

# Modelling operation of the PEP instrument in Ganymede orbit

**Gabriella Stenberg Wieser** (1), Stas Barabash (1), Pontus C. Brandt (2), Yoshifumi Futaana (1), André Galli (3), Dennis Haggerty (2), Geraint Jones (4), Norbert Krupp (5) and Elias Roussos (5)

(1) Swedish Institute of Space Physics, Kiruna, Sweden, (2) Applied Physics Laboratory, The Johns Hopkins University, Laurel, Maryland, USA (3) Universität Bern, Physics Institute, Bern, Switzerland (4) The Centre for Planetary Sciences at UCL/Birkbeck, London, UK (5) Max Planck Institute for Solar System Research, Göttingen, Germany (gabriella@irf.se)

## Abstract

In preparation for the JUICE mission we model the operation of the Particle Environment Package (PEP) instrument. Our aims are to look at power consumption, data production rates and usage of the onboard memory. We also want to evaluate if a certain operation scheme allows us to fulfill our science goals. We start by making a simplified operation scenario for the phase of the mission when JUICE orbits Ganymede.

## 1. Introduction

The Particle Environment Package (PEP) [1] is one of the scientific instruments on the JUICE (Jupiter Icy Moons Explorer) mission to Jupiter. PEP consist of six sensors listed in Table 1. With the launch scheduled in 2022 building and testing of the sensors are ongoing with more than full speed. In parallel we model how PEP should be operated once JUICE is launched. The available resources on the spacecraft in terms of power, telemetry and on-board storage will be limited and most of the time only some of the PEP sensors will be run at the same time.

Table 1: The six sensors in the Particle Environment Package (PEP)

Sensor	Measures
JDC	Plasma ions and electrons
JEI	Plasma electrons and ions
JoEE	Energetic electrons
NIM	Thermal neutrals and ions
JNA	Low energy ENA
JENI	High energy ENA and ions

## 2. Modelling

We use our own developed Python code (SpiXel) to model different operation scenarios. An illustration of the implementation is shown in Figure 1. The input needed as well as the output returned are described below.

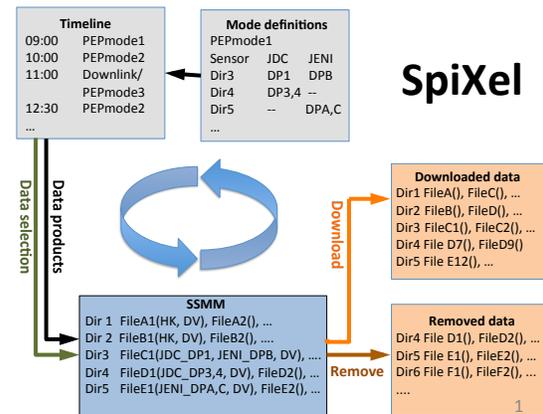


Figure 1: Illustration of the implementation of SpiXel.

### 2.1 Input

Input is provided using two Excel documents. The first contains information about the different data products each sensor can generate and the telemetry associated with each of them. The same document also describes the power usage of a sensor depending on the state of the sensor. Outgoing from the capabilities and properties of each sensor we combine them into macro modes (so-called PEP modes), which requires a certain amount of power and produces a specified set of data products from the different sensors, either for direct downlink or for temporary storage on-

board. The second input document is a simple timeline, where we specify when we want to run different PEP modes. The timeline also contains information about downlinking periods, during which the available power for science measurements is lower than normal. In a first full scale operation scenario we choose to look at the last phase in the mission, when JUICE is orbiting Ganymede. To keep it simple we will only use a few PEP modes.

## 2.2 Running

SpiXel uses SPICE kernels to get the position and the attitude of the spacecraft as well as distances to different objects in the Jupiter system. The time resolution of SpiXel can be adjusted. The basic resolution is given by the times specified in the scenario file, but to get a smooth output it is recommended to increase the resolution. For each time step SpiXel is book-keeping how much data volume is put into the different available directories, how much power is used and how much data is downlinked. The modeler can specify which data is prioritized for downlinking and for example select certain time intervals, mimicking the selective downlink process.

## 2.3 Output

As output from a run we get time series of power consumption, telemetry production, downlink rates (from each directory), content of the downlinked data (that is, which data products) and the fill level of the solid state mass memory. A simple example how the output can look like is given in Figure 2. The first panel shows how the distance to Jupiter change during this week-long sequence. In the second panel one sees which of the sensors and DPUs (Digital Processing Units) are on at a certain time. The grey vertical stripes indicate downlinking periods. In this example all PEP sensors are off during downlink. The third panel shows the data production into different directories and the bottom panel shows how much data is downlinked, both instantaneously and on average.

## 3. Summary

We model PEP operations when JUICE is in orbit around Ganymede. We use only a few operation modes to evaluate the necessary use of power and telemetry. We also look at the usage of the selective downlink option and compare the scenario and the acquired data with our science objectives. This

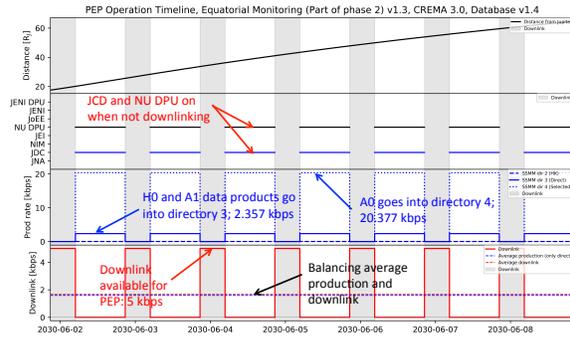


Figure 2: Example of output time series from SpiXel. Note: This example is not from the Ganymede phase.

modelling approach will eventually help us design and optimize the PEP macro-modes to be used during the mission.

## References

- [1] Barabash, S., P. Wurz, P. Brandt, M. Wieser, M. Holmström, Y. Futaana, G. Stenborg, H. Nilsson, A. Eriksson, M. Tulej, et al.: Particle Environment Package (PEP). In European Planetary Science Congress 2013, held 8-13 September in London, UK.