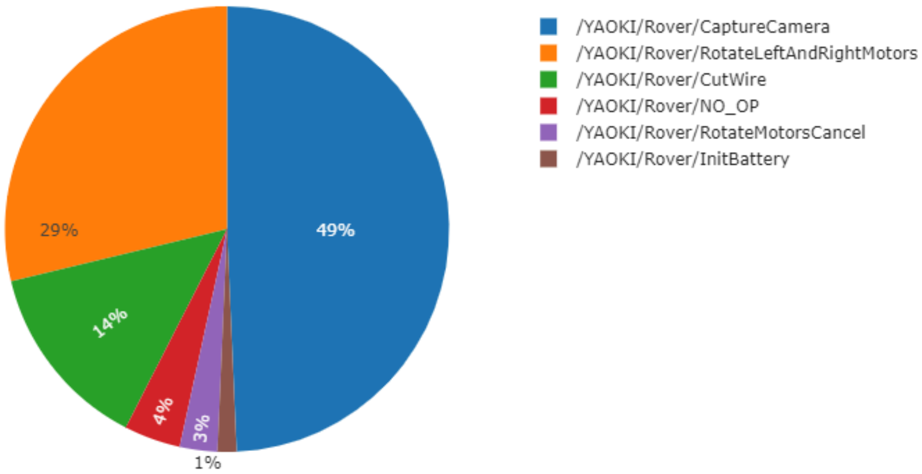
Final Resting Site: 84.79059° S, 29.19567° E, 5322 m elevation

LROC Quickmap view: https://bit.ly/4knPgcq

3.1 Telecommand Analysis

In total, 73 commands were sent to the YAOKI rover during surface operations. Most of the telecommands were for taking pictures and driving the rover.

Command Name	Count
/YAOKI/Rover/CaptureCamera	36
/YAOKI/Rover/RotateLeftAndRightMotors	21
/YAOKI/Rover/CutWire	10
/YAOKI/Rover/NO_OP	3
/YAOKI/Rover/RotateMotorsCancel	2
/YAOKI/Rover/InitBattery	1
Total	73



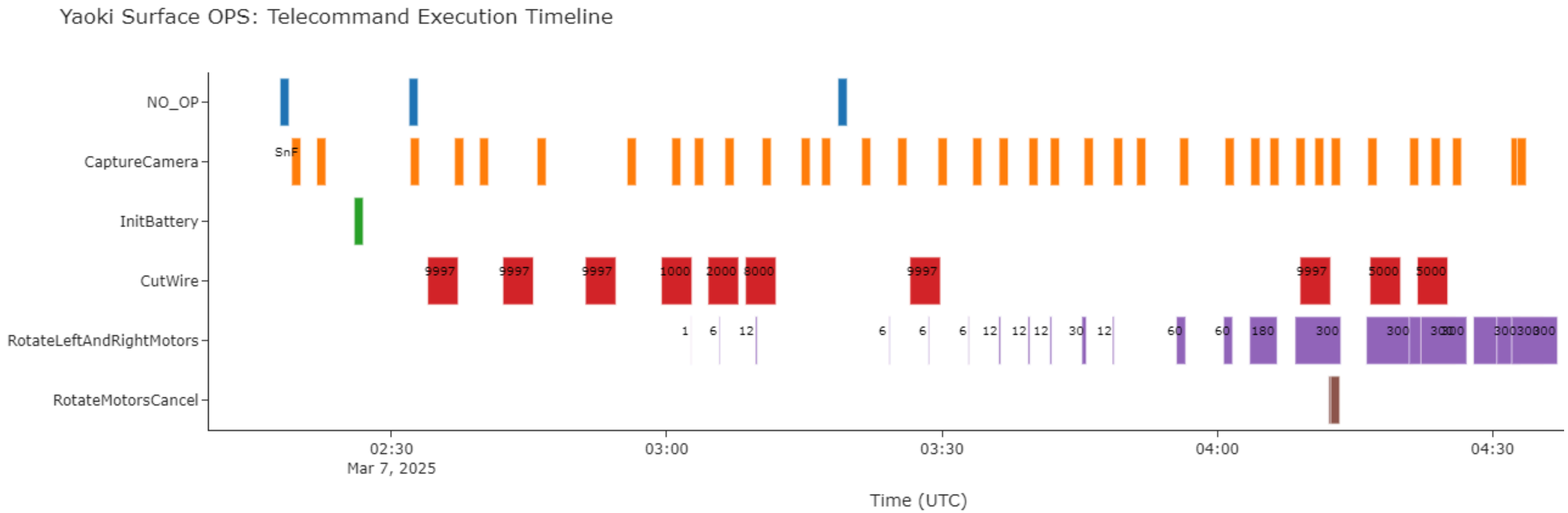
The figure below shows a timeline of each telecommand sent. The planned operating procedure was followed:

- First a NO_OP command to test the command uplink and verify from the telemetry that the TC was accepted
- Take a picture via the Store and Forward Channel (labelled as SnF in the figure below)
- Take a picture via the Streaming channel (nominal, no label)
- Change the rover to internal battery power (see discussion in section 3.4 this command did not reach the rover)
- Take another picture
- Cut the wire
- Take a picture to confirm deployment

Since initial deployment was not successful, the operations moved to the contingency operations plan:

- Repeat the CutWire and CaptureCamera commands

- After a while without success: start sending rover drive commands (Rotate left and right motors). First small movements then larger movements to try and break free from the deployer (see labels the number of seconds of motor rotation).
- After each drive command, take a picture to check for any rover movement.

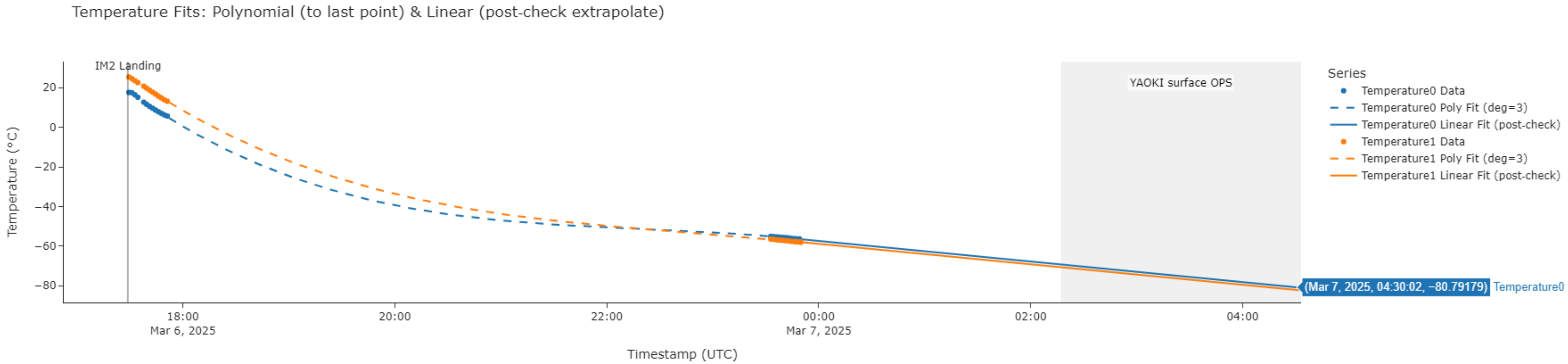


Note: the time argument given to the CutWire command is printed on top of the red bars. It should indicate the number of seconds that the CutWire command is running and that the Status_ICUT should be HIGH. However, there was a bug in the Yaoki flight software and the CutWire command timed out after about 3 to 5 minutes.

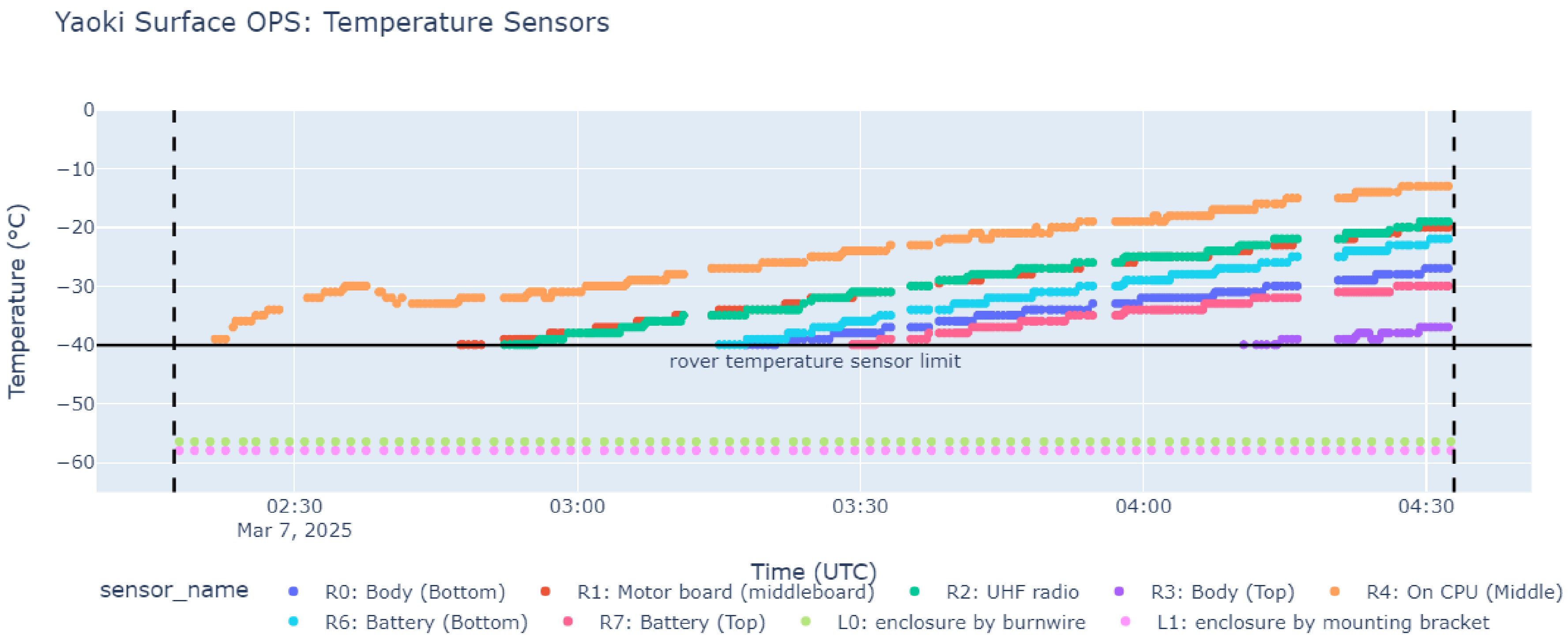
3.2 Temperature Data Analysis

The temperature data from the lander is missing for most of the surface operations because the IM operators turned off their thermal subsystem to conserve lander battery power. By fitting and extrapolating the available datapoints, the temperature environment of the YAOKI rover prior to and during surface operations can be estimated.

