DCAS

Summer Internship 2022

Data Exchange Protocol of CREME CubeSat

Summer Internship Report

Author:

Daniel OPOKA

Supervisors:

Jonathan DETCHART, Thibault GATEAU, Jérôme PUECH

26/08/2022

Table of Contents

List of Figures

List of Tables

1 Introduction

Consistent information exchange between the ground segment and a Satellite is a requirement most space missions must fulfill. This can be ensured by having potent antennas, attitude and orientation control system to ensure optimal pointing, and many more subsystems. In the advent of New Space, satellites are now being exchanged with constellations of smaller satellites due to the reduced price in launcher costs. This trend has thus allowed for many educational institutions and companies to develop smaller satellites and be part of an economy that was otherwise dominated by multi-million dollar companies [1]. This report will focus on the development of a consistent and optimal data exchange protocol for the CREME mission.

1.1 CREME CubeSat

The CREME (CubeSat for Radiation Environment Monitoring Experiment) is a 3 Unit CUbeSat nano satellite. The main mission of this CubeSat is the study of the South Atlantic Anomaly (SAA), a region of the Earth's Magnetic Field with high levels of disturbances which may cause damage to satellites in Low Earth Orbits [2]. The following image illustrate the overall organisation for the development of the satellite as well as all the organisations involved:

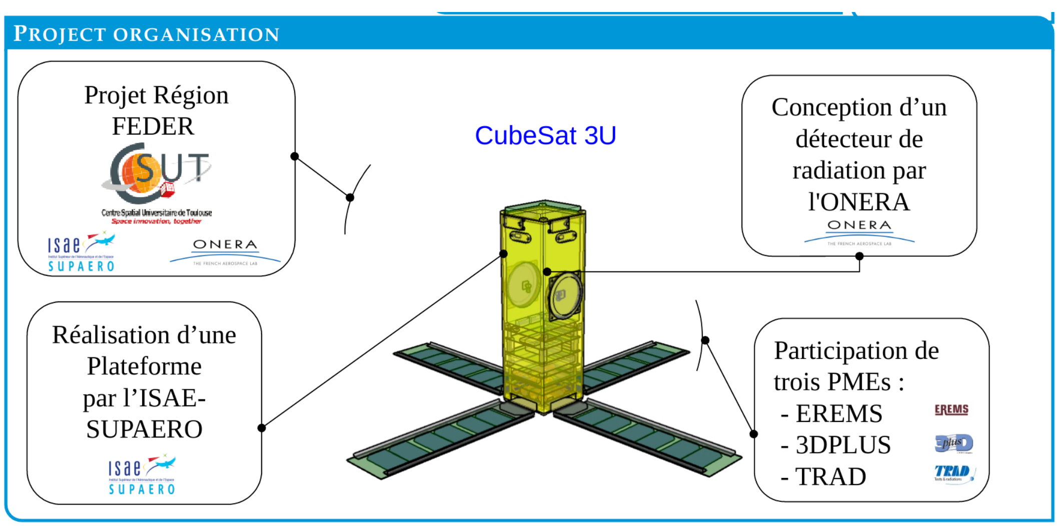


Figure 1: CREME project organisation

For communications the satellite is fitted with 2 S-Band TT&C Antennas, from the company ANYWAVES, with the following performance and characteristics:

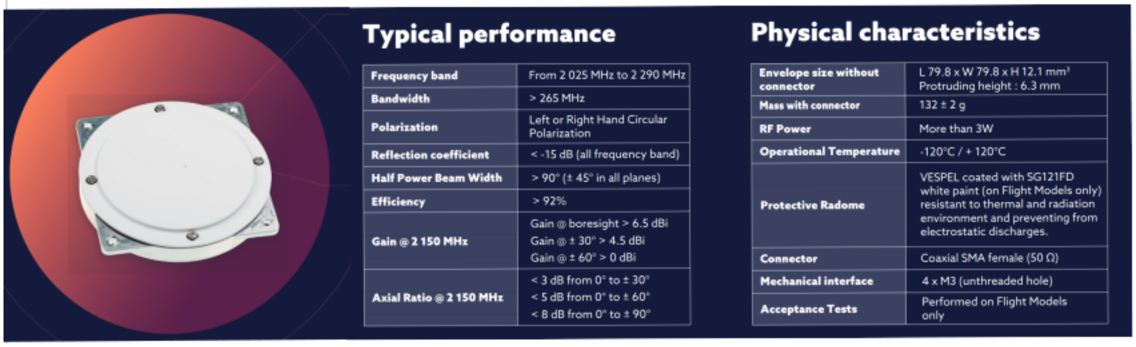


Figure 2: S-Band TT&C Antenna performances and characteristics

The satellite's main structure is organized in such a way, that both antenna point in opposite faces of the satellite. This configuration ensures that even if the nano satellite were to tumble, there could be a consistent communication link between the ground segment. The following figure shows the organisation of how the satellite components are placed:

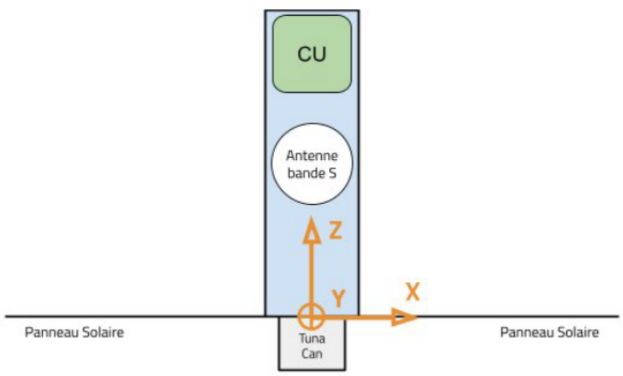


Figure 3: Setup of CREME main structure

1.2 Problem Statement

To ensure optimal measurement from the SAA by the payload sensors, the satellite must be spinning about it's z-axis at around a speed of $2\,\frac{\circ}{s}$. The satellite has magnetorquers which provide a slight attitude control from this tumble. For the majority of the orbit trajectory, the satellite will be in a "spinning sun pointing mode" where the satellite will always points towards the sun however it's z-axis is free to rotate. There are other configurations of trajectory which will be omitted for this report, as it is of interest to study the most common scenario where information can be sent to the ground segment. When the satellite is in range with the ground segment there may be moments when the satellite will lose connection. This occurs as the satellite's tumbling will cause the antennas to be pointing either towards or away to the ground segment. This is illustrated in the following figure:

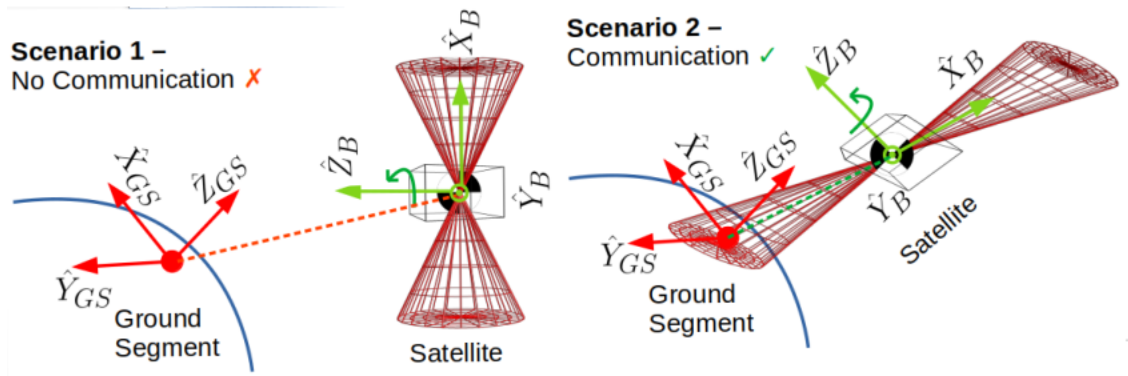


Figure 4: Satellite communication scenarios

The inconsistent connection will cause intermittency in the communication channel, messages that are sent from the satellite may sent either towards the Earth or to Outer Space. This loss of information can be modeled in many ways such as a Binary Erasure Channel (BEC), a communication channel where there is a probability that the sent 0 or 1 will be "erased". To counteract this, an Erasure Correcting code can be implemented. where redundant information will ensure that even if some information is lost the original message can still be decoded.

As such, the main objective is to develop an optimal data exchange protocol where the least amount of information is lost/erased. To accomplish this, a simulation of the spin on the satellite/tumbling of the satellite was created using a python script; where a mathematical model of the satellite's and the antenna's position will determine if the ground segment has an appropriate received power from the satellite to send information. This simulation will define the parameters for the Erasure Correcting code needed for this scenario. Additionally, the following table will illustrate important characteristics from previous studies:

Characteristic	**Quantity**
Daily bits (b_{daily})	67.6 Mb
Transfer Frame size (TF_{size})	1115 B
Space Packet size (SP_{size})	64 B
Downlink bit rate (br)	85 kbps
Connection Time Range per Day (t_{con})	[16 min, 25 min]
Residual spin (ω_z)	$[2\,\frac{\circ}{s}, 15\,\frac{\circ}{s}]$

Table 1: Characteristics of communication intermittence

This data is from the following repository of the GitLab of the CREME mission: *creme-common/1_management/7_presentation/1_PDR/CREME-PDR-0-006-ISAE-ED0.pdf* in pages 21, 24, 75 and 78.

2 State of the Art

There are many ways of solving the stated problem, many research papers have been done on the communication of satellites in LEO and deep space. CubeSats have many limiting factors, their communication may suffer intermittence due to the reasons stated in Section 1.2. As such DTN (Delay Tolerant Networks) are a necessity to ensure reliable communication. Signals that fade even for just a few milliseconds may yield to the loss of information. As such, these networks may permit the capabilities for not just ensuring that messages are received but also a probability that they will be lost. To alleviate this issue, constellations of many small satellites can ensure low intermittence [3], [4]; this however, is not the configuration of the CREME mission, as it only involves 1 satellite.

