IAC-22,A6,IPB,2,x68374

Light Curve Inversion for Attitude Reconstruction of Tumbling Space Debris

Matteo Gallucci, Prof. Mauro Massari, PhD Candidate Riccardo Cipollone, PhD Candidate Andrea De Vittori, Research Fellow Giovanni Purpura

Abstract

In recent years, the problem of space pollution has become more and more important in the space field. Space debris monitoring and tracking is fundamental to keep space accessible for human operations. Within the SST framework, observations through optical telescopes allow space agencies to directly observe an orbiting object and study the corresponding trail (streak or tracklet) acquire by the detector. From these streaks, a post-processing phase extracts the in-time light variation reflected by the object (light curve). Light curves embed key features such as the debris orbital parameters and attitude state. Particularly, the information about the latter is crucial when it comes to actively removing debris or predicting how it will behave when/if it will re-enter the atmosphere. In the literature, attitude motion estimation relies on methods developed for the determination of asteroids and variable stars rotational period (period finding algorithms). More complex approaches, such as light curve inversion, allow to reconstruct the attitude state by simulating the dynamics of a geometry-defined object and the way it reflects the light coming from the Sun. In this work, a variation of the latter is used to extract the angular velocity of a generic tumbling object approximated by a simplified geometry. Its dynamics evolution is leveraged to build synthetic light curves through a basic reflection model. They are used to fit the real ones extracted from FITS images solving an optimization problem that outputs the attitude state of the target. Moreover, synthetic images containing streaks are generated to validate the pipeline with the same simulation process. Considering the results, the algorithm correctly extracts and analyses streaks from both real and synthetic images. For moderate rotational regimes, the object angular velocity norm and out-of-plane component are accurately retrieved. On the other hand, the in-plane components are sometimes inverted and, in general, their optimization is less accurate. Furthermore, the relative error between optimized and simulated angular velocity norm is computed for multiple runs of the algorithm on the same image. For simulated angular velocities less than 10 °/s, the error is kept below 5% in 80% of the cases, whereas with faster rotations (angular velocities above 90 °/s), its magnitude and statistical behaviour worsen. In general, for high rotational regimes, the overall validity of the results slightly decreases, albeit remaining acceptable for a preliminary analysis of the target tumbling motion.

Keywords: space debris, optical telescope, light curve inversion, attitude motion, FITS.

Nomenclature

ω_x [°/s]	Angular velocity along x
ω_y [°/s]	Angular velocity along y
ω_z [°/s]	Angular velocity along z
$\boldsymbol{\omega}$ [°/s]	Angular velocity vector
I_{mref} [-]	Control star instrumental magnitude
m_{ref} [-]	Control star reference magnitude
$\mathbf{A_{B/N}}$ [-]	Direction cosine matrix
$\hat{\mathbf{S}}$ [-]	Debris-Sun versor
$\boldsymbol{I}$ [kg m^2]	Diagonal inertia matrix
FunTol [-]	Function tolerance
A_{var} [-]	Illuminated-projected area variation array
$\boldsymbol{\omega_0}$ [°/s]	Initial angular velocity
ω_{0x} [°/s]	Initial x angular velocity
ω_{0y} [°/s]	Initial y angular velocity
ω_{0z} [°/s]	Initial z angular velocity
$\|\|\boldsymbol{\omega_0}\|\|$	Initial angular velocity norm
θ_0 [°]	Initial pitch angle
ψ_0 [°]	Initial roll angle
ϕ_0 [°]	Initial yaw angle
$\boldsymbol{angles_0}$ [°]	Initial yaw-pitch-roll angles
$\boldsymbol{var_{input}}$	Input variable for optimization
ω_{lim} [°/s]	Limit angular velocity
MaxGen [-]	Maximum number of generations
k [-]	Multiplicative factor
$\hat{\mathbf{N}}$ [-]	Normal to the plate
$\hat{\mathrm{O}}$ [-]	Observer-Debris versor
β [°]	Observer-Normal angle
θ [°]	Pitch angle
$\mathrm{L_{var}}$ [-]	Pixel intensity variation
A_{var} [-]	Plate area variation
err [-]	Relative error for the angular velocity norm
ψ [°]	Roll angle
sat [-]	Saturation threshold
n [-]	Shift
A_{ill} [-]	Simply illuminated area
A_{proj} [-]	Simply projected area
$\boldsymbol{mag_{sim}}$ [-]	Simulated magnitude light curve