The graph below shows the estimated (linear fit from just a few minutes of data) temperature environment outside the YAOKI deployer. The result is very uncertain due to the lack of data and time between the last datapoint, but we can estimate the temperatures to be between - 65 and - 80 degC at the end of the YAOKI operations.



The graph below shows the evolution of the temperature inside the YAOKI rover during surface operations.



Note that due to an issue in the YAOKI Flight Software, the temperature values below -40 degC are clipped. From the similar trend across all seven sensors, it is likely that the temperature **inside** the rover at the beginning of the surface operations was below -70 degC.

As expected, the CPU temperature increases the fastest and reaches the highest temperature: -12 degC after 2 hours 15 minutes of operation. The coldest area is at the top of the rover body which always stays below -35 degC.
The two temperature sensors from the lander measuring temperature **outside** the deployer last indicated temperatures around -60 degC (at midnight UTC, 2.5 hours before the start of YAOKI operations) and therefore likely colder than -70 degC from that point of view as well.

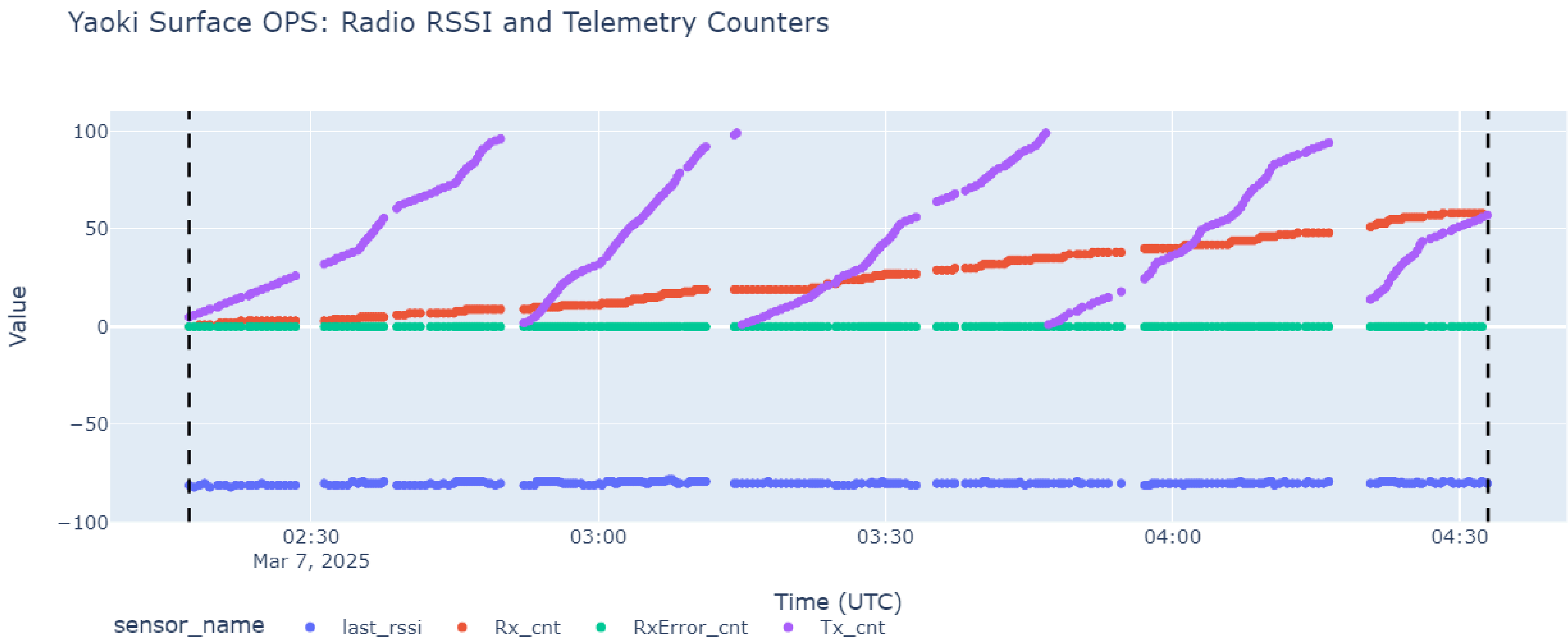
In these circumstances, it is especially successful that all rover electronics, especially the battery and camera and CPU and radio operated so well at such low temperatures. It is especially unfortunate that the raw temperature data was not sent (lessons learned for future missions!).

The table below provides more information on the position of each sensor inside the YAOKI rover:

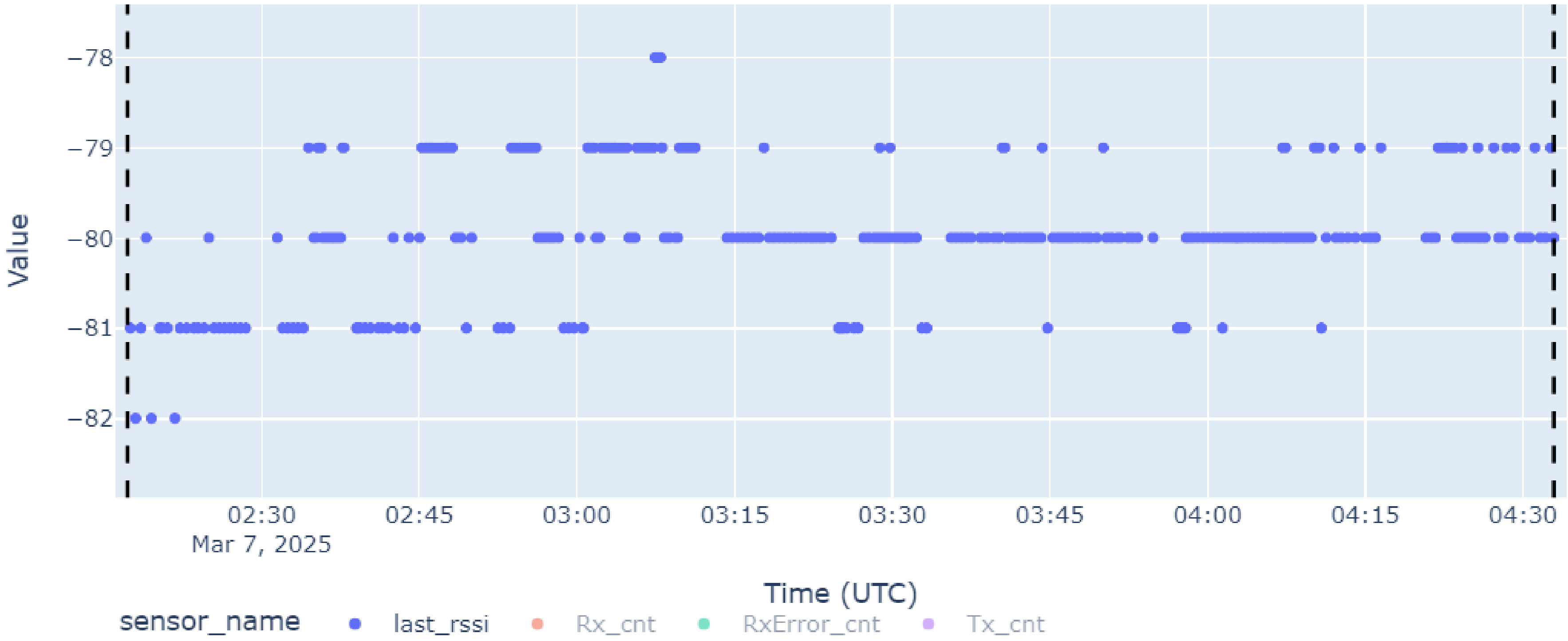
R0: Body - back side of board (Bottom)	ボディ-基板裏側
R1: Motor board (transistor) (middleboard)	モーター基板(トランジスタ)
R2: UHF radio	無線機
R3: Body - back side of radio (Top)	ボディ-無線機裏側
R4: Main board (on CPU) middle board	メイン基板（CPU上）
R5: [temperature data not sent]	N/A
R6: Battery - Bottom side (viewed from the board)	電池-下側（基板から見て）
R7: Battery - Top side (viewed from the board)	電池-上側（基板から見て）

3.3 Radio Data Analysis

As shown in the figures below (and the camera data transmission analysis in the next section), the radio subsystem between the rover and the lander performed very well, with a very high level of transmission reliability.



The figure below shows that RSSI values are constant, within measurement accuracy, with no discernable increase or decrease. No correlation with deployment or movement commands.



Note two outlier values (the two outliers are part of a single outlier event: the two values are one after the other):