Coding rates, the ratio between uncoded and uncoded plus coded bits, help quantify how much redundant data there. Forward Error Correcting codes and Erasure Correcting codes both use this concept for their algorithms. Coding rates for CubeSats, given their limited link budget and visibility window, are around R = $\frac{7}{8}$ [5]. These coding rates thus present a theoretical upper limit of how much redundant data the proposed algorithm will have as well as help in verifying the results of the simulation used.

The following document from the CCSDS, the Consultative Committee for Space Data Systems, is the cornerstone of this research:

ERASURE CORRECTING CODES FOR USE IN NEAR-EARTH AND DEEP-SPACE COMMUNICATIONS [6]

The following figure illustrates the coding scheme of how to implement the selected algorithm:

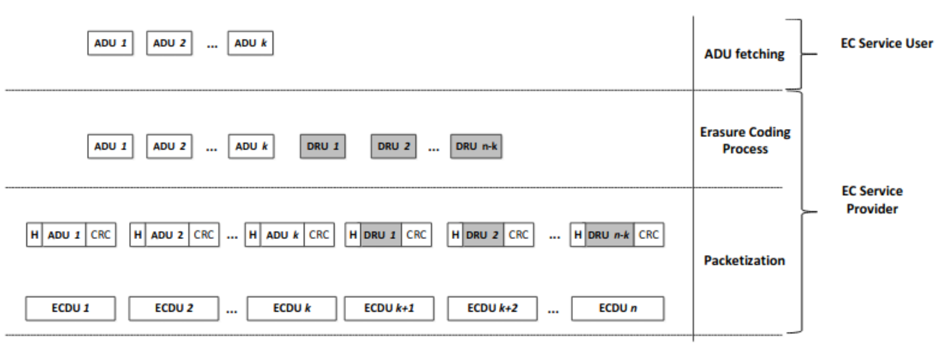


Figure 5: Erasure Coding process description

Where an ADU stands for Application Data Unit, a DRU stands for Data Redundant Unit and an ECDU Erasure Correcting Data Unit. Essentially, the data provided by the satellite (ADU) will have redundant information (DRU) such that even with the loss of information the original message can be decoded from the received data (ECDU).

The document explains the standardized approach used by satellites to handle erasure of information during communications. In it, the proposed algorithms are F-IRA codes (Flexible - Iregular Repeat-Accumulate), a type of LDPC codes (Low Density Parity Check). These codes allows for vast amounts of information to be reliable sent even with high probabilities of erasure.

Many models exists which allow to more appropriately simulate the intermittence of communication, this research proposes 2, the BEC (Binary Erasure Channel) and the 2-State Finite Markov Chain. The BEC is a simple probability of a message being erased, while the Markov Chain represents a more realistic interpretation with further complex probabilities. This chain can be used to accurately simulate any medium for which information may be lost, ranging from packet losses in the internet to even the intermittent communication of a CubeSat [7]. Only the BEC was able to be implemented in this investigation due to time constraints, however as a future development it would be worthwhile to create the 2nd proposed model for higher accuracy in the simulations to what the real network may be.

Satellite's use a standardized method of communication made by the CCSDS, where they use space packet and transfer frames. Space packets are packages of information which are sent towards the Earth; these packages are then organized inside what is called a transfer frame such as to send multiple of these at one time. The following figure illustrates the structure of these:

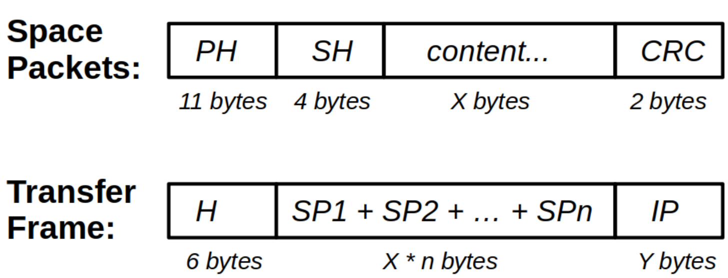


Figure 6: Data structure of space packets and transfer frames

Space packets are made up of a primary header (PH), a secondary header (SH), its content, and lastly a cyclic redundancy check (CRC). The primary header helps distinguish what kind of information it has and the secondary header allow for determining the Unsegmented Time Code (CUC). The content can range from being housekeeping data to measurements done by the scientific equipment onboard. Lastly the cyclic redundancy check are added at the end to ensure the proper management of the space packet. These do not have a fixed size.

Transfers frames are made up of a header (H), the space packets which are going to be send and an idle packet (IP). Idle packets are miscellaneous data that is added at the end to ensure that the transfer frame will always have the same size.

3 Simulation of Tumbling

To accurately simulate the intermittence of the communication channel between the satellite and the ground segment, two simulations were made. One involves the kinematics and orbital mechanics involved in the trajectory and attitude of the satellite, and the other with creating a probabilistic interpretation of the results of the other simulation as a BEC. Both of these will help in illustrating the capabilities of the communication channel in terms of how much data can actually be sent and how much of it is erased. The results will thus help in parameterizing the Erasure Correcting codes.

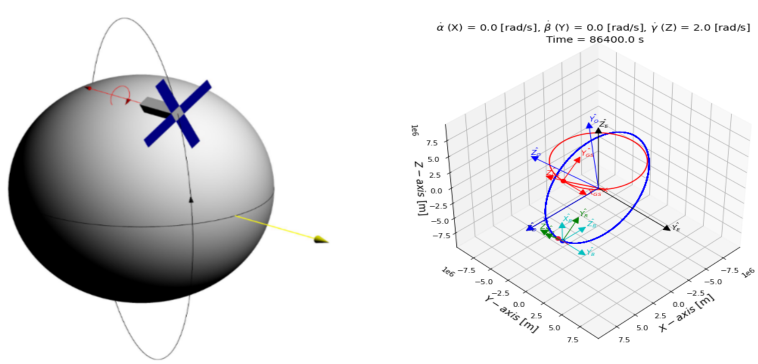
3.1 Orbital Mechanics

For this simulation, an Earth Centered Inertial (ECI) frame is used, where the x-axis points towards the spring equinox of 2022 and the z-axis points towards the north pole. This fixed reference frame will act as the base reference frame of the entire coordinate system. From this, an orbital plane will be created where the satellite will revolve as a function of time about the elliptic/circular orbit. The satellite's position in this orbital plane will be that of its Center of Gravity (CoG). Independently of the orbital plane, the satellite may tumble as per defined in the simulation parameters about its x, y or z axes. To simulate the CREME mission, an orbital period of 96.69 *minutes* was chosen and the following are the orbital parameters used to create the orbital plane:

Semi-major axis (a)	**Eccentricity (e)**	**Inclination (i)**	**RAAN**	**Argument of Perigee**	**Mean Anomaly**
6978.14 km	0	97.79°	TBD	0°	0°

Table 2: Keplerian orbital parameters for simulation

To simplify the simulation, only the "spinning sun pointing" mode was considered for the trajectory. This is illustrated in the following figure by an illustration from previous CREME investigations and the results of the simulations plotted in 3 dimensions:



(a) Spinning sun pointing mode illustration

(b) Simulated orbit

Figure 7: Illustration of orbit

To determine channel intermittence, an axis, placed close to the CoG of the satellite, will act as the antenna Field of View (FoV). If the ground segment, on Earth, is within the range of the antenna, there will be higher chances of communication. This FoV represents the ideal line for which if the receiver antenna is placed along it, the maximum amount of information can be sent. This axis will rotate given the rotation parameters of axes and this spin will cause intermittence, as even though the satellite is in range of the ground segment, it may not be pointing correctly towards it.

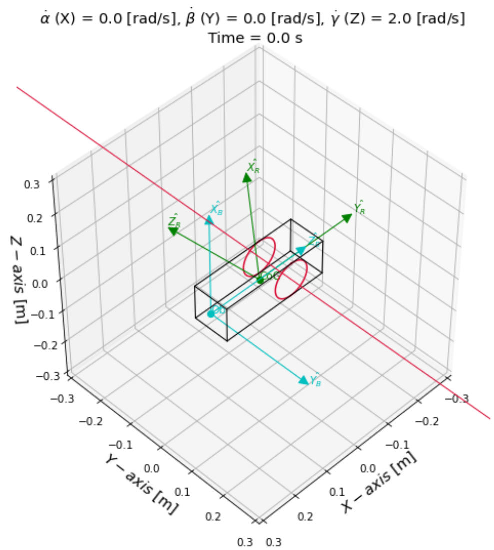


Figure 8: CREME project organisation

In the previous figure, on the left is an illustration of the ”spinning sun pointing” mode. On the right is, the satellite's FoV axis, Center of Gravity reference frame (hereafter denoted as the Orbit reference frame R) and the satellite reference frame S.

This FoV axis acts as a parameter to the antenna model, which uses the following link budget equation which is in decibel units:

$$\left(\frac{E_b}{N_0}\right)_{real} = P_{TX} - L_{TX} + G_{TX} - L_{FS} - L_{GP} + \left(\frac{G}{T}\right)_{GS} + k + BR + M_S \tag{1}$$

Where $\left(\frac{E_b}{N}\right)_{real}$ is the real normalized signal-to-noise ratio, P_{TX} is the transmitter output power, G_{TX} the transmitter antenna gain, L_{FS} is the free space loss, L_{GP} is the ground pointing losses, $(\frac{G}{T})_{GS}$ is the ground segment's antenna gain-to-noise temperature, k is the Stefan-Boltzmann constant in decibels, BR is the bit rate and M_S the system margin. All of the previously described variables are constant except for L_{FS} and L_{GP} which vary depending on the location of the satellite with respect to the ground segment. Thus the equation for the free space loss L_{FS} is:

$$L_{FS} = 10 * log_{10}\left(\left(\frac{4*\pi*d*f}{c}\right)^2\right) \tag{2}$$

Where d is the distance of between ground segment and satellite antenna (hereafter denoted as DGS), f is the antenna frequency, and c is the speed of light.