σ [°/s]	Standard deviation
μ [°/s]	Statistical mean of the angular velocities norm
M [-]	Streak apparent magnitude
I_m [-]	Streak instrumental magnitude
γ [°]	Sun-Normal angle
ϕ [°]	Yaw angle

Mathematical Notation

- Scalar: a
- Vector: $\boldsymbol{a}$
- Matrix: $\boldsymbol{A}$

Acronyms/Abbreviations

ADR: Active Debris Removal
ASI: Agenzia Spaziale Italiana
Az: Azimuth
B: Body
CDF: Cumulative Distribution Function
DEC: Declination
ECI: Earth-Centred Inertial
EE: Euler's Equations
El: Elevation
FITS: Flexible Image Transport System
FOV: Field Of View
GA: Genetic Algorithm
LEO: Low Earth Orbit
PdM-MITE: Pratica di Mare - MIlitary TElescope
RA: Right-Ascension
SAP: Streak Analysis Pipeline
SST: Space Surveillance and Tracking
TIG: Tracklet Image Generator

1. Introduction

The number of space debris, its combined mass and area has been steadily increasing since the beginning of the space age [1]. A worsening of this situation could lead to an overcrowding of the Low Earth Orbit (LEO) regime, triggering collision chains that would hinder the safety of space missions. The monitoring and tracking of resident space objects, performed by space agencies' SST (Space Surveillance and Tracking) segments, proves fundamental in this regard. Particularly, direct observation of a debris through the collection of the sunlight reflected from its surface, allows to record in a short video or in a long-exposure image the variably illuminated trail left by its motion (streak or tracklet). From the extraction of light intensity over time (light curve), angular and astrometric information can be retrieved and used to determine the orbit and the attitude state of the object. Information about the latter would hugely benefit the predictions on how the debris will re-enter the atmosphere and the different active debris removal (ADR) techniques.

The main methods used to determine the space debris attitude state inherit most of their features from studies on light curves of asteroids and variable stars. These techniques are primarily able to determine the rotational period of the object (period finding algorithms), while only in few cases they can fully characterise the overall attitude state (epoch method). Conversely, a full state reconstruction is granted by light curve inversion methods [2]. They consist in a fit of the real light curve with a simulated one is performed taking as input the attitude state guess; then, the difference between real and simulated signal is minimized over attitude states. In this work, a variation of light curve inversion methods is employed to determine the space debris attitude state. Since the overall work is intended to be applicable in the first stages of the study of the motion of an unknown object, the light curve simulation is performed according to simplified shape and reflection model. In addition, both real and simulated FITS (Flexible Image Transport System) images are considered.

2. Fundamentals

The space debris shape is simplified and approximated as a flat square plate. In this way, the body frame (B) rotation with respect to an inertial reference frame (Earth-Centred Inertial, ECI) can be represented by the torque-free Euler's rotation Equations (EE):

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} = \boldsymbol{I}\boldsymbol{\omega} \times \boldsymbol{\omega} \tag{1}$$

Where $\boldsymbol{\omega}$ is the vector of angular velocities (ω_x, ω_y and ω_z) and $\boldsymbol{I}$ is the diagonal inertia matrix.

In addition, the orientation of B with respect to ECI can be represented through the yaw (ϕ), pitch (θ), and roll (ψ) angles. They identify three consecutive rotations with a 321 sequence that is condensed in a rotation matrix ($\mathbf{A_{B/N}}$). The change in time of $\mathbf{A_{B/N}}$, in relation to the rotational dynamics in B (kinematics), is expressed by:

$$\dot{\boldsymbol{A}}_{B/N} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & -\omega_x & 0 \end{bmatrix} \boldsymbol{A}_{B/N} \tag{2}$$

To retrieve the kinematics and dynamics time evolution, Eq. (2) can be integrated alongside EE with specified initial conditions.