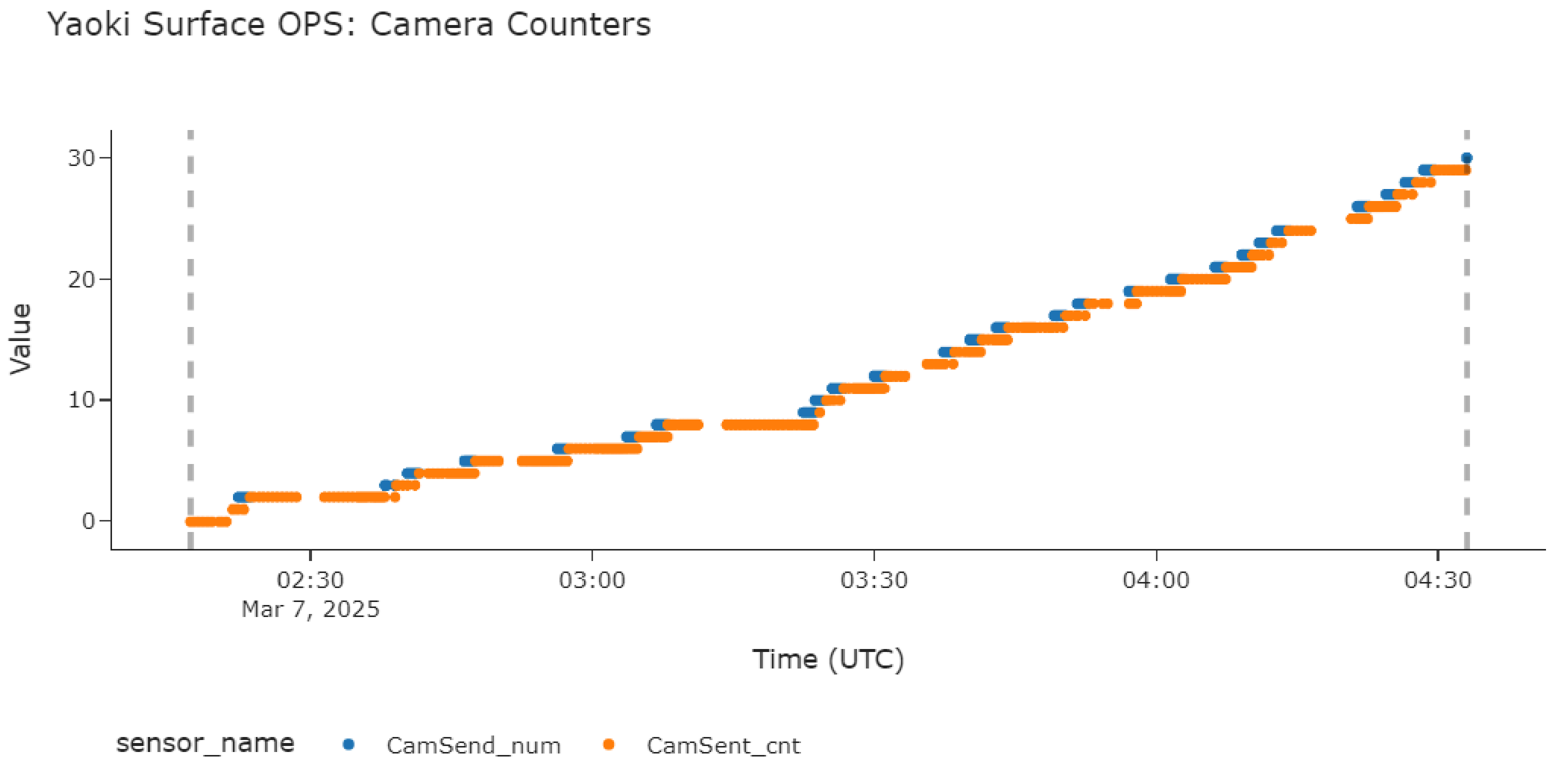
	sensor_name	timestamp	value
205	last_rssi	2025-03-07 03:27:58.127000+00:00	9999
206	last_rssi	2025-03-07 03:28:08.284000+00:00	9999

Furthermore, as shown in the Status Bits section, the NORX status is always low: this means that there never was a period of time longer than three minutes where lander UHF radio pings were not received by the rover UHF radio.

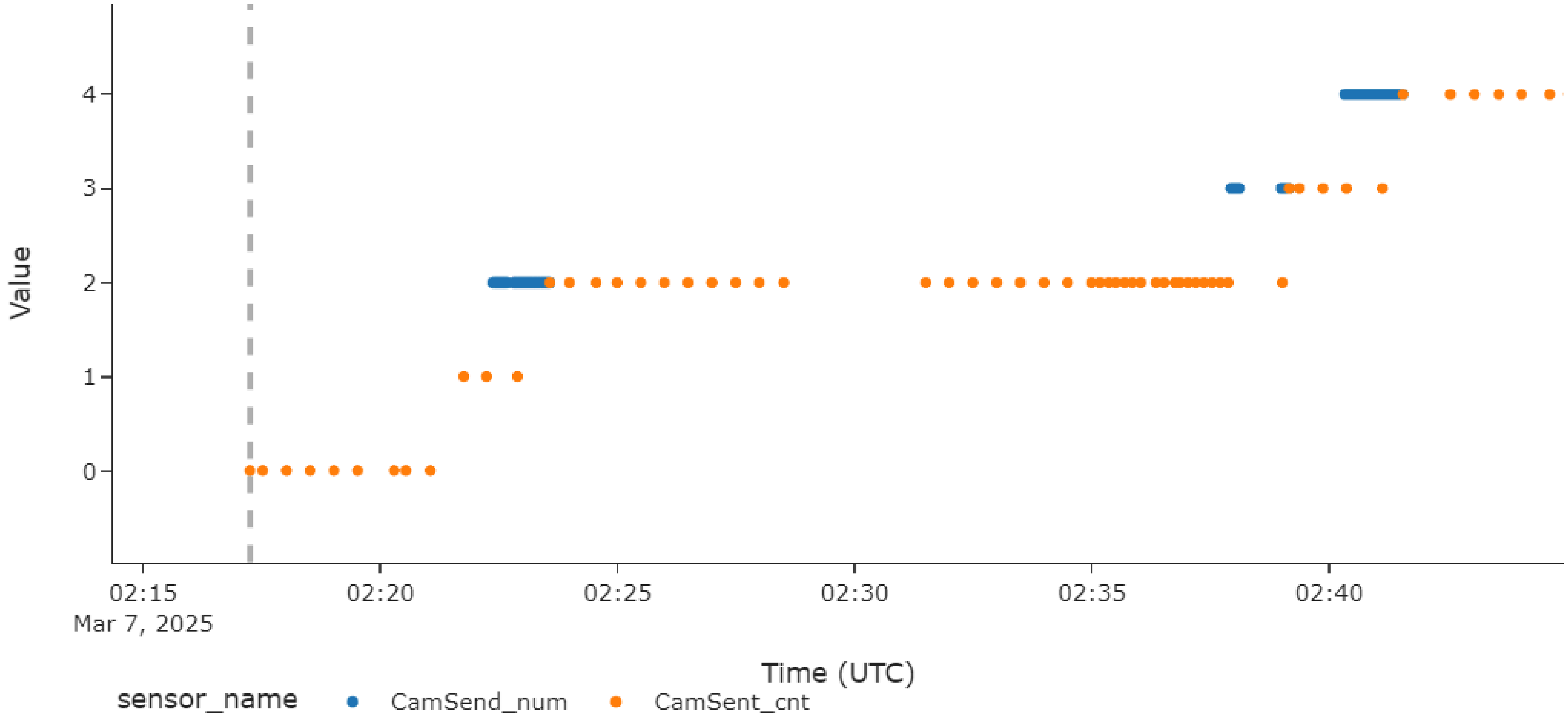
3.4 Camera Data Analysis

30 images captured by the rover, 28 images received on the ground:

- 1 image received via store and forward (downlinked via IM2 storage and retrieved via NovaCore REST API)
- 27 images received via the streaming channel.
- For images #13 and #25, zero packets were received, but the YAOKI flight software incremented the image counter.
- For the last image (image #30), only the first 33 (out of 960) packets were received before Intuitive Machines disconnected the YAOKI streaming channel (this is an expected transmission cut, no transmission error there)



CamSent_cnt is the housekeeping telemetry counter (received every ~30 seconds)
CamSend_num is extracted from the header of streamed image packets (only received when an image is downloaded)



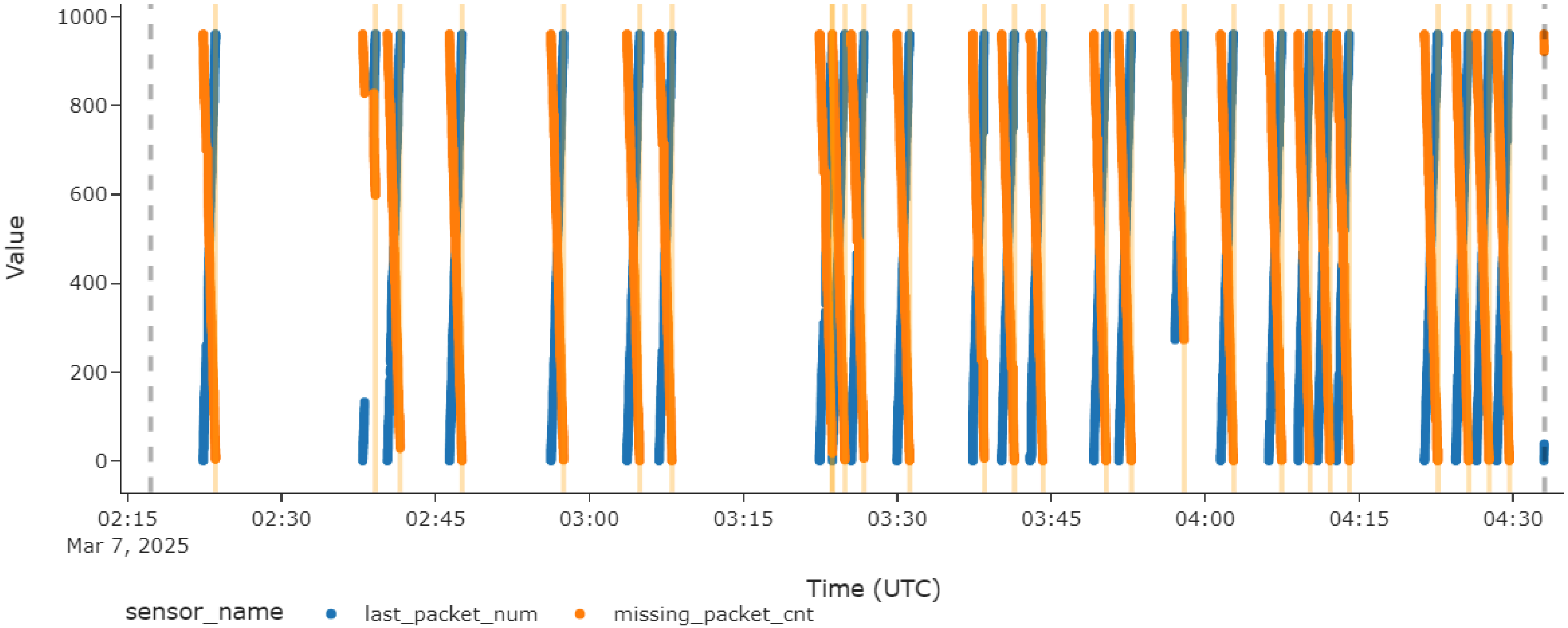
Note :

- Image #1 was sent via store and forward channel -> no streamed image packets : only the housekeeping telemetry (orange) increments.
- Housekeeping telemetry counter only increments once the image has been fully transmitted