As there is experimental data for the 2 S-band TT&C antennas, the mean pointing losses from the antenna performance experiments will be taken into account. Therefore, the ground pointing losses L_{GP} will be a function of the angle between the satellite and the ground segment α_A. This helps in making a more realistic approach to the ground pointing losses, as it makes sense that if the antenna is pointing away from the ground segment there will be less transmitting power and thus more pointing losses and vice-versa.

The following figure illustrates the geometrical relation between the satellite's trajectory and the proposed antenna model:

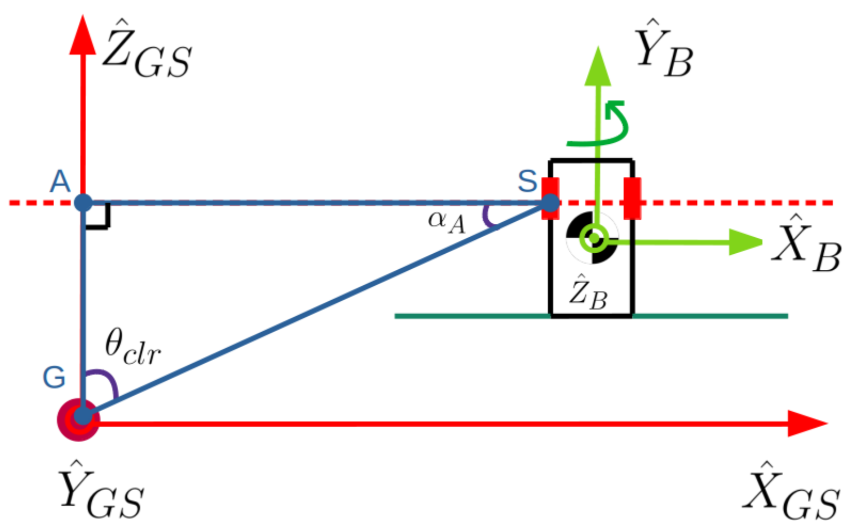


Figure 9: Geometrical relation between ground segment and satellite

With this geometrical relation, two critical parameters can be calulcated, those being the distance between the ground segment and the satellite's antenna dish (DGS) and the angle between the antenna of the satellite and the ground segment (α_A). Thus, from the previous figure, this can be calculated by the following equations:

$$DGS = \vec{GS} \tag{3}$$

$$\alpha_A = tan^{-1}(\frac{\vec{GA}}{\vec{AS}}) \tag{4}$$

Where the antenna parameters L_{FS} and L_{GP} are functions of the distance DGS and angle α_A

respectively.

All of these parameters ultimately lead to the following equations:

$$P_{RX} = \left(\frac{E_b}{N_0}\right)_{real} - \left(\frac{E_b}{N_0}\right)_{req} - M_{dwl}, \tag{5}$$

Where $\left(\frac{E_b}{N_0}\right)_{req}$ is the required normalized signal-to-noise ratio and M_{dwl} is the downlink system margin; previous studies have been done on the antenna's performance which leads the following values of the previous variables:

$$\left(\frac{E_b}{N_0}\right)_{req} = 2.5\, dB, M_{dwl} = -19.09dB.$$

The end result of this simulation is the ground segment's received power from the antenna of the satellite (P_{RX}). This received power will thus allows us to determine when the satellite is not only in visible range of the ground segment, but also when it is in contact with it. The satellite's antenna can only be in contact with the ground segment once it's emitted signal power is larger than the interference between them, in this represented by the antenna's characteristics and performance. Lastly, the following 3 requirements define the "communication clearance" for the simulations:

1. The satellite must be above the ground segment: $DGS \cdot \hat{z}_{GS} > 0$
2. The satellite must be in view of the ground segment: $\theta_{clr} > \theta_{vis}$ where $\theta_{vis} = 10°$
3. The received power must be greater than 0: $P_{RX} > 0$

All of the mathematics is explained rigorously in Annex I.I.

3.2 Binary Erasure Channels

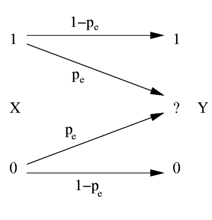


Figure 10: Binary Erasure Channel schematic

To simplify the obtained results, a BEC is used. It is a channel where information, consisting of bits, has a probability of not being received at all. This is illustrated in the figure on the left. This can be represented in a real communication channel as fading, where if the received power is not sufficient for the receiver antenna to collect the data, this information will be lost. This fading defines the erasure probability for the BEC. According to the CCSDS [6], fading can occur in the LEO with a time frame between 2 and 6 ms as shown in the following figure:

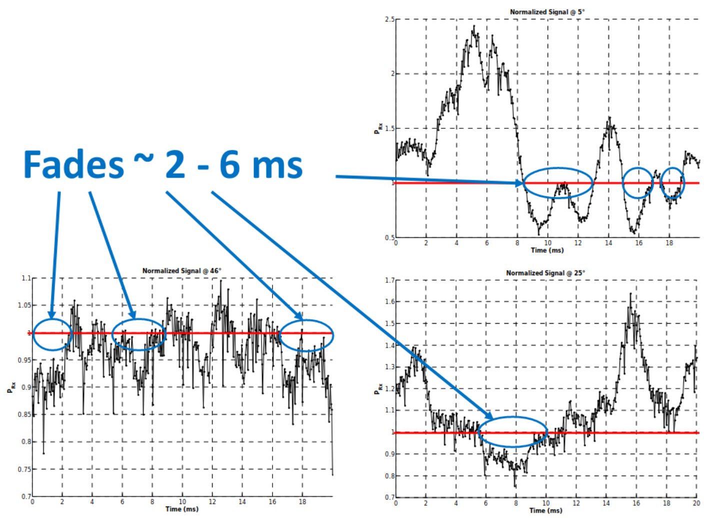


Figure 11: Typical pattern fading in optical LEO-downlinks

Thus, it is of great interest to quantify what the fading time of the described scenario for tumbling. Depending on this fading time, bits of data, space packets or even transfer frames will be lost.

To quantify how much data must be sent, it will be parameterized as a time variable. This variable will determine how much time it takes to completely send all the daily bits given the downlink bit rate, which will hereafter be denoted as the download time expressed in the following equation:

$$t_{dwl} = \frac{b_{daily}}{br} = 795.037\, s = 13.251\, min \tag{6}$$

As the amount of data fixed but the Transfer Frame may vary, an equation which takes into account of this size must be implemented. The erasure probability is a function of the coding rate, it helps to determine how much information can be lost such as to still retain the useful data. Thus, first the coding rate must be calculated such as to calculate the erasure probability. The estimated range of coding rates R_{est} and erasure probability ϵ_{est} can be calculated with the information in sub-section 1.2 and the following equation:

$$R_{est} = \frac{t_{dwl}}{t_{con}} \qquad \epsilon_{est} = 1 - R_{est} \tag{7}$$

These equations result in the following values:

$$R_{est} \in [0.530, 0.828] \qquad \epsilon_{est} \in [0.172, 0.470]$$

The values of R_{est} and ϵ_{est} will help in determining the adequate parameters for the Erasure Correcting codes. Thus, the final step to link this probabilistic approach and the physical results from the Orbital Mechanics simulation will be to reinterpret its output; more specifically the ground segment's received power from satellite antenna, P_{RX}, must be quantified in terms of whether information was received, lost of pending which will determine the signal intermittence. To do so, a "lag time" t_{lag} has been created; where the satellite will send information once it has detected the ground segment and continue to do so until there is no more connection. However, even if the connection is lost, information will still be sent until the satellite's "lag time" has been exceeded without having connected again to the ground segment; this will represent the loss of information. A flowchart of this logic is shown in Annex I.II, the Algorithm for Intermittency.

The end result of this "lag time", is the antenna intermittency A_{int}, which is discretized in to the time step t_{stp} and has 3 possible states; these states are -1, 0 and 1 which represent when information is sent but lost, no information is being sent, information is sent and received respectively. The antenna intermittency A_{int} along with a defined lag time t_{lag} will validate the intermittency model when the calculated error probability ϵ is within the range of possible answers, those being the estimated erasure probability ϵ_{est}.

The simulated outputs of the BEC, the Space Packets received where a certain percentage were lost according to the Erasure Probability ϵ, will be the input to the Erasure Correcting code.