The target object reflects specularly the sunlight. The reflection is influenced by the relative positions between the Sun, the observer, and the debris (see Figure 1) that can be identified with two unit vectors:

- Observer-Debris unit vector ($\hat{\mathbf{O}}$): represented by the debris celestial coordinates (Right-Ascension (RA) and Declination (DEC))

- Debris-Sun unit vector ($\hat{\mathbf{S}}$): Sun position in ECI identified by its ephemeris

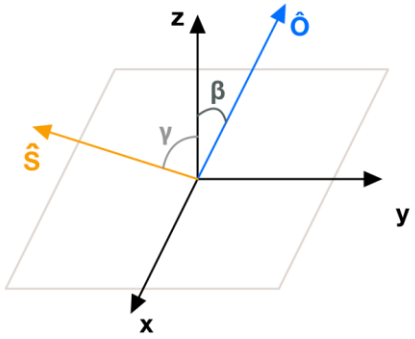


Figure 1: Reflecting Model

The plate area seen by the observer is given by the product of the simply projected (A_{proj}) and the simply illuminated (A_{ill}) areas, and it is influenced by the dynamics of the object. Therefore, the area variation expression is given by:

$$A_{var} = \left(\hat{\boldsymbol{O}} \cdot \hat{\boldsymbol{N}}\right)\left(\hat{\boldsymbol{S}} \cdot \hat{\boldsymbol{N}}\right) = \cos\beta \cos\gamma \quad (3)$$

Where $\hat{\mathbf{N}}$ is the normal to the flat plate face (either +z or -z). The area variation is different from zero only if it is illuminated ($\gamma < 90°$) and visible ($\beta < 90°$) at the same time. Eq. (3) identifies a light curve, and it is a scaled representation of how the light intensity reflected by the object changes over time (due to its dynamics). In fact, it is possible to link A_{var} to a pixel intensity – L_{var} - by introducing a shift (n) and a multiplicative factor (k), so that:

$$L_{var} = A_{var} \cdot k + n \quad (4)$$

If n is chosen as the mean pixel value, whenever $A_{var} = 0$, the streak is blended with the background and correctly non-visible.

In this way, the light curve is considered as a pixel intensity variation in time. A more classical approach is to consider the streak apparent magnitude (M) variation in time, that can be computed as:

$$M = m_{ref} + I_m - I_{ref} \quad (5)$$

To evaluate it, the procedure needs the reference magnitude (m_{ref}) of a control star in the Field Of View (FOV), the streak, and control star instrumental magnitudes (I_m and I_{mref}, respectively). For the former, star features can be queried from astronomical catalogues and matched with the stars in the image. The knowledge of the latter instead implies the study through aperture photometry of the flux received from the sources in the field [3]. This second technique is employed whenever the considered streak is saturating the detector pixels, thus losing the link between incoming radiation and pixel counts.

Once the real light curve is extracted, whether in terms of light intensity or magnitude, it is fitted with the simulated one by minimizing the difference between the two through genetic algorithm (GA). This algorithm mimics the Darwinian theory of survival of the fittest in nature and it is an iterative process in which the solutions are represented by individuals collected in a population. GA performs different operations on this population (selection, crossover, and mutation), and it evaluates the objective function value (fitness score) for the solutions at each generation. The worst solutions are discarded, and the process stops whenever the fitness scores rate of change is below a certain threshold (*FunTol*) or a maximum number of generations (*MaxGen*) is reached.

3. Angular Velocity Extraction Process

The proposed approach to determine the attitude state of the space debris requires three main computation steps:

1. Light curve extraction and post-processing
2. Light curve simulation through the dynamics and kinematics of a square flat plate
3. Optimization of the objective function, minimizing the difference between the two light curves

To validate the developed code, synthetic images are generated and analysed.