Image Counter	CamSent_cnt First Timestamp	CamSend_num First Timestamp
0	2025-03-07 02:17:15.150000+00:00	NaT
1	2025-03-07 02:21:45.615000+00:00	NaT
2	2025-03-07 02:23:34.408000+00:00	2025-03-07 02:22:22.497000+00:00
3	2025-03-07 02:39:09.399000+00:00	2025-03-07 02:37:55.901000+00:00
4	2025-03-07 02:41:33.197000+00:00	2025-03-07 02:40:19.396000+00:00
5	2025-03-07 02:47:36.171000+00:00	2025-03-07 02:46:22.171000+00:00
6	2025-03-07 02:57:30.278000+00:00	2025-03-07 02:56:16.163000+00:00
7	2025-03-07 03:04:58.329000+00:00	2025-03-07 03:03:41.362000+00:00
8	2025-03-07 03:08:03.309000+00:00	2025-03-07 03:06:48.166000+00:00
9	2025-03-07 03:24:12.407000+00:00	2025-03-07 03:22:27.636000+00:00
10	2025-03-07 03:24:54.204000+00:00	2025-03-07 03:23:41.320000+00:00
11	2025-03-07 03:26:46.171000+00:00	2025-03-07 03:25:32.530000+00:00
12	2025-03-07 03:31:13.409000+00:00	2025-03-07 03:29:59.302000+00:00
13	2025-03-07 03:35:36.303000+00:00	NaT
14	2025-03-07 03:38:32.356000+00:00	2025-03-07 03:37:22.986000+00:00
15	2025-03-07 03:41:25.677000+00:00	2025-03-07 03:40:11.385000+00:00
16	2025-03-07 03:44:17.292000+00:00	2025-03-07 03:42:58.292000+00:00
17	2025-03-07 03:50:23.274000+00:00	2025-03-07 03:49:10.665000+00:00
18	2025-03-07 03:52:49.274000+00:00	2025-03-07 03:51:36.338000+00:00
19	2025-03-07 03:57:58.423000+00:00	2025-03-07 03:57:05.167000+00:00
20	2025-03-07 04:02:47.650000+00:00	2025-03-07 04:01:32.411000+00:00
21	2025-03-07 04:07:28.413000+00:00	2025-03-07 04:06:14.418000+00:00
22	2025-03-07 04:10:14.389000+00:00	2025-03-07 04:09:07.689000+00:00
23	2025-03-07 04:12:13.325000+00:00	2025-03-07 04:10:57.396000+00:00
24	2025-03-07 04:14:03.370000+00:00	2025-03-07 04:12:50.389000+00:00
25	2025-03-07 04:20:46.095000+00:00	NaT
26	2025-03-07 04:22:41.704000+00:00	2025-03-07 04:21:23.365000+00:00
27	2025-03-07 04:25:42.325000+00:00	2025-03-07 04:24:27.332000+00:00
28	2025-03-07 04:27:41.388000+00:00	2025-03-07 04:26:28.374000+00:00
29	2025-03-07 04:29:40.352000+00:00	2025-03-07 04:28:26.406000+00:00
30	NaT	2025-03-07 04:33:02.402000+00:00

Missing Image Frames:

There is a packet counter inside each image packet. From 0 to 959. Assembling one image frame requires 960 packets.
`last_packet_num`: the latest image packet counter received (not always in order!)
`missing_packet_cnt`: a bitfield keeps track of each image packet counter received. The cardinality of the bitfield provides the number of missing packets -> guaranteed to decrease monotonically.



Vertical orange lines show computed transitions between one image frame and the next. Used to compute the missing packets shown below.

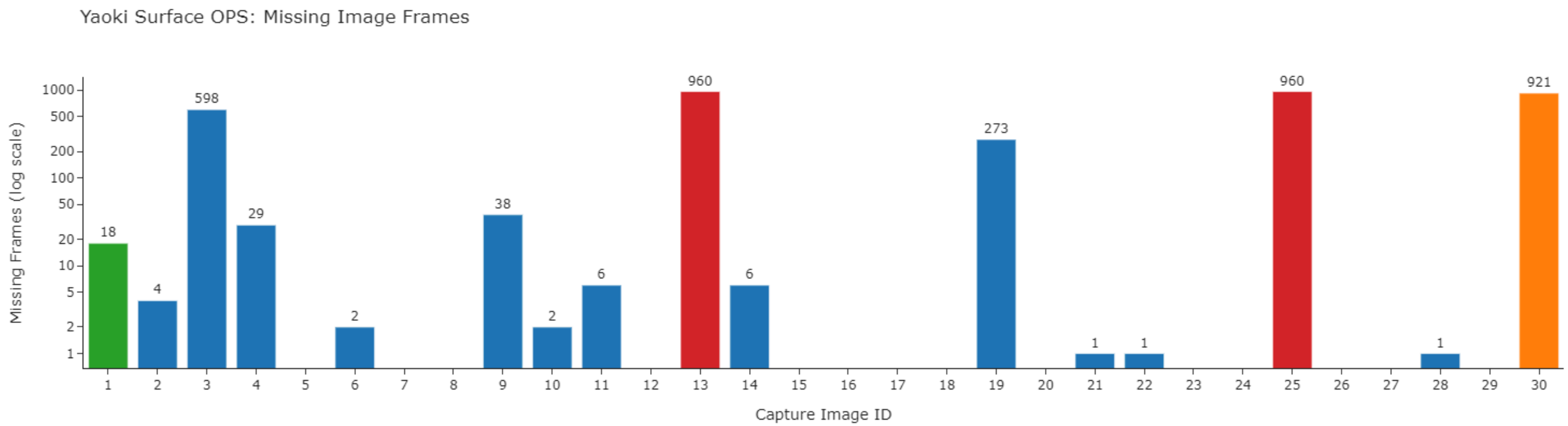
Out of the 30 images captured:

- The first image (in green) was transmitted via the store and forward channel (18 packets lost). All other images were transmitted via the Streaming channel.
- Images 13 and 25 (in red) are entirely lost (could be an issue during transmission between the rover and lander or during image capture): the only thing sure is that 0 packets were received on the ground but the image counter incremented on the Yaoki flight software.
- 12 images have 0 packet loss
- The last image (number 29, in orange) could only transmit 39 packets before the lander shut down communications (921 packets lost). This is expected and not a transmission issue.

For the images in nominal configuration (blue color) :

- The packet loss median is 0 packet loss
- The packet loss mean is 4%
- 75% of images have less than 0.4% packet loss.

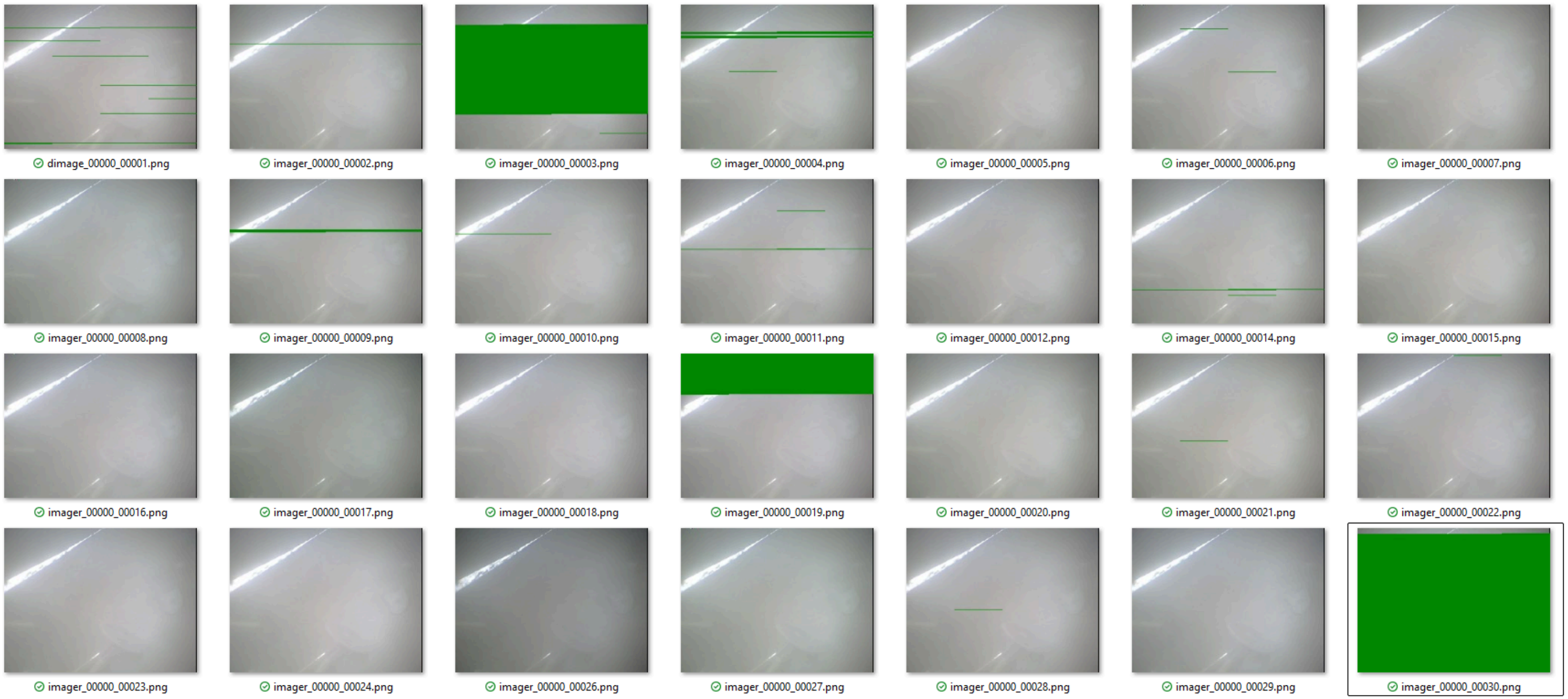
This shows a very high level of transmission reliability. The graph below shows the number of missing packets for each image: note the log scale and the many images with less than 10 packets lost (22 out of 30). Under these conditions, a simple mechanism for retransmission of missing image packets would be very beneficial.



The 28 images received on the ground are shown below. Since all the images are taken from the exact same position, it is highly likely that the rover did not deploy from the Transport Case.

In the images below, green pixels denote missing image packets.

Interestingly, the autoexposure of image 0026 is slightly different from the others.



© Dymon

Stacking all the 28 images together and taking the median value of each pixel allows for removing the effect of missing packet and reduces the overall image noise.

Top left: the sunlit edge of the dark crater is clearly visible.