3.3 Erasure Correcting Codes

As information can be erased in certain communication channels, it is necessary for information to be reliably sent despite this setback. When useful data is sent through this "erasing" channel, erasure correcting codes will add redundant data into what is sent; thus, even when information is lost, if a minimum amount of it is received then the useful data can be decoded [8]. In the scenario of this investigation, the useful data are the space packets within the transfer frames that the satellite sends to the ground segment. This concept is illustrated in the following figure:

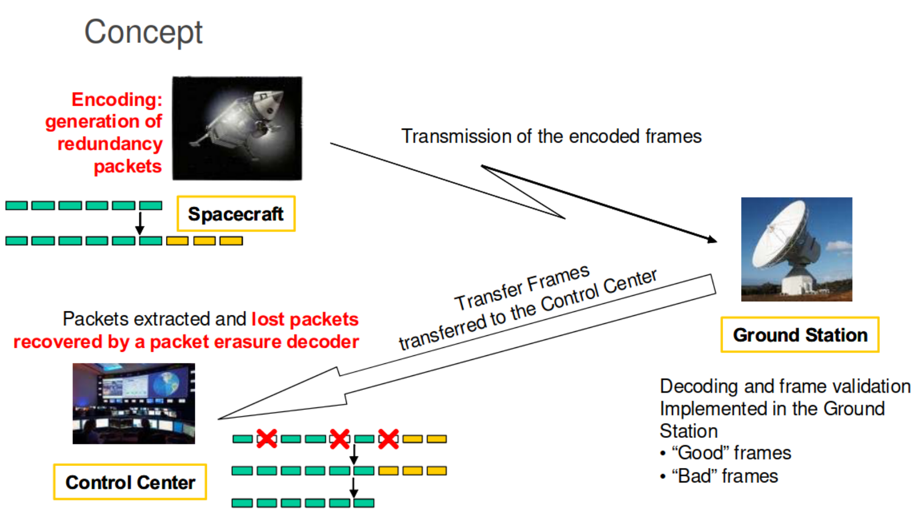


Figure 12: Illustration of information erasure (property of DLR)

There are many proposed algorithms that allow for the implementation of this concept, some of which are zfec, Low-Density Parity-Check codes (LDPC), Turbo Codes, Raptors and potentially even Sliding Windows. All of this algorithms work best with as much data as possible, but of course there must be a compromise between ideal performance and the data budget of the system. This compromise will thus reduce the performance of the codes. Due to time constraints only the zfec algorithm has been able to be implemented. however the other proposed ones may prove to be better solution.

An important concept for this codes, is the coding rate. The ratio between the useful data and the total data which includes the redundant and useful data. This is represented in the following equation under the notation of R:

$$R = \frac{usefuldata}{usefuldata + redundantdata} = \frac{k}{n} = \frac{k}{k+r} \tag{8}$$

Where k is the useful data, n the total data which is equal to the sum of k and r the redundant data. The lower the coding rate the higher the amount of redundant data, therefore in the most ideal scenario we would have as much redundant data as possible to avoid ever losing information. According to the CCSDS, the following figure shows the performances that may be achieved by using their proposed F-IRA codes:

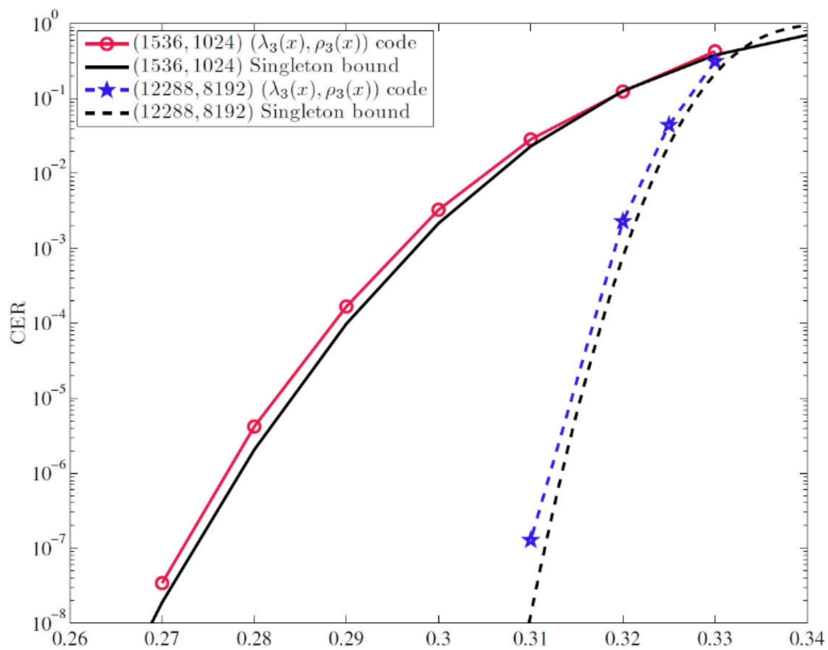


Figure 13: CCSDS - CER vs. erasure probability on the packet erasure channel for Two Codes from the Ensemble (Fig 2-4) [6]

Where the y-axis is the CER, Codeword Error Rate or Bit Error Rate; it represents the amount of information that even after decoding is lost. In our scenario we will be using BLER, BLock Error Rate, as it is of interest to study the loss of entire space packets, which are blocks of bits, and not its components, its bits. In the legend of the figure, there are two code candidates, one denoted as $(1536, 1024)(\lambda_3(x), \rho_3(x) code)$ and the other as $(12288, 8192)(\lambda_3(x), \rho_3(x) code)$; the Singleton bound represents the theoretical limit of this codes. Both have a coding rate of $R = 2/3$ but have different amount of input bits. As one can see the $(1536, 1024)$ code has a lower performance where information will begin to be lost after an erasure probability of ~ 0.27 and the $(12288, 8192)$ has one of ~ 0.31. Theoretically, the most ideal erasure correcting code given this coding rate would not lose any useful data before an erasure probability of $\epsilon = 1 - R = 1/3$. This however would require a lot of data, where in our scenario the input and outputs represent in this standard needs downlink bit rates of a few Megabits per seconds to avoid latency. Nonetheless, this concept can be scaled down with expected lower performance to better accommodate the intermittent scenario of this research. This scaled down code must also be easy to implement, simple to use and over all be a free software. Thus, a very promising candidate for this code is the zfec.

3.3.1 zfec

The zfec is an efficient, portable erasure coding tool which generate redundant blocks of information such that if some of the blocks are lost then the original data can be recovered from the remaining blocks. This package includes command-line tools, C API, Python API, and Haskell API. It utilizes the RAID-5 algorithm, where RAID stands for ”redundant array of independent disks” or ”redundant array of inexpensive disks”. It is used to created reliable data stores between the disks of the computers for data loss prevention and recovery schemes. The RAID-5 algorithms consists of block-level striping with distributed parity. Upon failure of a single drive, subsequent reads can be calculated from the distributed parity such that no data is lost [9]. The following figure shows the layout of this algorithm, where each color represents the group of data blocks and associated parity block (a stripe):

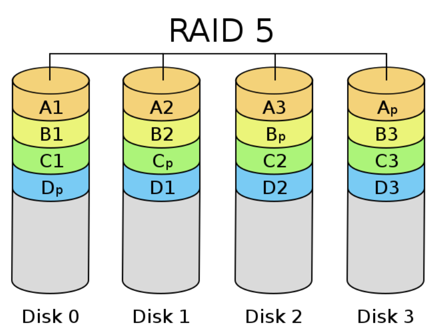


Figure 14: Diagram of a RAID-5 layout

The zfec requires 3 inputs:

- The useful data ($data_k$).
- The number of useful data units (k).
- The number of redundanta data units (n).

For this algorithm to work, the input useful data must be a list of data units with equal length. This is thus an algorithm which is not flexible like the F-IRA code. However, because of this drawback, the complexity of implementation is much lower.

The CCSDS recommends encoding the space packets for erasure correction; however their communication channel considers fading times of $2 - 6\,ms$ as shown in sub-section 3.2. Our scenario may have fading times of a seconds, which drastically changes the necessary encoding needed. Thus, two encoding schemes using zfec are proposed:

1. **Space Packet Encoding**: the content of the space packet will be the input for the encoder.
2. **Transfer Frame Encoding**: the content of the transfer frame, including the idle packet, will be the input for the encoder.

Ideally, an optimal system which calculates the fading times would help in determining which of these two to choose from. As fading times of a *few milliseconds* would best utilize the *Space Packet Encoding*, or on the other hand fading times of a few seconds would best utilize the *Transfer Frame Encoding*. The performance of both schemes will be presented in the sub-section 4.3.

3.4 Future developments

Due to time constraint, many issues are still left unsolved and there are several missing implementations of concepts. The following is list of these:

- **Trajectories:** The trajectory of the satellite does not incorporate all 3 trajectory modes: "spinning sun pointing", "compass" and "SAA measuring mode.
- **Bit erasure:** The current BEC considers the erasure of blocks/space packets in its intermittency model. A real model would consider whether individual bits are sent of lost. The Gilbert Elliot model: finite 2-state Markov Chain may lead to a more realistic model of bit erasure and not packet erasure.
- **Other FEC candidates:** Only the zfec algorithm was studied, the LDPC, Turbo, Raptor and Sliding Window codes may prove to be better solutions.

4 Results

This section will show the results of section 3 using the data from table 1 and a simulation time of 1 day (86400 s). To acquire accurate enough data, a time step of 0.1 s is used as this is roughly the amount of time it takes for a transfer frame to be sent as shown in the following calculations:

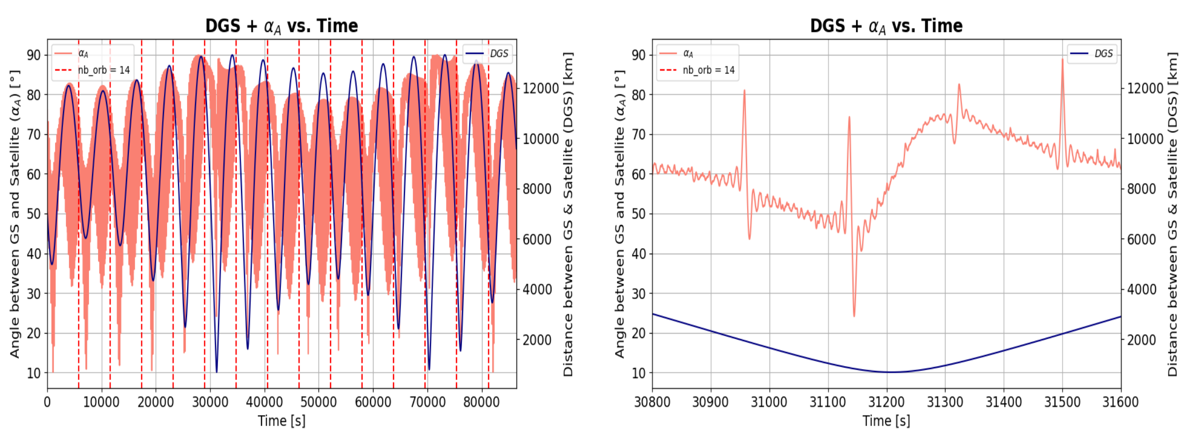
$$\#_{SP} = \lfloor \frac{TF_{size}}{B_{SP}} \rfloor = 17\,SP \qquad t_{stp} = t_{TF} = \frac{TF_{size}}{br} = 0.105\,s =\sim 0.1\,s$$

To mimic the "spinning sun pointing" mode, the z-axis of the satellite ($\hat{Z}_B$) was aligned with the x-axis of the ECI frame. Only a spin about the satellite's z-axis was considered as this is the ideal motion for measuring data from the SAA. A range of simulations was done with different spin parameters, ranging from 0 $\frac{\circ}{s}$ to 15 $\frac{\circ}{s}$ at intervals of 1 $\frac{\circ}{s}$. This data's statistics will then be studied to determine the parameters which affect the intermittency. This information was obtained from Table 1; although the range is from 2 $\frac{\circ}{s}$ to 15 $\frac{\circ}{s}$ as per the table, it was considered of interest to see what would happen if there were no spin along the z-axis.