The images generation step regards the plate dynamics simulation, following the main physical concepts exposed in Sec. 2. The simulation, developed in MATLAB programming environment, is carried out by a function (*dynamics.m*) that takes as input an array ($\boldsymbol{var_{input}}$) containing:

- Three initial yaw-pitch-roll angles: $\boldsymbol{angles_0}$
- Three initial angular velocities: $\boldsymbol{\omega_0}$

The output is the illuminated-projected area variation array ($\boldsymbol{c_{tot}}$) that is fed as a .txt file to TIG (Tracklet Image Generator [4]). TIG is a Python application that takes as input night sky acquisitions and outputs a faithful representation of the object tracklet, according to its dynamics. To this work, some modifications are done to the program for the integration with the other portions of the code, while its main structure is left untouched. In this new version, the inputs to TIG are:

- One or more night sky FITS images
- The .txt file of the area variation
- An additional .txt file containing the Azimuth (Az) and Elevation (El) evolution for a given tracked object

According to this last file, TIG identifies the number of pixels needed to represent the given passage of the object, and it places the streak selecting the pixel coordinates in the image. The tracklet colouring is performed by substituting these central pixel values with

$\boldsymbol{c_{tot}}$ scaled values, according to Eq. (4), with the introduction of some noise. A certain thickness is assigned to the streak and the colouring is performed also for the pixels around the central line, with decreasing intensity. In this new version, TIG can represent saturated or partially saturated streaks. Saturation entails an increase in the smearing of the streak itself due to the bleeding effect. Furthermore, the pixel values must be clipped to the maximum value (saturation threshold, *sat*) according to the input sky image.

Now, either real or synthetic FITS images are fed to Streak Analysis Pipeline (SAP), a novel Python application developed for the extraction and analysis of streaks. SAP, exploiting several Python astronomical packages, performs the following main tasks:

1. Detection and extraction of the streak(s) from the input FITS file
2. Extraction of central line luminosity variation (light curve in terms of light intensity)
3. Evaluation of the streak thickness
4. Source extraction with DAOPHOT python package [TBD], control star (lowest magnitude) identification, and circular aperture photometry
5. Catalogue query and catalogue matching with a cone search within the image FOV
6. Rectangular aperture photometry on the streak; each aperture is one pixel wide and with length equal to the maximum identified streak thickness. The light curve is extracted in terms of apparent magnitude variation in time

Tasks from 4 to 6 are performed in the case of saturated streaks. For each run on an image, SAP produces three HTML diagnostic pages to track how the application worked. Moreover, it outputs:

- Two tables containing useful information about the streak RA and DEC evolution in time and the observer location and time
- A .txt file featuring the real light curve evolution (either central line luminosity or apparent magnitude).

These outputs are used by the last part of the code, i.e. the optimization solver. The GA MATLAB implementation (*ga.m*) fits the extracted real light curve lum_{real} with the simulated one $lum_{simulated}$. Both light curves are scaled with respect to *sat* (saturation threshold to have values between 0 and 1) for a faster optimization; then, an objective function (*obj_fun.m*) sums all the elements of the difference absolute value between the two:

$$\begin{aligned} n_{scaled} &= \frac{n}{sat} \\ lum_{real} &= \frac{lum_{centre}}{sat} \\ lum_{simulated} &= \boldsymbol{c}_{tot} \cdot k + n_{scaled} \\ \boldsymbol{J} &= \sum_{i=1}^{n} |lum_{real}(i) - lum_{simulated}(i)| \end{aligned} \tag{6}$$

In this way, a scalar *cost* measure is obtained, and its value is minimized by GA over the selected design variables, that are:

- Attitude initial conditions: three initial angular velocities (ω_{0x}, ω_{0y}, and ω_{0z}) and three initial yaw-pitch-roll angles (ϕ_0, θ_0, and ψ_0, cf. Sec. 2), so that only six parameters are needed to simulate the dynamics and input in *dynamics.m*, as $\boldsymbol{var_{input}}$.
- Scaling parameters: only multiplicative factor *k* since *n* is taken as the mean pixel value of the image coming from SAP.