Bottom center: the broken leg of the IM2 lander is visible

© Dymon

More advanced (and destructive) processing reveals additional features:

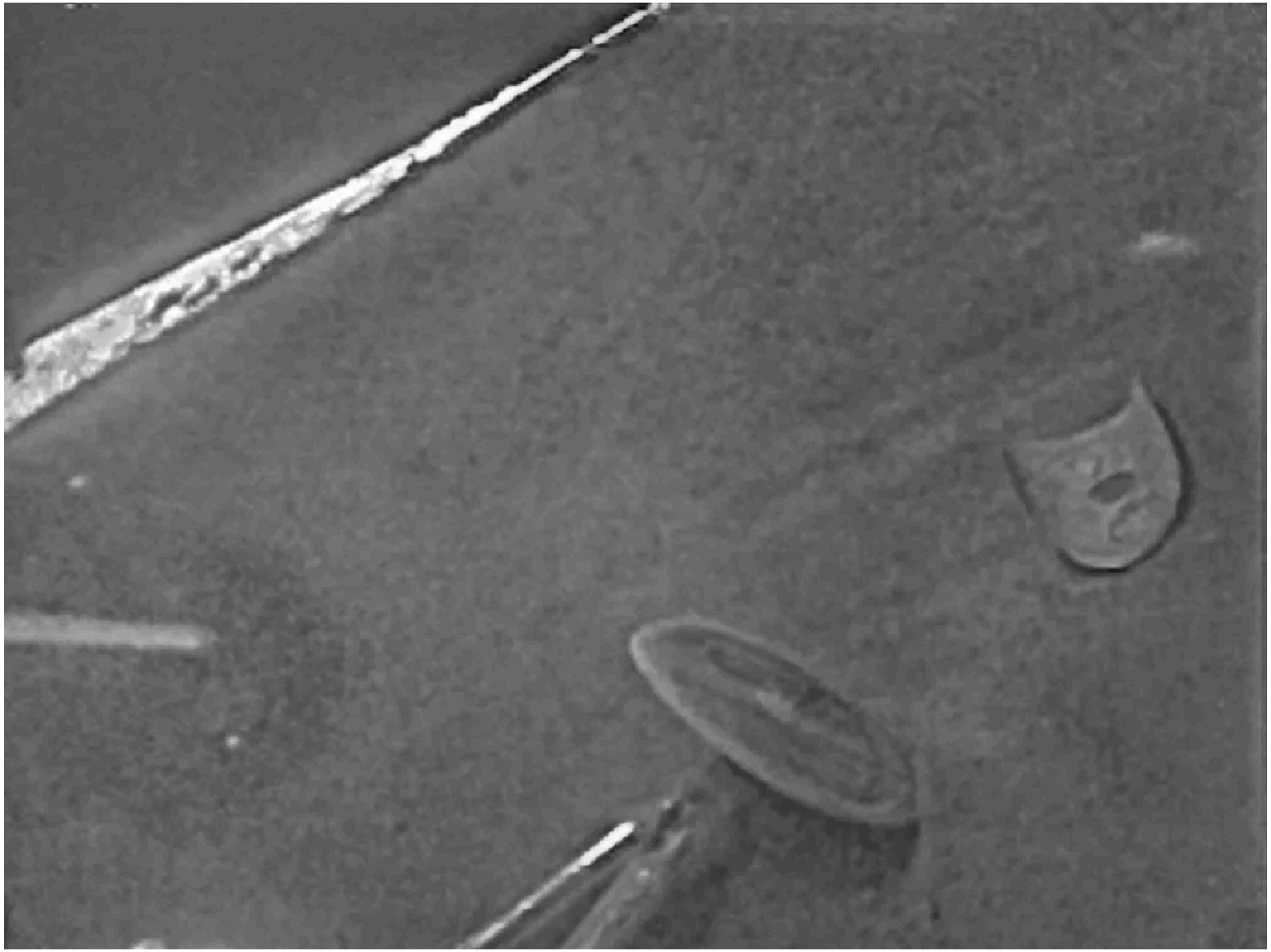
Center right: appears to be the sticker of the footpad that peeled off and curled up. The Arizona and Maryland logo is somewhat visible

Above right: appears to be a rock

Center between the main leg and the sticker: linear regolith pattern

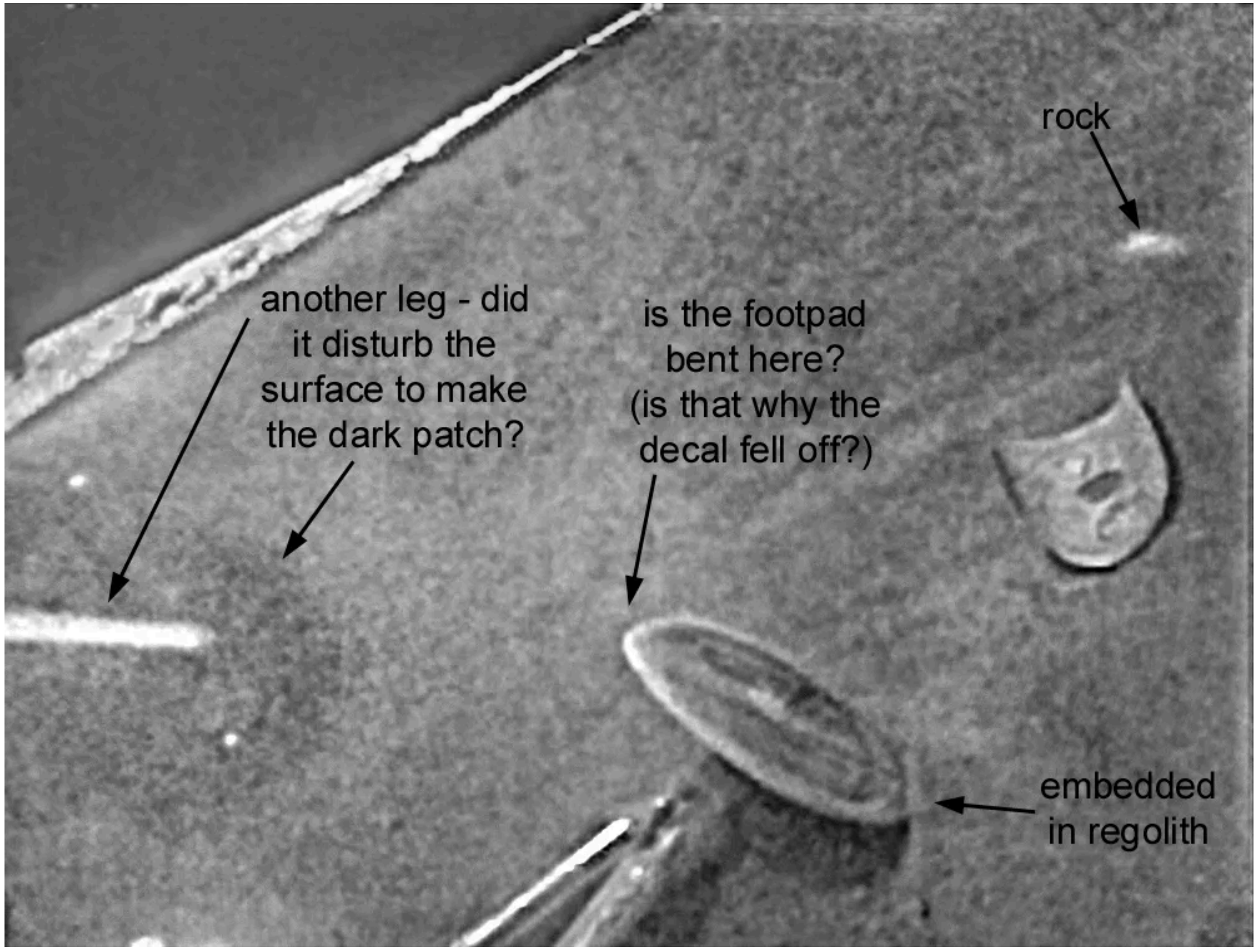
Center left: appears to be another (broken?) lander leg

© Dymon



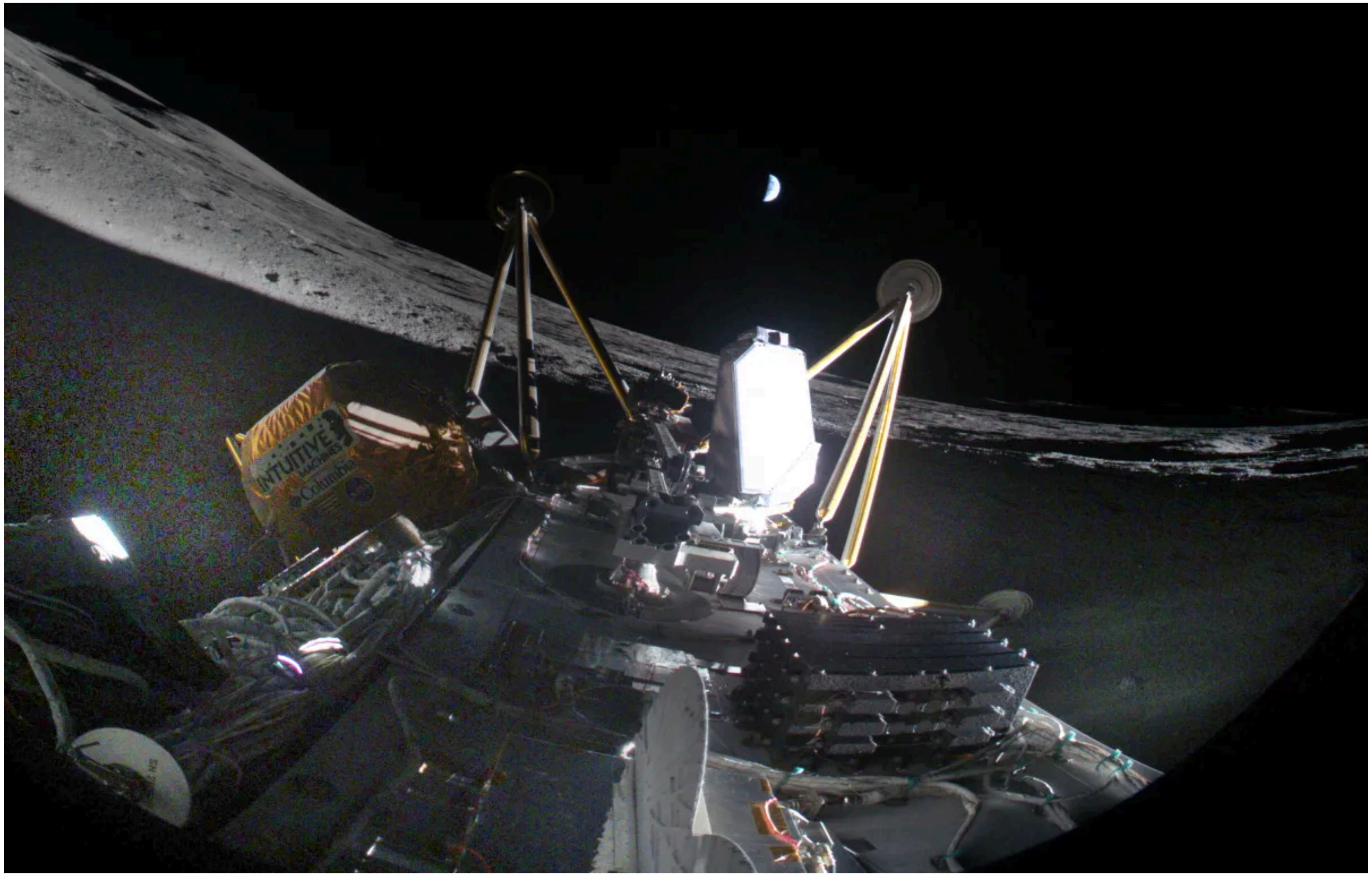
Further processing by Phil Stooke highlights additional features of the image