4.1 Orbital Mechanics

The statistics results from the different angular velocities seems to provide valuable information, however no conclusions have been able to be drawn from it. Nonetheless, this statistical analysis will be shown in Annex I.II.

The following figures show the results of average of the different angular velocities from 16 simulations, each with different spin parameters ranging from $0[^\circ/s]$ to $15[^\circ/s]$, that use the orbital parameters shown in Table 2:

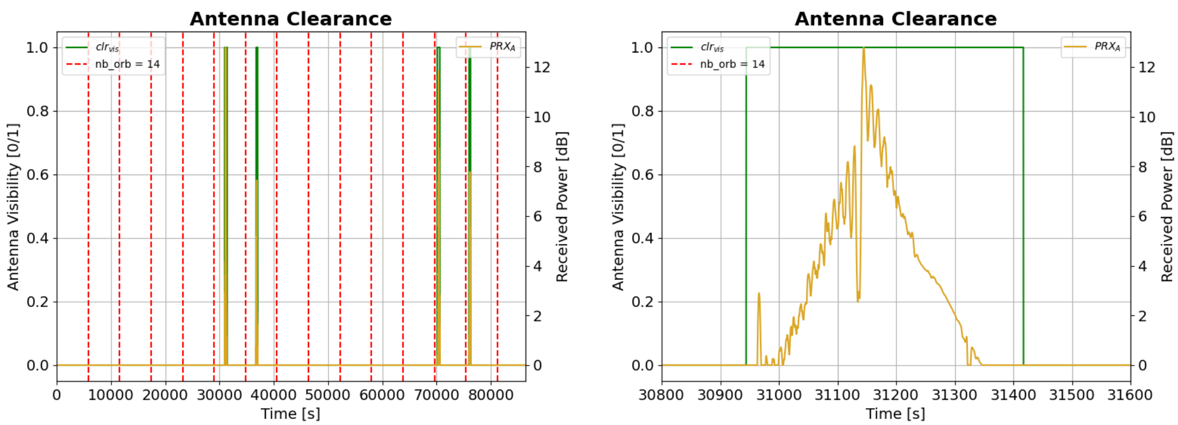


(a) 1 day

(b) The visibility window of the 5^{th} orbit in the day

Figure 15: Distance between ground segment and satellite's antenna (DGS) and angle between ground segment and satellite's antenna (α_A)

As one can see the navy blue line represents the DGS, which is independent of the spin parameter, as it is entirely made up of the orbital parameters. On the other hand, the salmon colored line represents the average of the angles between the ground segment and the satellite's antenna from all the different spin parameters. The following figures show the results when the 3 requirements for communication "clearance" from sub-section 3.1 are met:



(a) 1 day **(b)** The visibility window of the 5^{th} orbit in the day

Figure 16: Cleared received power from satellite's antenna

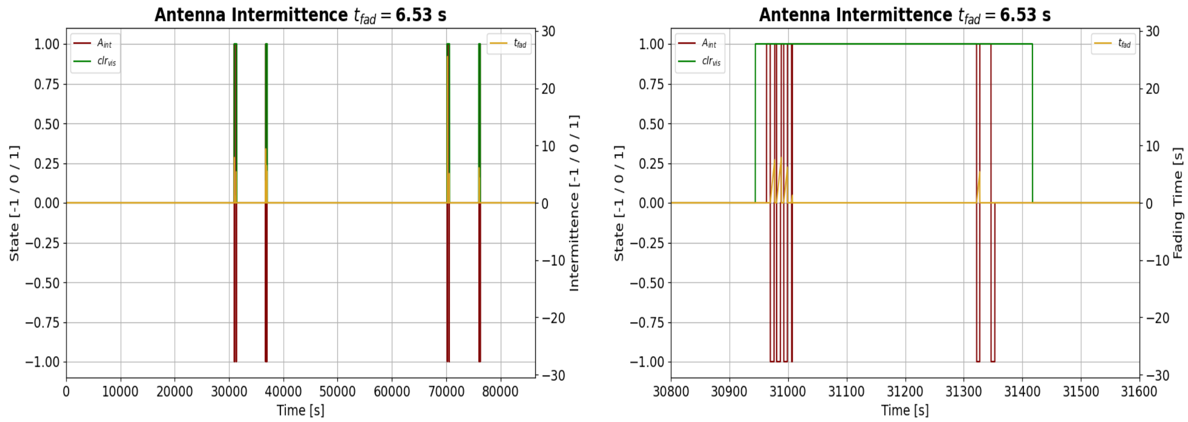
The green line, represents the visibility window when the first 2 requirements for the "communication clearance", fundamentally it represents when one can view the satellite from the ground segment. The gold line represents the power received by the ground segment in this time window. These results are in-line with the previous investigations done of 4 visibility windows per day with a total visibility time of 18.53 min.

4.2 Binary Erasure Channel

The ground segment's received power from the antenna of the satellite (P_{RX}) is then plugged into the algorithm for intermittency explained in sub-section 3.2 and illustrated in Annex I.II. First a lag time of 75 seconds is selected for the intermittency. This is because fadings time greater than this are unrealistic even in this scenario. This value was heuristically chosen as it was noted that no fading time could be larger than this. The intervals when the received power dropped below 0 s are calculated. These are then averaged out and thus we obtain a mean fading time per day, this results in the following number:

$$\bar{t}_{fad} = 6.526\, seconds$$

The resulting intermittency by plugging in a lag time equal to the above and the fading time are shown in the following figure:



(a) 1 day

(b) The visibility window of the 5^{th} orbit in the day

Figure 17: Antenna Intermittency as per the calculated fading time

Again, the green line represents the visibility window. The maroon line shown is the antenna intermittency, it helps in understanding when data is erased (-1), not sent (0), and received (1). Lastly, the gold line is the fading time calculated placed at the final interval before another detection is made.

With all of this information, an estimation can be made of the BEC parameter of erasure probability ϵ_{est} and coding rate R_{est}. To do so, the following equation is used, which is a reinterpretation of equation 7 which better suits the results of the antenna intermittency:

$$\epsilon_{fad} = \frac{t_{dis}}{t_{con}} \qquad R_{fad} = 1 - \epsilon_{est}$$

Where t_{dis} is the time of disconnection which is when information is lost and t_{con} is the time of connection which is the allocated window for information to be sent. These equation result in the following values:

$$R_{fad} = 0.836 \qquad \epsilon_{fad} = 0.164$$

These values, are lower than the minimum in the range of erasure probabilities given by the information in sub-section 3.2; where the lowest was of $\epsilon_{est,min} = 0.172$. What this represents is that in the intermittent model where a mean fading time of $6.526s$, only 16.4% of data is lost. Coding rates of $R = \frac{5}{6}$ or lower would generate enough redundant information to mitigate this erasure. This fading time however, if continuous, leads to the loss of more than 63 complete transfer frames as shown in the following calculation:

$$TF_{lost} = \lceil \frac{t_{fad}}{t_{TF}} \rceil = 63\, TF$$

Even though, this ultimately represents an erasure probability of 0.164, this data would be completely erased should only the space packets be encoded. This represents an unrealistic scenario where too much data is lost. Thus, a new BEC parameter must be made.

A new scenario will be considered, where the visibility window will now be considered the total time in which messages can be sent, similar to that shown in sub-section 3.2. This will be done for each of the 4 visibility windows in a day, where the final result will the mean of these calculations. Thus, the following calculations are made from the results of the intermittency:

$$\bar{R}_{vis} = \frac{\bar{t}_{con}}{\bar{t}_{vis}} = 0.739 \qquad \bar{\epsilon}_{vis} = 1 - R_{vis} = 0.261$$

Where $\bar{R}_{vis}$ represents the mean coding rate according to the visibility window. The $\bar{t}_{con}$ is the mean connected time per day as per the visibility window, and the $\bar{t}_{vis}$ is the mean visibility time per day. Lastly, the $\bar{\epsilon}_{vis}$ represents the mean erasure probability according to the visibility window which is the BEC parameters searched for. This represent a much harsher erasure probability to overcome, however the main objective is to best mitigate this problem.

4.3 Erasure Correcting Code

This sub-section will present the performance of candidate code, zfec, for space packet encoding and transfer frame encoding as per explained in sub-sub-section 3.3.1. To better understand the performance of the the zfec the uncoded stream of information through a BEC is also plotted as a black line with "+" markers; this will provide an idea of how much information is being recoeved despite erasure. The x-axis of the following plots will be an increasing erasure probability ϵ from 0 to 0.5, and the y-axis will be the BLock Error Rate (BLER) which represents how many blocks of bits (in this case, space packets or transfer frames) are ultimately lost.