The lower and upper boundaries for the design variables range are defined between 0° and 360° for the angles, between 0 and 1 for *k*, and between -ω_{lim} and ω_{lim} for the angular velocities. The value of ω_{lim} is determined by performing a Fourier series expansion of the real light curve up to the second order and evaluating its frequency. For the fitting of saturated streak light curves a different scaling is carried out; no shift is considered, and the simulated light curve is built as $\boldsymbol{mag_{sim}}$. Therefore, the scaling of the real one is performed with respect to $(m_{ref} - I_{ref})$ as follows:

$$\begin{aligned} \boldsymbol{mag}_{real} &= \frac{\boldsymbol{m}}{m_{ref} - I_{ref}} \\ \boldsymbol{mag}_{sim} &= \mathbf{1} - \boldsymbol{c}_{tot} \cdot k \\ \boldsymbol{J} &= \sum_{i=1}^{n} |mag_{real}(i) - mag_{simulated}(i)| \end{aligned} \tag{7}$$

Where m_{ref} is the apparent magnitude and I_{ref} the instrumental magnitude

4. Results

The overall procedure of angular velocity extraction is applied on both synthetic and real images. In the realization of synthetic images, different rotational regimes of space debris are considered, to account for different real case scenarios [5]. Three different regimes are arbitrarily defined:

- Slow Rotation: maximum angular velocity below 10 °/s
- Medium Rotation: maximum angular velocity below 90 °/s
- Fast Rotation: maximum angular velocity below 180 °/s

Moreover, both non-saturated and saturated streaks are analysed, and the overall goodness of the different methods is compared.

The real images are taken from the Pratica di Mare ground station through a 350mm-optical telescope (PdM-MITE, Pratica di Mare - MIlitary TElescope) operated by personnel of Aero-Space System Engineering Group of Flight Test Wing [6].

Concerning the machine on which the tests were performed, it is a PC with an Intel i7-6500U CPU@2.5 GHz with a 16 GB RAM.

4.1 Synthetic Images

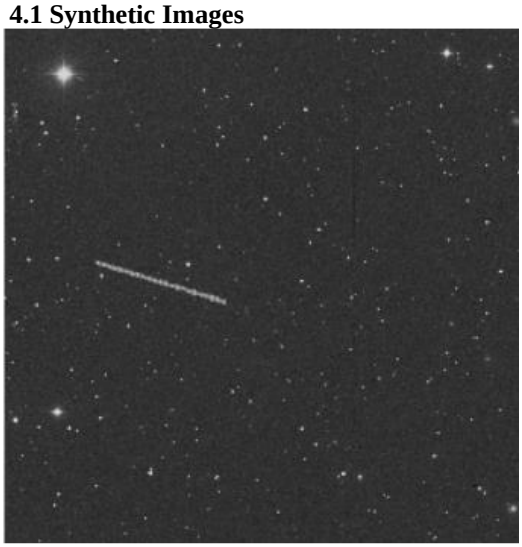


Figure 2: Example of a medium-rotating debris

The main distinction between the generated synthetic images is given by the attitude initial conditions and the multiplicative factor *k*.

For each rotational regime, among the various synthetic images that were generated, a representative one is selected (see Figure Figure 2). The results show the effectiveness of the method and the variability between different runs on the same image. Moreover, the differences between rotational regimes are outlined. In general, a key role is also played by the simulation time since the solver speed is penalized when it increases. For each rotational regime, the initial conditions for the dynamics simulation and the multiplicative factor *k* are summarized in Table 1 (Sim), along with the results obtained through the optimization for a given run (Opt). In addition, the same values for a slow-rotating saturated streak are presented. The results show that:

1. The angular velocity norm ($\|\boldsymbol{\omega_0}\|$) is accurately retrieved in all the rotational regimes. The accuracy decreases with the increasing of the rotations and in general for saturated streaks.
2. The angular velocity z-component is the best fitted one. The in-plane components are sometimes inverted due to the plate symmetry.
3. The attitude angles are usually not properly fitted because of the presence of multiple solutions related to the light curve fitting and of the noise introduced in TIG.
4. The *k* value is optimally fitted in most cases.

Table 1: Synthetic Images – Results: for the slow-rotating regime the results are presented for both non saturated (non sat) and saturated (sat) streaks. For the latter no optimization of k is considered.

	Slow				**Medium**		**Fast**	
	Sim (non sat)	Opt (non sat)	Sim (sat)	Opt (sat)	Sim	Opt	Sim	Opt
ω_{0x} [°/s]	0.1	0.15	5	2.75	25	9.5	81	76.47
ω_{0y} [°/s]	1.5	1.4	2	-2.3	11	25.26	50	57.44
ω_{0z} [°/s]	4	4.03	8	-8.3	50	50.1	120	120.1
$\|\boldsymbol{\omega_0}\|$ [°/s]	4.27	4.26	9.64	9.03	56.9	57.1	153.2	153.6
ϕ_0 [°]	240	106	0	281	0	124	200	191
θ_0 [°]	245	351	80	272	0	43	15	20
ψ_0 [°]	165	341	200	89	0	8	45	42
k [-]	0.153	0.120	-	-	0.229	0.226	0.183	0.182

The results can be characterized in terms of the relative error for the angular velocity norm for each run (*err*):

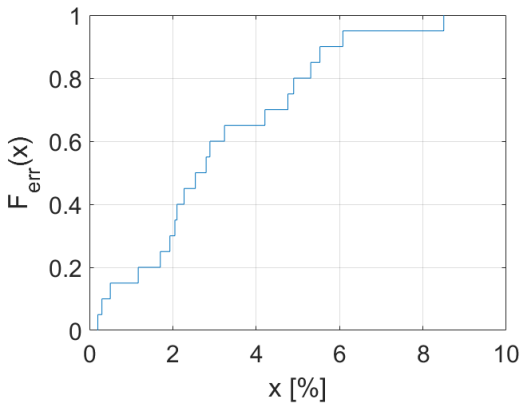
$$err = \left| \frac{\left\|\boldsymbol{\omega}_0^{sim}\right\| - \left\|\boldsymbol{\omega}_0^{opt}\right\|}{\left\|\boldsymbol{\omega}_0^{sim}\right\|} \right| \tag{8}$$

To obtain a statistical relevance of this parameter, the optimization with *ga.m* is carried out multiple times for

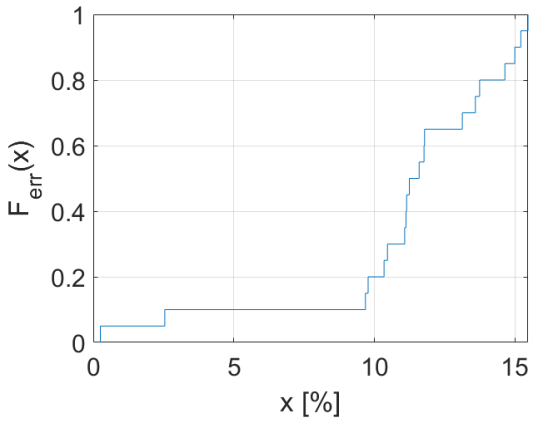
the selected image. Then, the cumulative distribution function (CDF) of *err* values for each run is computed to evaluate the changes between different rotational regime cases.

As it can be seen in Figure 3, the steepness of the CDF decreases with faster rotations; this implies a worsening of the performances since the probability of having higher err values is increased from slow to fast rotational regimes. Furthermore, it can be noted a similar accuracy penalization for saturated streaks in terms of CDF.

(a)



(b)



(c)

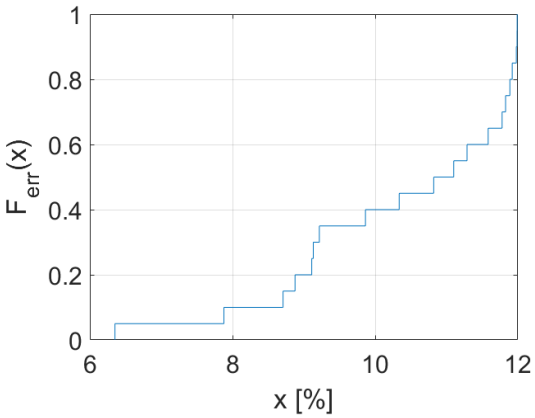


Figure 3: CDF of the relative errors. **(a)** CDF for slow rotation case. **(b)** CDF for fast rotation case. **(c)** CDF for saturated case

4.2 Real Images

In the real images case, the results of the optimization can be directly shown, and the overall process is lighter in terms of tasks to be performed. The analysed images are coming from a series of acquisitions performed at the Pratica di Mare ground station during an observation campaign in September 2017.

As it can be seen in Figure 4, the image acquired by PdM-MITE contains a faint streak left by a debris identified as the COSMO SkyMed 2, an ASI (Agenzia Spaziale Italiana) artificial LEO satellite launched in 2007.

Figure 4: Real image from the PdM-MITE 2017 observation campaign. The rendering of the image is in logarithmic scale to better visualize the faint streak.

The curve extracted by SAP is quite noisy (see Figure 5). Therefore, a filtering of the signal is performed, as well as a re-sampling (considering possible aliasing problems) to reduce the dimensions of the luminosity

array. Then, this filtered-optimized signal is fed to the optimization.

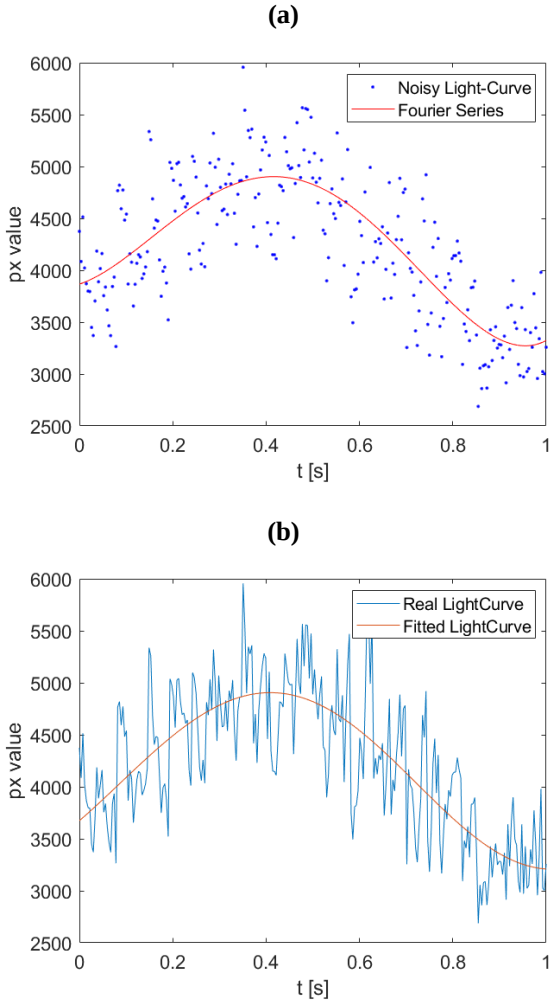


Figure 5: Light curve filtering and results of the fitting. **(a)** Real noisy light curve and filtered light curve through Fourier series expansion. **(b)** Results of the optimization: real light curve in blue, optimized one in red.

Table 2: Real Image - Results

ω_{0x} [°/s]	ω_{0y} [°/s]	ω_{0z} [°/s]	$\|\|\boldsymbol{\omega_0}\|\|$ [°/s]
-11.5	2.8	149	149.47

The results in terms of angular velocity (see Table 2) are rather consistent with a fast-spinning object in LEO and, as seen before for fast rotating debris, the solver performance decreases with the increased value of angular velocity both in terms of prediction accuracy and of optimization speed. Nevertheless, for the identified object, the value of ω_{0z} seems quite high. The short-evolution dynamics of only 1 second does not allow a more precise estimate.

The optimization is performed for 100 times on the same image to evaluate the variability of the results. To achieve so, a new relative error is defined as:

$$err = \left| \frac{\|\boldsymbol{\omega_0}\| - \mu}{\mu} \right| \tag{9}$$

Where μ = 140.01 °/s is the statistical mean of the angular velocities norm. Then, it is possible to evaluate the CDF of *err* and consider the probability of having a $\|\boldsymbol{\omega_0}\|$ that differs more than 1σ (standard deviation, σ = 31.58 °/s) from μ.