© Dymon / Phil Stooke

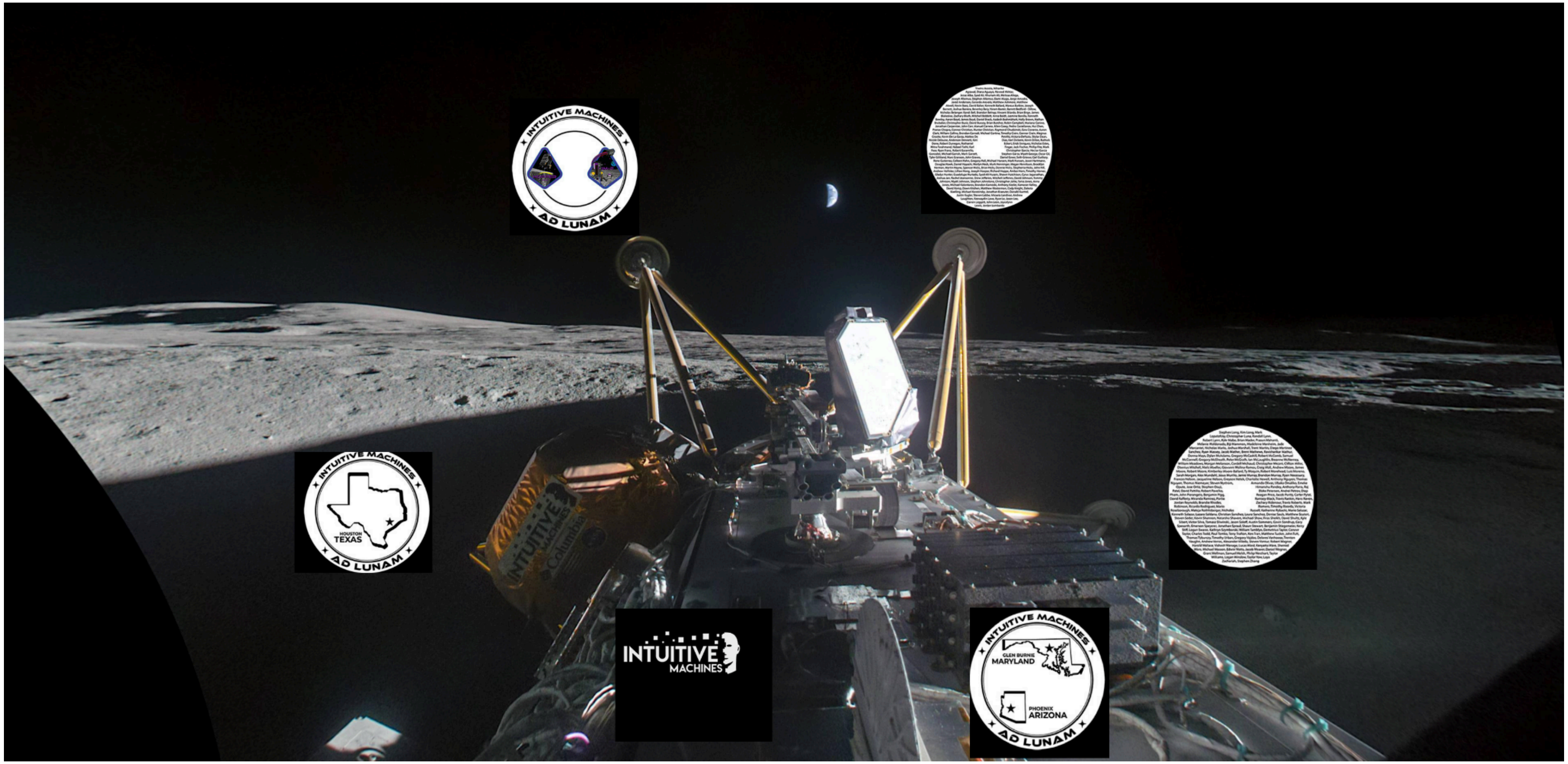


Processing by Phil Stooke of the image taken by the IM-2 lander brings out details in the shadows around the lander: note the rock on the right side of the lander (the same side that the YAOKI rover is pointing towards)

© Intuitive Machines / Phil Stooke

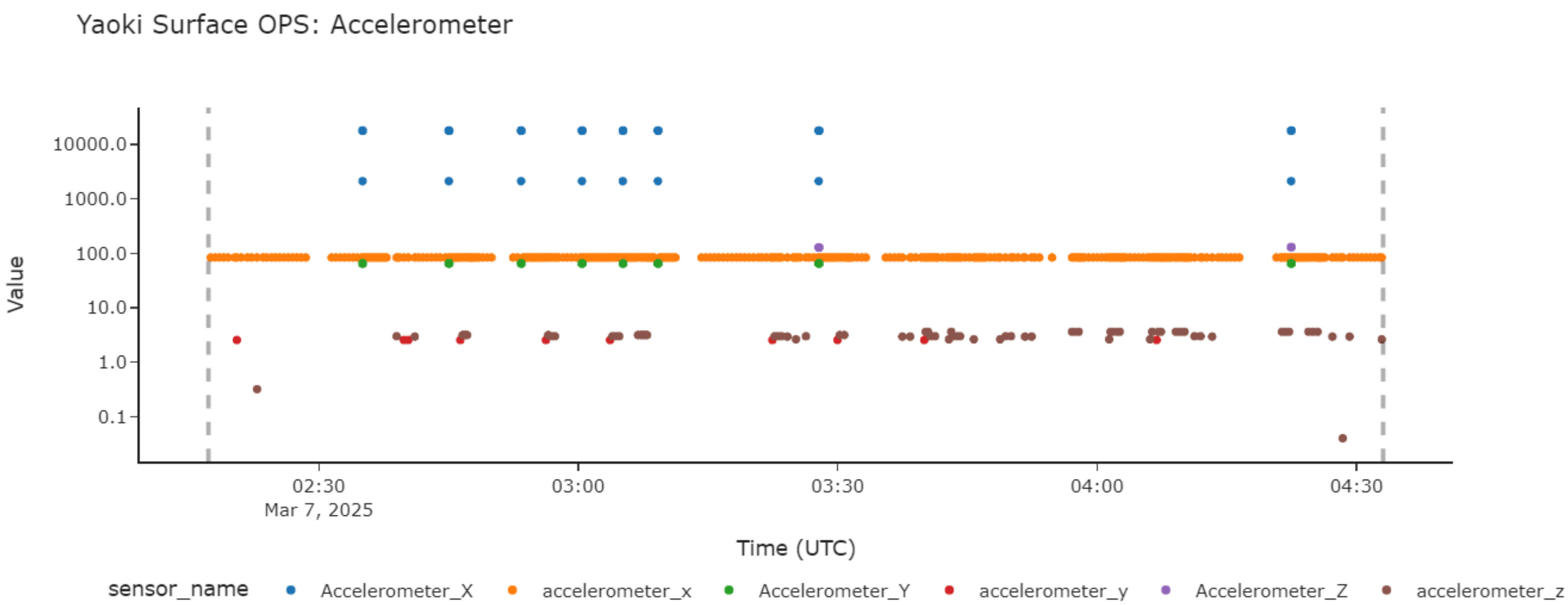


Reconstruction of the position of the footpad sticker on each of the six lander legs. The YAOKI rover is integrated between the struts of the lander leg with the Arizona and Maryland sticker.