4.3.1 Space Packet Encoding

For this encoding scheme, it was necessary to know how many space packets must be encoded. The number of space packets, $\#_{SP}$, is considered as the number of total data units allowed, n; which consists of the useful, k and redundant r data. Additionally, multiple coding rates will be examined such as to analyze better the code's performance. Lastly, the mitigated fading time, $t_{fad,mit}$, will also be studied, this represents how much loss of data, in terms of time, can occur before the data cannot be decoded. This is all represented in the following equations:

$$B_{SP} = 64\, bytes \qquad \#_{SP} = \lfloor \frac{TF_{size}}{B_{SP}} \rfloor = 17\, SP \qquad t_{fad,mit} = \frac{r * B_{SP}}{br}$$

The following table shows the coding rates, R, selected and their respective number of useful, k, total, n, redundant r space packets, and the mitigated fading time $t_{fad,mit}$ measured in ms:

R []	k [# SP]	n [# SP]	r [# SP]	$t_{fad,mit}$ [ms]
2/3	10	15	5	30
3/4	12	16	4	24
5/6	10	12	2	12
7/8	14	16	2	12

Table 3: Parameters from coding rates for space packet encoding

The following figure illustrates the performance of encoding the content of individual space packets with the information from the previous table as its input:

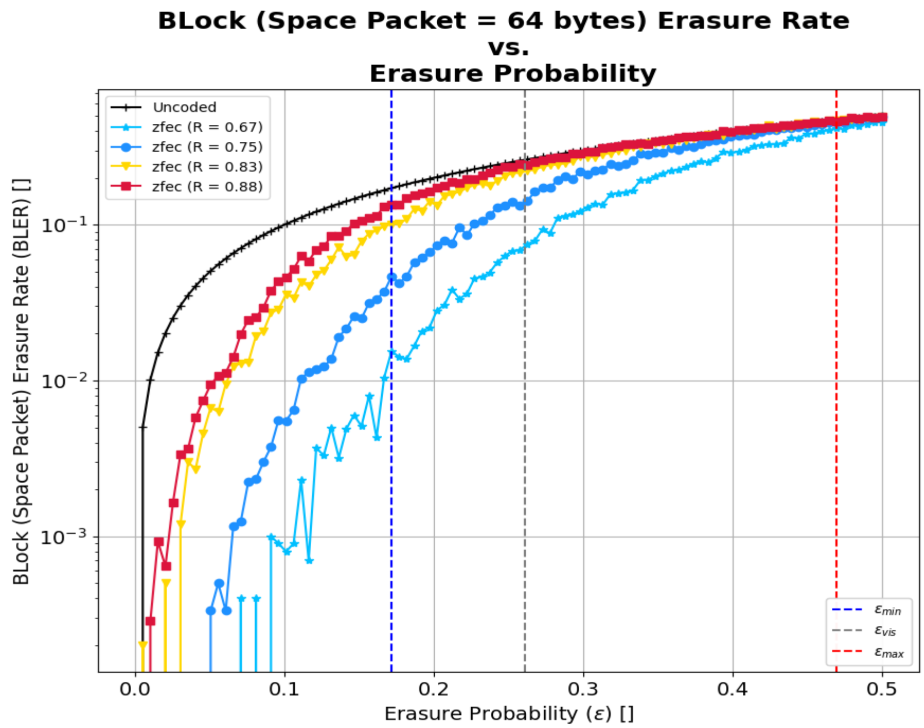


Figure 18: zfec space packet encoding performance

To interpret the data presented in the previous graph, the amount of actual data required to send the daily bits, b_{daily}, needs to be calculated in terms of how many space packets of useful data are allowed. To do so, the following equation is used:

$$\#_{SP,tot} = \lceil \frac{b_{daily}}{SP_{size}} \rceil \qquad b_{daily,SP} = \lceil \frac{\#_{SP,tot}}{k} \rceil * TF_{size} \tag{9}$$

These formulas define $\#_{SP,tot}$, the number of total space packets needed to send the daily bits, and $b_{daily,SP}$ the size of the data needed to send all useful daily bits packaged inside transfer frames. Thus, we obtain the following graph when considering space packet of lengths 32 bytes, 64 bytes and 128 bytes as well as a table for the performance of the zfec with candidate coding rates:

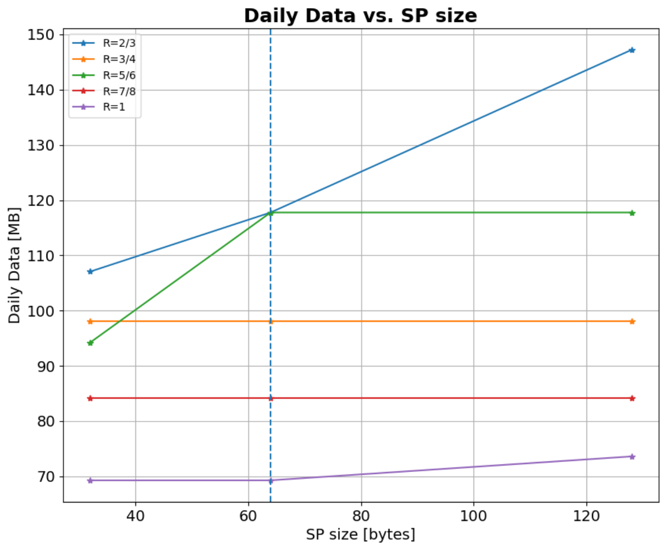


Figure 19: Daily bits estimate depending on number of useful space packets

	R			
	2/3	3/4	5/6	7/8
BLER ($\epsilon = 0.261$) []	0.075	0.143	0.221	0.242
$b_{daily,SP}$ **[Mb]**	117.771	98.147	177.771	84.126

Table 4: zfec with space packet encoding performance at $\bar{\epsilon}_{vis}$ and $b_{daily,SP}$

From this, we can conclude that a lower coding rate will yield the best performance at the cost of a lot more data being sent. The coding rate $R = 2/3$ presents a performance of 7.5% erasure of data and daily bits size of 177.771 Mb when the erasure probability reaches the scenario explained in sub-section 4.2. On the other hand, a coding rate of $R = 7/8$ has a performance of 24.2% erasure of data with a smaller daily bits size of 84.126 Mb. It is noteworthy that coding rates $R = 2/3$ and $R = 5/6$ have an equal amount of data needed to be sent, however the mitigated fading time for $R = 5/6$, which is of 0.012 s, is less than half of the one for $R = 2/3$, which is 0.030 s; thus it there is no reason to pick a coding rate of $R = 5/6$ over $R = 2/3$, unless other considerations are made.

4.3.2 Transfer Frame Encoding

This encoding scheme has the objective of mitigating the mean fading time per day of 6.53 s found in sub-section 4.2; to do so, the redundant data created must be equal to the lost transfer frames TF_{lost}. The following table illustrates the candidates and their respective number of useful transfer frames, k, size of input data, $b_{k,SP}$, and the total transfer frames, n:

R []	**k [# TF]**	$b_{k,TF}$ **[Mb]**	**n [# TF]**
2/3	126	1.124	189
3/4	189	1.686	252
5/6	315	2.810	378
7/8	441	3.934	504

Table 5: Parameters from coding rates for transfer frame encoding

To reiterate, the number of redundant, r, transfer frames is equal to 63 TF for each of these candidate coding rates, as this is the objective of this encoding scheme. It is noteworthy that the coding rates higher than $R = 2/3$, all the the number of useful transfer frames greater than $150TF$ which is equal to a total of 1.338 Mb of useful information. We are considering this as the limit of how much input data this can handle as this is amount creates a latency far too great for an adequate communication channel; thus, only a coding rate of $R = 2/3$ is considered. The following figure illustrates the performance of encoding the content of individual transfer frames with this coding rate, afterwhich a table showcasing the performance results is shown as well:

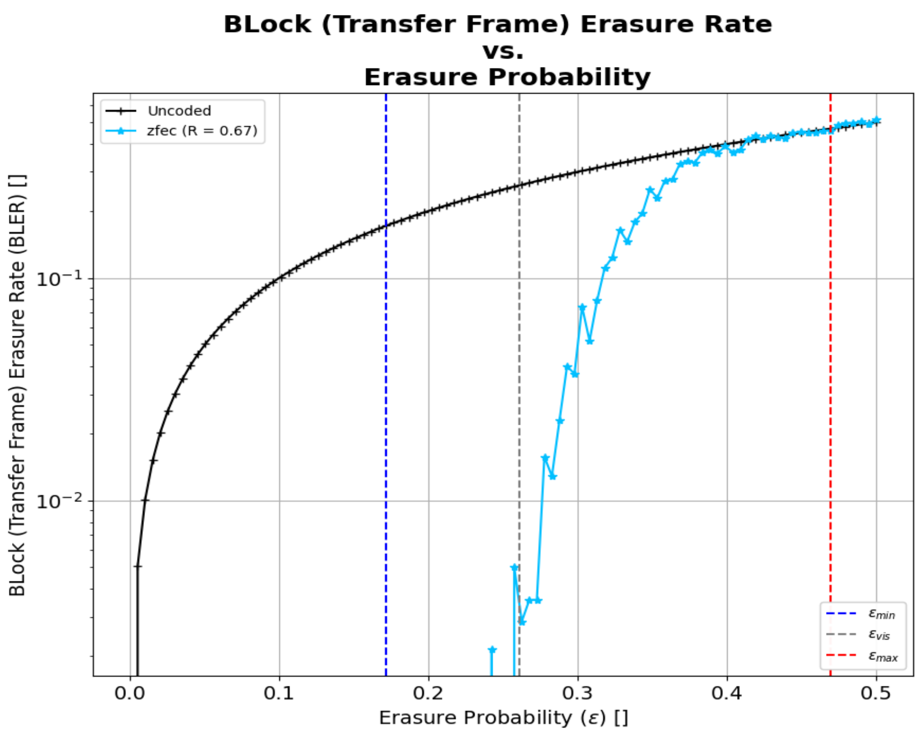


Figure 20: zfec transfer frame encoding performance

Coding Rate (R)	**BLER** ($\epsilon = 0.261$)	$b_{k,TF}$ **[Mb]**
2/3	0.003	1.124

Table 6: zfec with space packet encoding performance at $\bar{\epsilon}_{vis}$

The performance illustrates that the required erasure probability needed to be mitigated will only create a statistical average loss of information equal to 0.3% of Transfer Frames; which is a significant improvement over the uncoded stream loss which is of 26.1%. Of course, the input data of $1.124\,Mb$ may cause latency as the CubeSat must generate this much information before beginning its encoding procedures. The following section will explain what considerations are missing to improve this investigation.