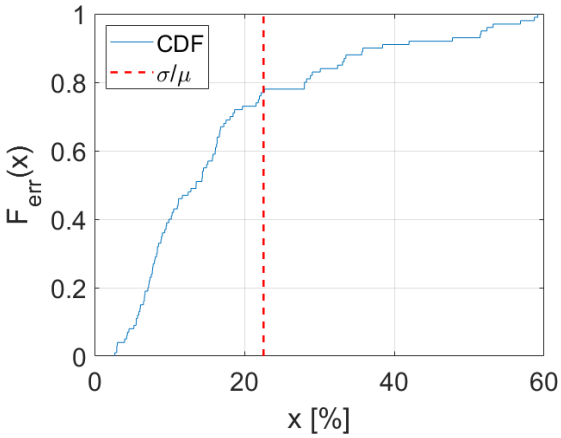


Figure 6: CDF of the relative error for real streaks.

The CDF is quite steep, even if *err* is not really contained. Nevertheless almost 80% of the results are within 1σ of μ.

The results for 100 runs, as it can be seen in Figure 6, are quite variable. This is mainly due to the short evolution of the extracted light curve that may have different optimal fittings and to the high rotational speeds, leading to worse results.

6. Conclusions

The attitude determination of a tumbling space debris is quite a difficult task to achieve. In this work, a light curve inversion for attitude reconstruction is carried out on both real and synthetic images. The generation of synthetic images performed using TIG and the dynamics simulator, outputs FITS files that resemble the characteristics of a real streak left by a tumbling debris. Moreover, SAP successfully extracts, analyses, and post-processes the streak present in any input FITS image. The optimization part of the code can identify different rotational regimes of the tumbling object and conduct the analysis accordingly. The performance of the optimization generally decreases with the increase of the rotation rate (fast rotating debris) and if the streak is saturated. The analysis proved to be quite successful in the determination of the norm of the angular velocity vector and, in some cases, of its components. The results

for real images containing tumbling space debris are quite variable and their accuracy should be further validated. More complex models could be used for the reflection and shaping of the debris in case a more refined analysis is required.

Acknowledgements

First of all, I wish to show my deepest gratitude to Professor Mauro Massari, who followed me patiently during the development of this work. I would also like to thank the Italian Air Force for the precious images from the Pratica di Mare ground station used in the work. Moreover, I would like to pay my special regards to Andrea De Vittori, Riccardo Cipollone and Giovanni Purpura, who followed and helped me in the writing of the code, and for their precious advice and their extreme availability.

References

[1] ESA Space Debris Office. ESA's Annual Space Environment Report, 27 May 2021. https://www.sdo.esoc.esa.int/environment_report/Space_Environment_ Report_latest.pdf.

[2] F. Piergentili, G. Zarcone, L. Parisi, L. Mariani, S. H. Hossein, and F. Santoni. LEO Object's Light-Curve Acquisition System and Their Inversion for Attitude Reconstruction. Aerospace, 8(1), 2021.

[3] W. Romanishin. An Introduction to Astronomical Photometry Using CCDs. https://www1.phys.vt.edu/~jhs/phys3154/CCDPhotometryBook.pdf, March 2002.

[4] R. Cipollone and A. De Vittori. Machine Learning Techniques for Optical and Multi- beam Radar Track Reconstruction of LEO Objects. Master's thesis, Politecinco di Milano, School of Industrial and Information Engineering, Department of Aerospace Science and Technology, Milan, Italy, 2020.

[5] J. Šilha, J-N. Pittet, M. Hamara, and T. Schildknecht. Apparent rotation properties of space debris extracted from photometric measurements. Advances in Space Research, 61(3):844–861, 2018.

[6] G. M. D. Genio, J. Paoli, E. D. Grande, F. Dolce. Italian Air Force Radar and Optical Sensor Experiments for the Detection of Space Objects in LEO Orbit.