© Intuitive Machines / Phil Stooke / Louis Burtz

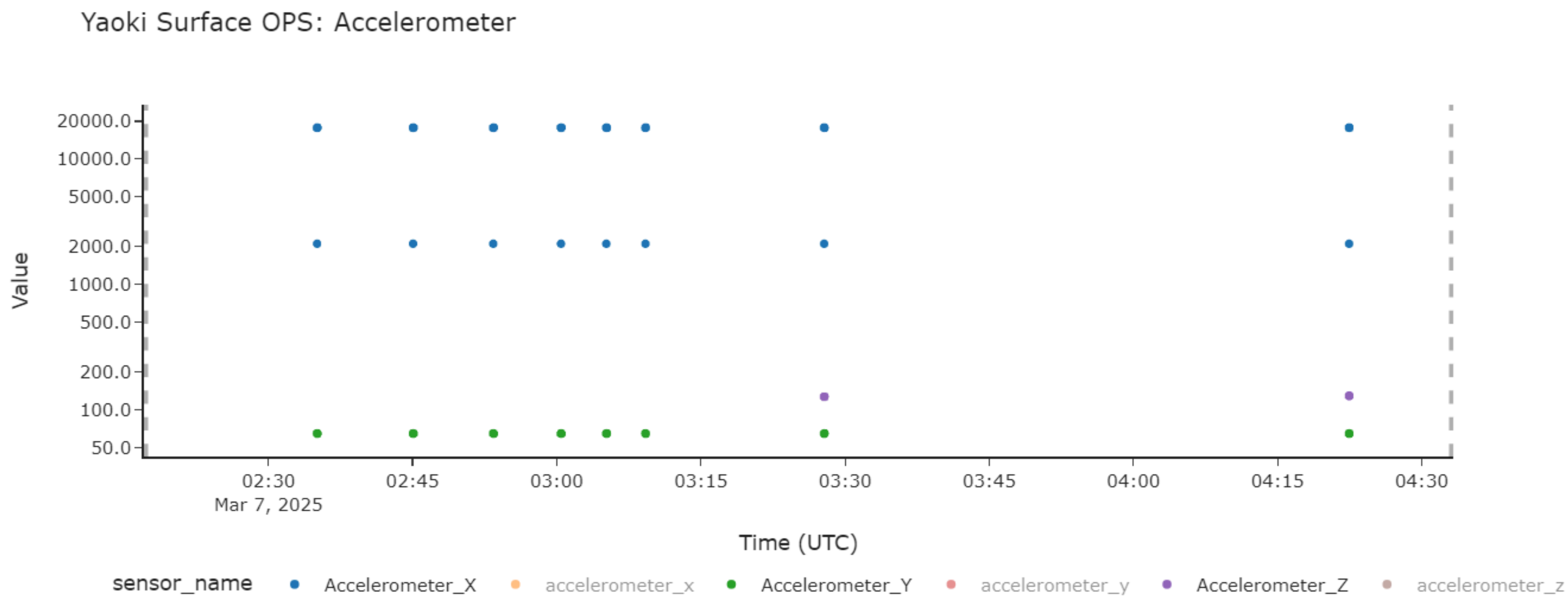
3.5 Accelerometer Data Analysis



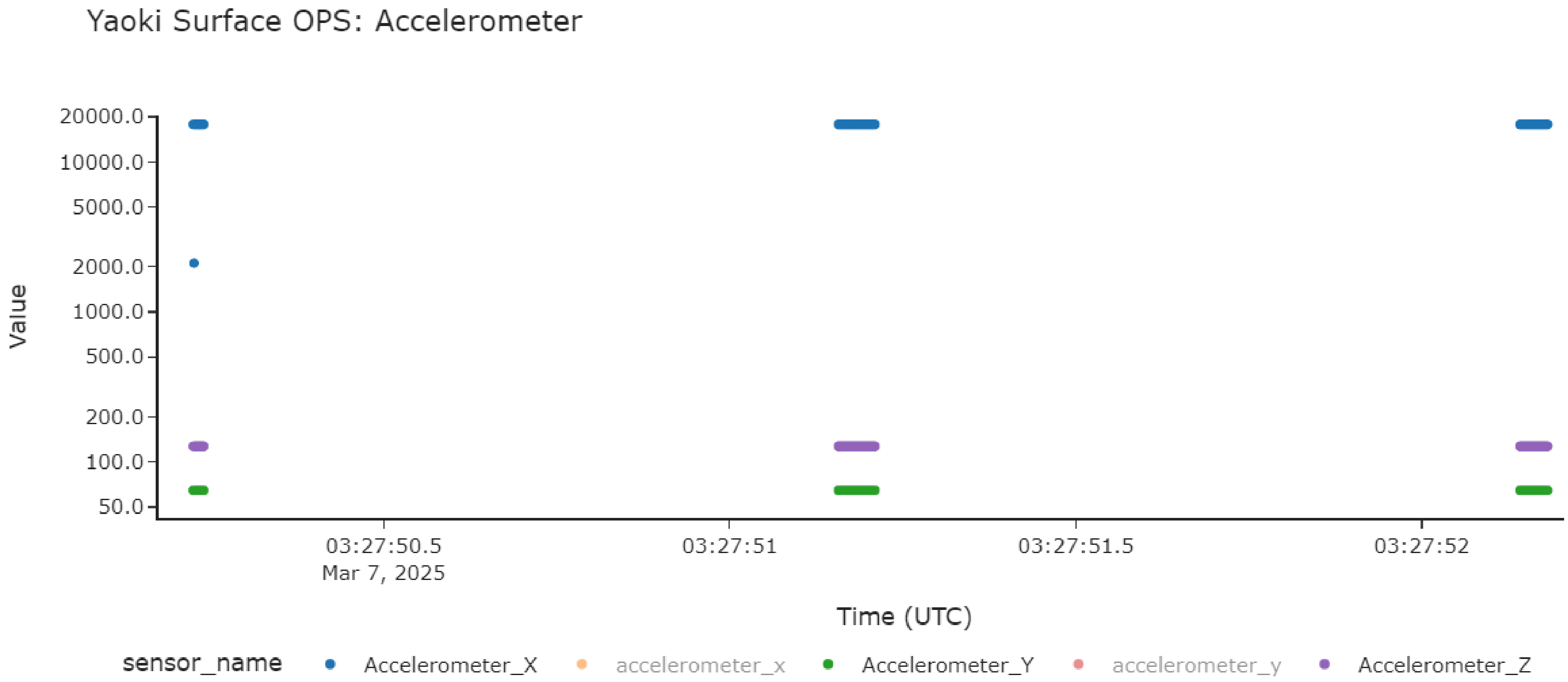
Clearly there are several issues in the YAOKI flight software for gathering and sending the accelerometer data and also most probably issues in the IMU hardware itself.

Even for a static rover perfectly still on the Moon, there should be variations due to noise. But most values are exactly the same (within the quantization resolution, which is another issue)

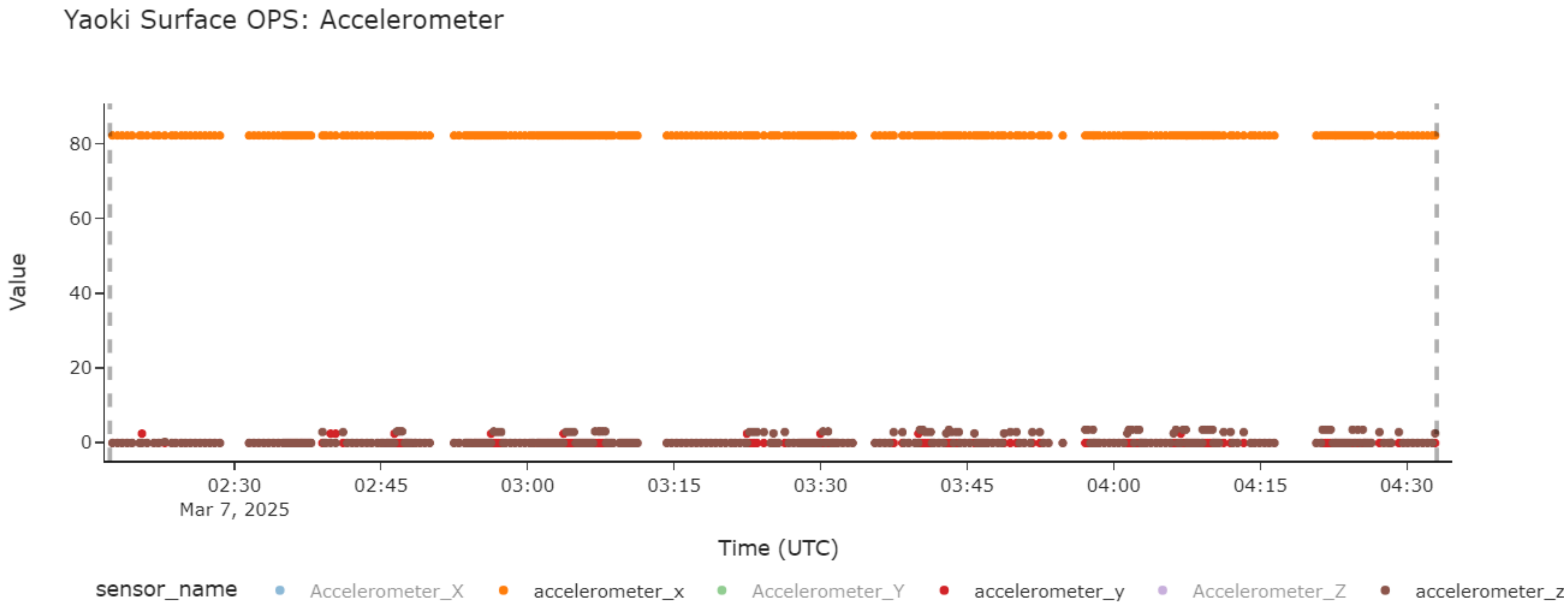
As shown in the graph below, “Accelerometer_Z” values are only received for the last two wire-cut sequences: unknown reason.



Data from the accelerometer labeled as ‘raw’ : after each ‘wire cut’ command, it is sent for 300 cycles (a few seconds)



The graph above shows one burst of accelerometer data when the ‘cut wire’ command is sent. (the three bursts are from the same ‘cut wire’ sequence, they are packetized and timestamped when received on the ground). This is another issue: the timestamps are meaningless and the order that the measurements are received is not guaranteed to be the order in which they were produced.



The graph above shows data from the accelerometer labeled as ‘scaled’: it is sent every ~30 seconds as part of regular housekeeping telemetry.
The data is also constant, without noise fluctuations and without meaningful units. No correlation was found between the accelerometer_z values (brown) and rover drive commands.

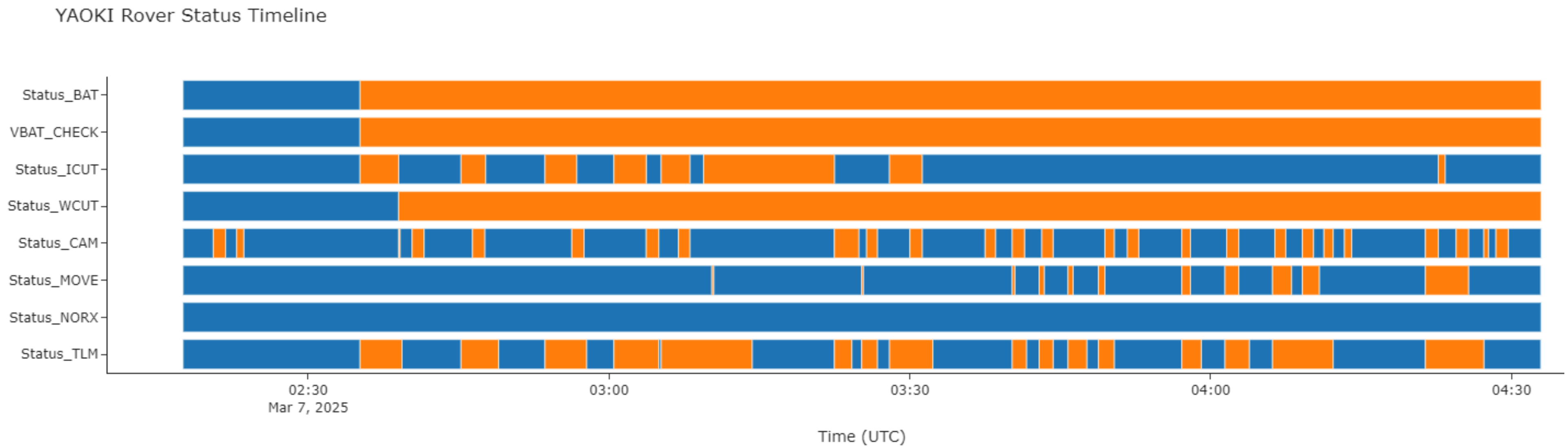
The data is also very low frequency, about one data point every 15 or 30 seconds. This is too low to understand any dynamic behavior of the rover. (Lessons learned: the accelerometer data should be logged onboard the rover at a higher frequency and then sent to the ground e.g. every 30 seconds)

See the YAOKI FSW code below for the conversion:

```
    g->x = 0xFFFF&(((uint16_t)data[1] << 8) | data[0]);

    g->y = 0xFFFF&(((uint16_t)data[3] << 8) | data[2]);

    g->z = 0xFFFF&(((uint16_t)data[5] << 8) | data[4]);

    g->fx = (int16_t)g->x * 0.0392266;

    g->fy = (int16_t)g->y * 0.0392266;

    g->fz = (int16_t)g->z * 0.0392266;
```

This type of data conversion / calibration should not be performed on the flight software. The raw data should be sent and any processing shall be implemented in the ground system (Yamcs has extensive support for processing raw values into engineering data). This avoids irrevocably altering the data onboard and enables adjusting calibration during post processing.