5 Future Developments

The following is a list of the missing features that should be implemented:

1. **Simulation of Tumbling:**
   - ***Satellite Attitude Modes:*** The satellite has 3 attitudes modes: the sun-pointing mode (of 100 minutes), the compass mode (for 30 minutes every 2 orbits) and the SAA measuring mode (for 20 minutes); they should all be implemented in the astrodynamics of the satellite.
   - ***Earth's Magnetic Field:*** Modelling of the EMF for the satellite attitude modes.
   - ***Atmospheric Disturbances to Radio Signals:*** Possible fading times of a ms may be caused by atmospheric disturbances, this should be modelled.
   - ***Longer Time Frame:*** An entire year should be simulated for more statistical data.
2. **Erasure Correcting Code:**
   - ***Comparison with COP-1:*** Compare the zfec performance with the currently selected uplink data exchange protocol of COP-1.
   - ***Other Candidate Codes:*** Determine the performance of other open source Erasure Correcting codes, such as: OpenFEC, etc. Additionally, other kinds of codes should also be studied, such as: LDPC, Turbo Codes, Sliding Windows, etc.
   - ***Latency Studies:*** Determine the latency in the on-board computer caused by the required input data for the encoding scheme featuring space packets or transfer frames.
3. **Test Rig:**
   - ***Digital experiment:*** Create a digital network with the defined intermittency by the visibility erasure probability.
   - ***Physical experiment:*** Embed the Erasure Correcting to the on-board computer and acquire experimental data.

6 Conclusion

In terms of the simulation of tumbling, which is a merging of astrodynamics and antenna modelling, it was determined that a possible way of measuring erased data is through a fading time. It was calculated through a simulation of the length of 1 day (86400 seconds discretized with 0.1 seconds) that there are 4 visibility windows; in these windows the received power from the satellite and ground segment leads to the calculation of a mean erasure probability as per the visibility window,ϵ_{vis}, of 0.261 and of a mean fading time per day, $\bar{t}_{fad}$, of 6.53 seconds. These two parameters show the necessary performance erasure correcting codes must surpass to ensure that no data can be lost.

In terms of the erasure correcting code, the open source python code zfec is a promising candidate for encoding and decoding through an intermittent communication channel imposed by the satellite's aerodynamics and antenna performance. Two encoding schemes are proposed, the encoding of space packets and the encoding of transfer frames; the first is a standardized approach already recognized by the CCSDS, however the latter is not. The reason for which transfer frame encoding into erasure correcting codes, is that the data budgets of normal medium or large satellite is of MB in size; whereas the data budget of the CubeSat in question is of a few kb. Thus, a new approach to data encoding must be developed if one were to implement such kinds of codes into a CubeSat.

To model the intermittent communication channel, a binary erasure channel is used, it uses the erasure probability previously mentioned. The received information from this channel is the input into the erasure correcting code candidate zfec. The performance of this candidate is studied for the encoding scheme for space packets, with a size of 64 bytes, and for transfer frames, with a size of 1115 bytes. The consideration for the appropriate header and data structure is also taken into account for both encoding schemes. It is also important to note that the daily bits needed to be sent is considered to be 67.6 Mb.

The performance of the zfec is measured by the BLER found at the erasure probability equal to that to the erasure probability as per the visibility window. It is of interest to determine the capability of the code in a very pessimistic scenario. In terms of space packet encoding, the coding rate of $R = 2/3$ has the best overall performance with a BLER of 0.075 or a loss of information of 7.5% with a daily data size of 117.771 Mb. On the other hand, the coding $R = 7/8$ has an overall performance with a BLER of 0.242 or a loss of information of 24.2% with a daily data size of 84.126 Mb. Overall, one would choose a coding rate of $R = 2/3$ for better performance a the cost of nearly twice the amount of daily data to send, while a coding rate of $R = 7/8$ yields a lesser performance but with creating less total data. Lastly, the transfer frame encoding scheme with a coding rate of $R = 2/3$ leads to a performance of a BLER of 0.003 or a loss of information of 0.3% and an input data size of 1.124 Mb. Lastly, the encoding of space packets and transfer frames ensure the mitigation of fading times of a few milliseconds and seconds respectively.

References

[1] O. Kodheli, E. Lagunas, N. Maturo, *et al.*, “Satellite communications in the new space era: A survey and future challenges,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 70–109, 2020.

[2] J. R. Heirtzler, “The future of the south atlantic anomaly and implications for radiation damage in space,” *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 64, no. 16, pp. 1701–1708, 2002.

[3] M. A. A. Madni, S. Iranmanesh, and R. Raad, “Dtn and non-dtn routing protocols for inter-cubesat communications: A comprehensive survey,” *Electronics*, vol. 9, no. 3, p. 482, 2020.

[4] P. Muri, *Delay Tolerant Networking for CubeSat topologies and platforms.* University of Florida, 2013.

[5] S. Schaire, B. Horne, J. Schier, *et al.*, “Nasa near earth network (nen) and space network (sn) cubesat communications,” in *14th International Conference on Space Operations*, 2016, p. 2598.

[6] C. O. Book, *Erasure correcting codes for use in near-earth and deep-space communications*, 2014.

[7] G. Haßlinger and O. Hohlfeld, “The gilbert-elliott model for packet loss in real time services on the internet,” in *14th GI/ITG Conference-Measurement, Modelling and Evalutation of Computer and Communication Systems*, VDE, 2008, pp. 1–15.

[8] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Efficient erasure correcting codes,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[9] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, “Raid: High-performance, reliable secondary storage,” *ACM Computing Surveys (CSUR)*, vol. 26, no. 2, pp. 145–185, 1994.

[10] H. Curtis, *Orbital mechanics for engineering students.* Butterworth-Heinemann, 2013.

I Annexes

I.I Simulation of Tumbling

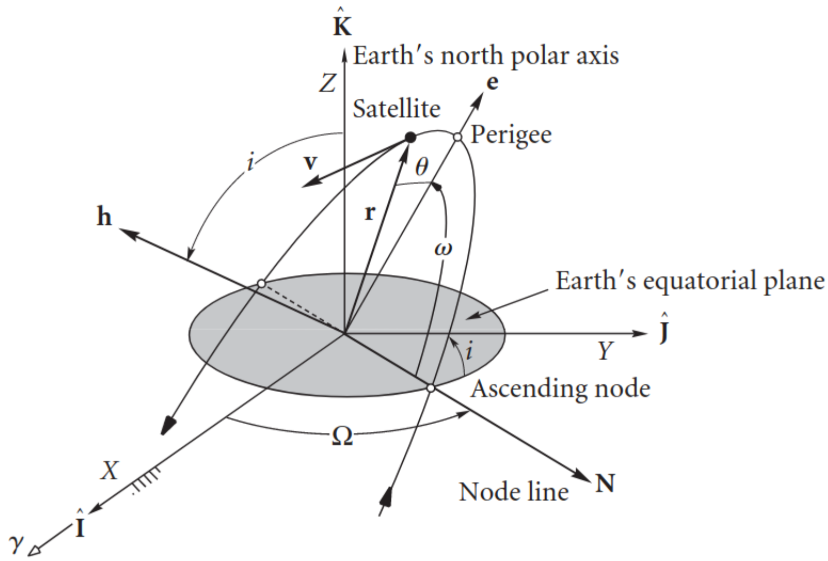


Figure 21: Geocentric equatorial frame and the orbital elements [10]

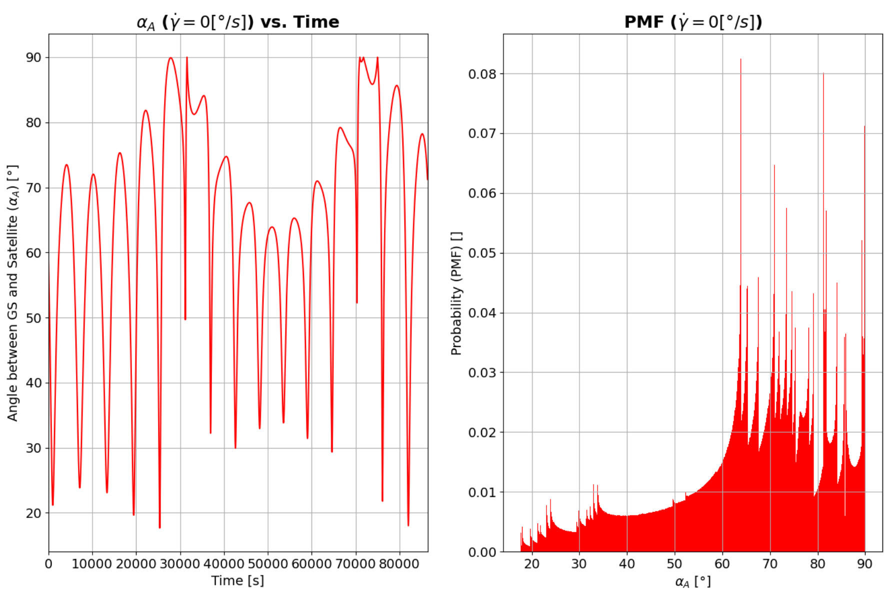


Figure 22: Probability Mass Function of the angle between satellite and ground segment with no spin

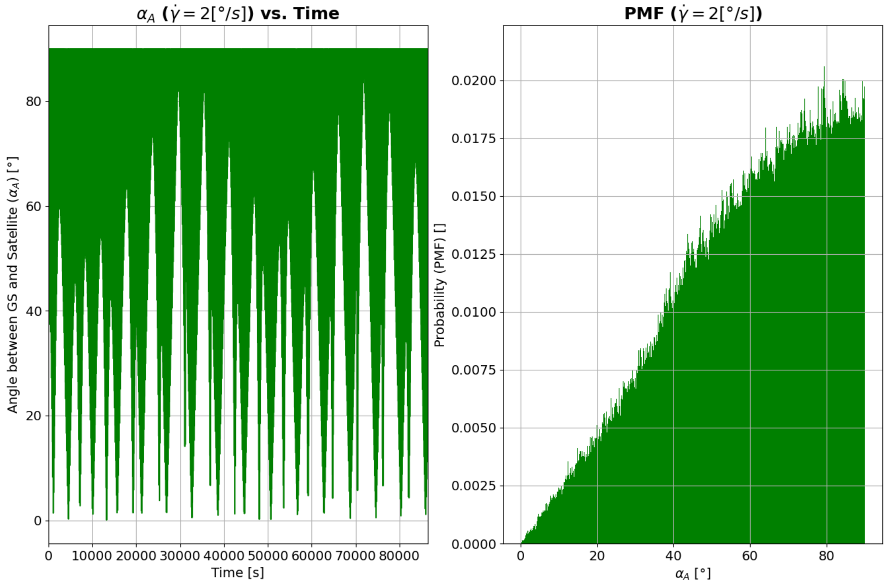


Figure 23: Probability Mass Function of the angle between satellite and ground segment with 2 $\frac{\circ}{s}$ spin

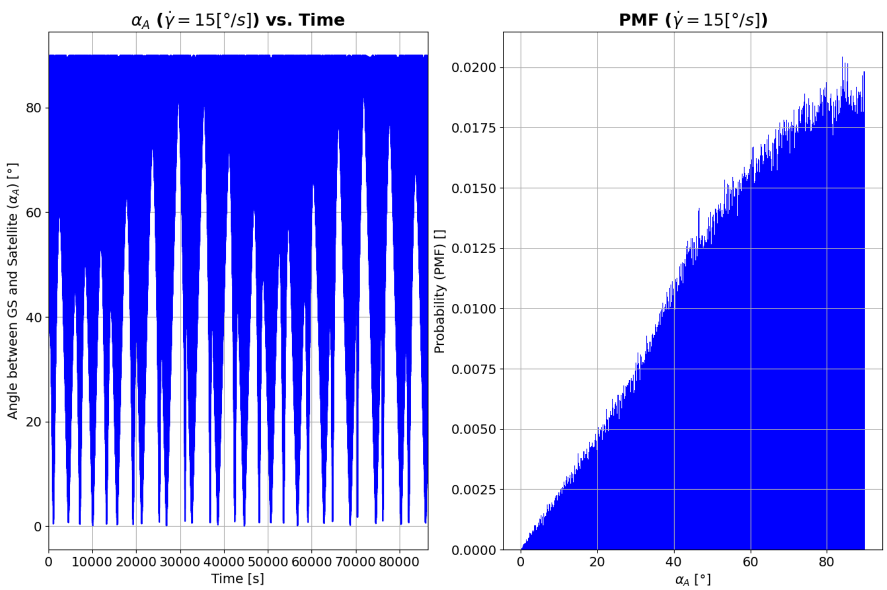


Figure 24: Probability Mass Function of the angle between satellite and ground segment with 15 $\frac{\circ}{s}$ spin

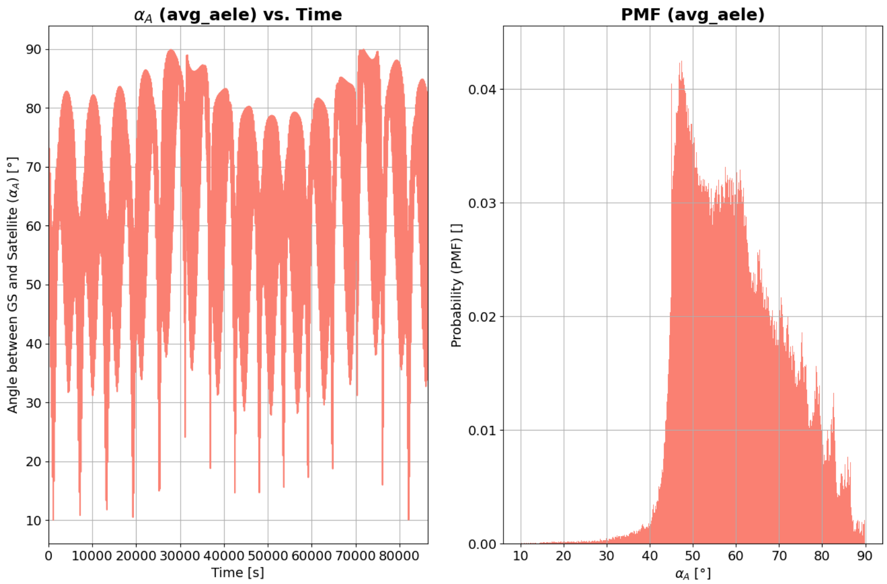


Figure 25: Probability Mass Function of the angle between satellite and ground segment from average of angular velocities

I.II Erasure Correcting Code

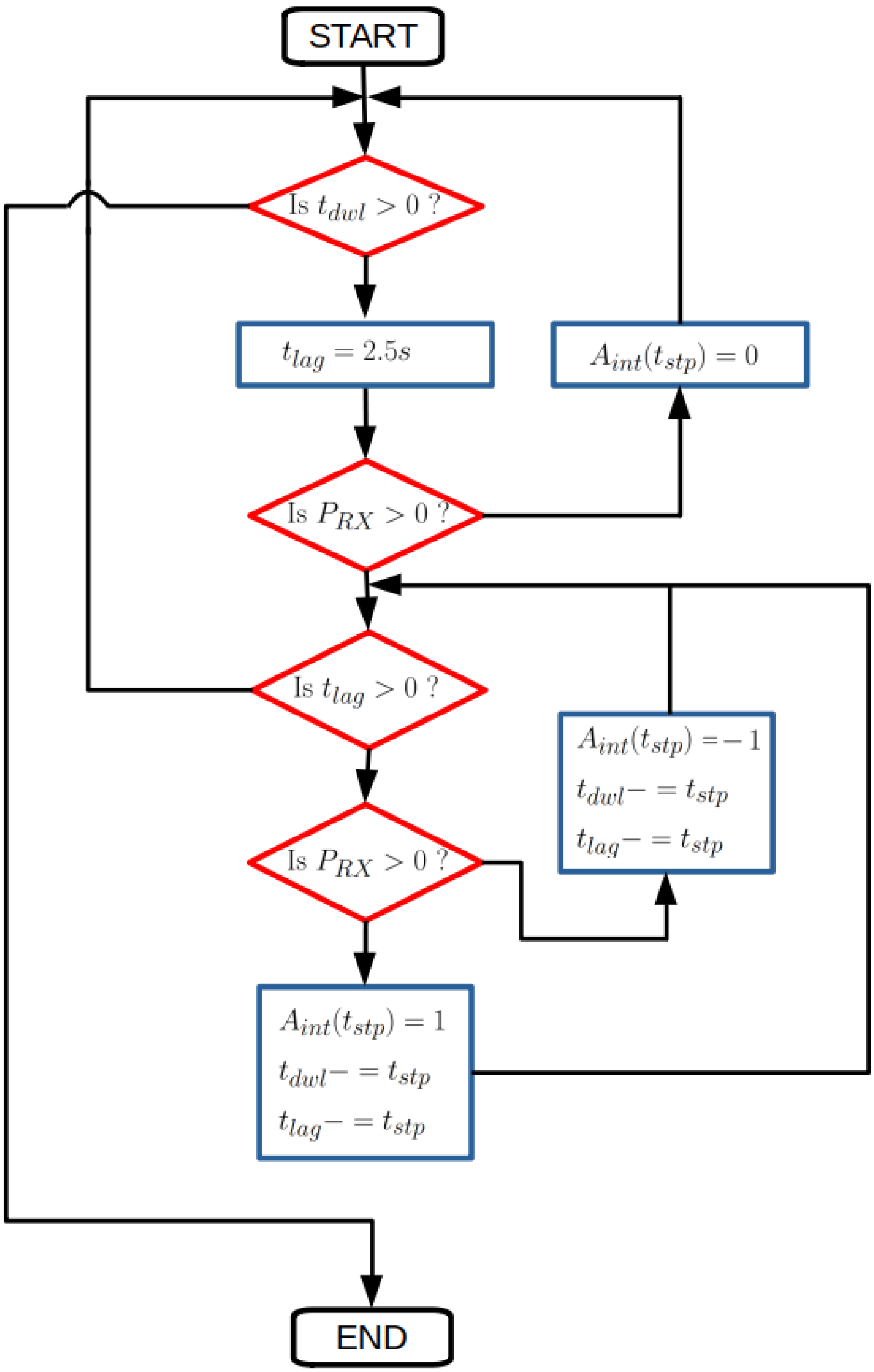


Figure 26: Flowchart for lag time intermittency model