3.6 Status Bits Analysis



Blue is 0 (OFF or LOW) and Orange is 1 (ON or HIGH)
Each status bit is explained in the table below:

Name	**Description**
Status_BAT	Status battery on
VBAT_CHECK	YAOKI power source: 0 if on lander power / 1 if on internal battery power
Status_ICUT	The deployment wire is currently being cut
Status_WCUT	The command to cut the wire was sent AND the time allocated for the wire cut heater has elapsed
Status_CAM	The camera is actively taking the picture / sending the data via CSP
Status_MOVE	The rover should be moving (e.g. time remains since the last move command set the move timer)

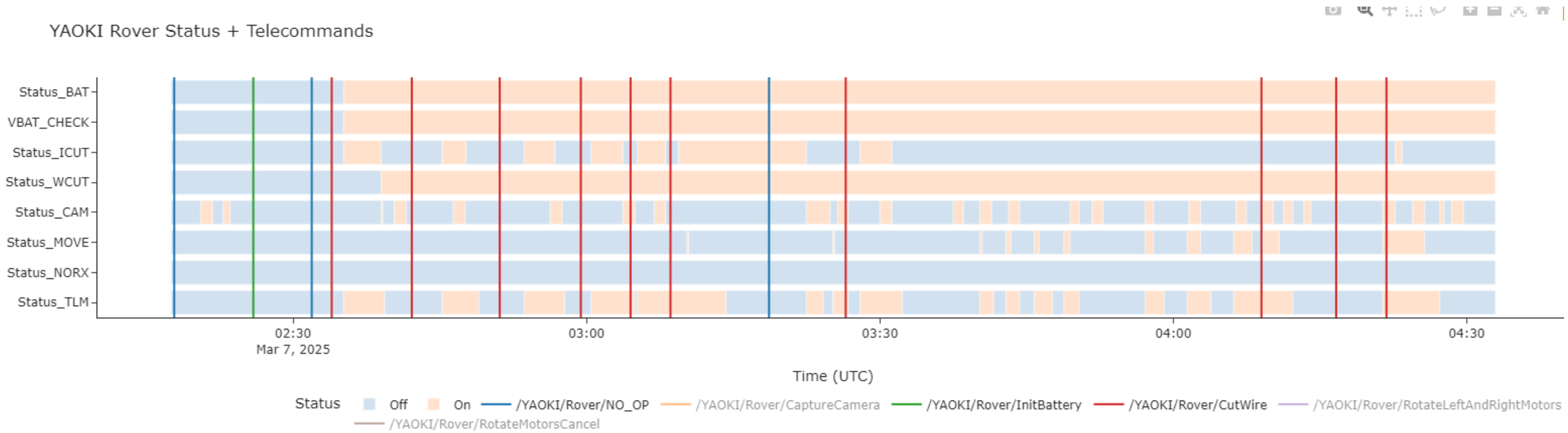
Status_NORX	No pings received for 3 minutes
Status_TLM	Telemetry transmission mode: 1 if high frequency / 0 if low frequency

Analysis:

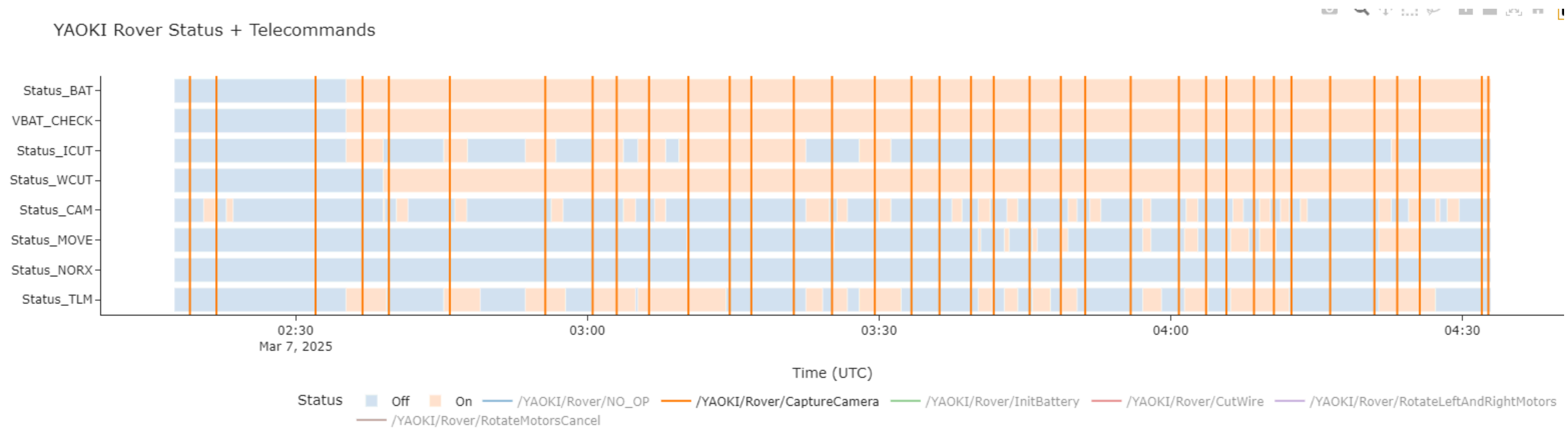
- The Satus_BAT and VBAT_CHECK items show when the rover was switched to internal battery power. They are identical as expected.
- Status_ICUT shows that we repeatedly sent commands to cut the deployment wire. The commands were sent with various time durations. Unfortunately a bug in the FSW seems that the time duration was not respected above a certain threshold of around 4 minutes. **The sum of all durations where Status_ICUT is ON is 32 minutes, 30 seconds which is too low to cut the burnwire in the cold conditions.** Furthermore, consider that in between the several WireCut commands, the burnwire is cooling down, which drastically reduces the effective heating time and drastically reduces the maximum burn wire temperature. Furthermore, consider that the lander-provided power needed to activate the deployment mechanism was turned off by the lander team early into the surface operations (due to the lander's lack of power availability).
- Status_WCUT was correctly set as soon as the first command to cut the wire was finished
- Status_CAM shows the rover in camera capture mode. Due to the low frequency of the telemetry updates, some image captures appear longer than others and some are missed. This is probably only due to the low frequency of the telemetry updates.
- For the same reason the Status_MOVE does not always reflect when move commands were sent
- The NORX status is always low: this means that there never was a period of time longer than three minutes where radio pings were not received by the rover. This is a positive result showing the reliability of the radio transmission system.
- The Status_TLM item shows when the telemetry transmission is in high mode. Which is supposed to happen when the rover is moving.

Further analysis could verify the proper function of the status bits in the dynamic environment. This should be performed on the engineering model of the next rover mission (after software updates have been made), as part of the standard test protocols before flight.

3.7 Combined Analysis of Telecommands and Status Bits

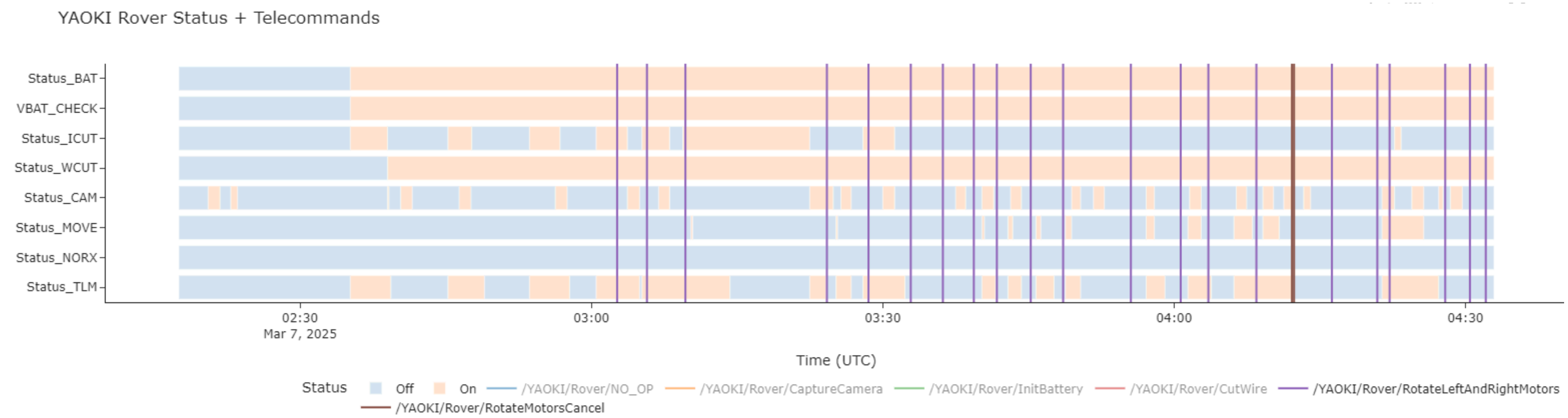


The InitBattery command did not work: the Status_BAT and VBAT_CHECK and Status_ISCUT and Status_WCUT show that the transition to internal battery was not performed at the time of the InitBattery command. That transition was only performed by the automatic function (safety measure) as part of the CutWire command. This shows the importance of the safety measure and the need for better operations procedure to verify the battery status after the InitBattery command.



The Status_CAM shows that the CaptureCamera commands were executed successfully almost all the time.

The CaptureCamera command was sent 36 times but 11 commands did not lead to the Status_CAM status change. This is compatible with the fact that only 30 images were taken by the rover (as per the camera counter): some commands did not reach the rover (this is expected due to the UHF radio link not being full duplex) and maybe some commands were not accepted by the YAOKI FSW (hypothesis)



The Status_MOVE shows that the RotateLeftAndRightMotors commands were executed successfully almost all of the time.

For short movement commands (at the beginning of the operations) it is probable that the command was received and executed but the low frequency of the status telemetry did not report the status change.

The entire overlay of the commands and the status bits is given below: