SAPIENZA
UNIVERSITÀ DI ROMA

Dipartimento di Ingegneria Astronautica
Elettrica ed Energetica

Dottorato in Ingegneria Aerospaziale

Ciclo XXVI

A Search Algorithm for Stochastic Optimization in Initial Orbit Determination

Candidato
Luigi Ansalone

Relatore
Prof. Fabio Curti

Anno Accademico 2012/2013

Alla mia famiglia, a Romina

Questa tesi è stata composta con LaTeX.
Visit GuIT & Happy TeXing!

Contents

Abstract

Optical observations constitute a source of angular measurements of a satellite pass. Commonly, these observations have short durations with respect to the satellite orbit period. As a consequence, the use of classical orbit determination algorithms, as Laplace, Gauss or Escobal methods, give very poor results. This thesis faces with the problem of estimating the orbital parameters of an orbiting object using its optical streak acquired by a telescope or a high accuracy camera. In this thesis a new technique is developed for the Initial Orbit Determination from optical data by exploiting the genetic algorithms. The algorithm works without restrictions on the observer location. A recent challenging problem is the Initial Orbit Determination with space-based observations. This thesis focuses on the problem of determinating the orbital parameters of a satellite from an orbiting observer in LEO, using short time observations. We present the results based on a simulation with the observer on a sun-synchronous orbit with a single observation of just 60 s. Monte Carlo simulations are presented with different levels of sensor accuracy to show the reliability of the algorithm. The algorithm is able to yield a candidate solution for each observation. The coplanar case is analyzed and discussed as well. Several test show the reliability of the algorithm varying the number of the observations, the initialization method, the observation period and the noise seed.

Abstract in Italiano

Le osservazioni ottiche costituiscono una possibilità per misure angolari di un passaggio di un satellite. Tipicamente le osservazioni tramite camera, o telescopio, hanno una durata molto limitata, essendo il tratto dellorbita osservabile solo una piccola frazione dell'orbita completa. I classici metodi di determinazione iniziale dell'orbita non forniscono risultati affidabili se l'arco di osservazione è troppo piccolo. I metodi di Laplace, Gauss o Escobal non riescono a fornire una stima dell'orbita con un passaggio molto breve e tipicamente si utilizzano con più archi di osservazione. Questa tesi è rivolta allo studio di un algoritmo numerico per la risoluzione di archi piccoli di osservazione. L'approccio è improntato sugli algoritmi di ricerca per spazi continui delle soluzioni. Una review iniziale degli algoritmi di ricerca è servita per identificare la strategia migliore per ottenere una stima dei parametri orbitali. L'algoritmo di ricerca scelto è un algoritmo genetico, che fornisce ottime prestazioni in spazi continui e buone garanzie che la soluzione ottima venga raggiunta. Come per ogni altro algoritmo di tipo euristico, i risultati dell'algoritmo genetico dipendono molto dallinizializzazione. Diverse tipologie di inizializzazione sono state studiate, identificando un metodo efficace e computazionalmente poco oneroso. L'algoritmo funziona senza restrizioni sulla posizione dell'osservatore; quindi è molto utile pensare di testarlo con osservazioni eseguite da un osservatore anche esso in orbita. L'osservazione di satelliti e detriti da un telescopio spaziale è un tema molto attuale che ha la necessità di confrontarsi con lo studio e il trattamento di archi piccoli. La tesi presenta due casi distinti, con osservatore in orbita e osservatore sulla Terra; per il secondo caso osservazioni reali sono state utilizzate per il caso di satelliti geostazionari colocati. L'algoritmo è stato testato variando diversi parametri per testare l'affidabilità e la convergenza dei risultati. Una particolare attenzione è stata anche rivolta alle proprietà dei generatori di numeri pseudo-casuali, che sempre ricorrono negli algoritmi euristici. Le analisi sono state verificate anche con metodi Monte Carlo, dimostrando che l'algoritmo è sempre in grado di fornire una soluzione che riesca a riprodurre le misure ottenute. Il problema della complanarità tra osservatore e target è *stato trattato, rilevando che non ci sono problemi numerici e che una soluzione fattibile è sempre raggiungibile. Per il test dell'algoritmo con immagini vere da telescopio, un modello molto accurato della rotazione terrestre è preso in considerazione per ruotare la posizione dell'osservatore in un sistema di riferimento inerziale. Un particolare sforzo è stato dedicato alla calibrazione astrometrica delle immagini, che hanno necessitato di un catalogo stellare molto fitto ed accurato. L'estrazione delle informazioni angolari raggiunge accuratezze dell'ordine del decimo di arcosecondo, con una definizione subpixel.*

Acknowledgement

I would like to express the deepest appreciation to my advisor Professor Fabio Curti, who has shown the attitude and the substance of a deep Mentor: he continually and persuasively conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to teaching. Without his supervision and constant help this dissertation would not have been possible. I would like to thank Professor Maurizio Parisse for his continuous and smart advices, I will not forget some very beautiful discussions with him. A special thank to Professor Emilio Frazzoli and Professor Sertac Karaman that allowed me to live an extraordinary experience of a research period at MIT. I would like to thank the friends and colleagues of this PhD adventure: Alessandro and Stefano. I would like to thank the students that have founded with me the ARCA Lab: Daniele, Simone and Luca, probably without them I would not have been fallen in love with space robotics and automation. I would like to thank all the students and friends I have met here, the list is probably too long, but I want to thank especially Cesare, Eleonora, Michela and Claudia for having suffered my cotutoring during their thesis; it has been a very formative experience. A new thank to the engineers I have been working with in this period: Dario and Martina; it has been a pleasure working with you. I want to thank the ATA research team, and especially Angelo Tomassini, for having provided the images I have used to test the algorithms. I want, of course, to thank my family and Romina for the patience and for being always the pillars of my life.

List of Figures

List of Tables

List of Acronyms

AAVSO American Association of Variable Star Observers

ACES Atomic Clock Ensemble in Space

ACO Ant Colony Optimization

ACT Advanced Concepts Team

APASS AAVSO Photometric All-Sky Survey

ASIM Atmosphere-Space Interactions Monitor

ATA Associazione Tuscolana di Astronomia

BBS Blum Blum Shub

CCD Charge Coupled Device

CEPF Columbus-External Payload Facility

Dec Declination

EAE Equivalent Angular Error

ECEF Earth-Centered Earth-Fixed

ESA European Space Agency

EuTEF European Technology Exposure Facility

FOV Field Of View

GEO Geostationary Earth Orbit

GFSR Generalized Feedback Shift Register

GPS Global Positioning System

IADC Inter-Agency Space Debris Coordination Committee

ICRF International Celestial Reference Frame

ICRS International Celestial Reference System

IERS International Earth Rotation Service

ITRF International Terrestrial Reference Frame

IOD Initial Orbit Determination

ISS International Space Station

LCG Linear Congruential Generator

LEO Low Earth Orbit

LFG Lagging Fibonacci Generator

LOS Line Of Sight

MC Monte Carlo

MCG Multiplicative Congruential Generator

MEME Mean Equator Mean Equinox

MFLG Multiplicative LFG

MPC Minor Planet Center

MSL Mean Sea Level

MTA Mersenne Twister Algorithm

NEO Near Earth Object

NFL No Free Lunch

NOMAD Naval Observatory Merged Astrometric Dataset

NORAD North American Aerospace Defense Command

NP Nondeterministic Polynomial time

PRNG Pseudo Random Noise Generator

PSF Point Spread Function

PSO Particle Swarm Optimization

RA Right Ascension

RAAN Right Ascension of the Ascending Node

RAM Random Access Memory

RD Research and Development

RMS Root Mean Square

RNG Random Number Generator

SBIG Santa Barbara Instrument Group

SDP4 Simplified Deep-space Perturbations 4

SGP4 Simplified General Perturbations 4

SNR Signal Noise Ratio

SSA Space Situational Awareness

TEME True Equator Mean Equinox

TLE Two Lines Elements

TOF Time Of Flight

TSA Too Short Arc

TSP Travelling Salesman Problem

UAI Unione Astrofili Italiani

UCAC USNO CCD Astrograph Catalog

USNO United States Naval Observatory

UTC Coordinated Universal Time

VSA Very Short Arc

WGS84 World Geodetic System 1984

List of Symbols

Symbol	**Units**	**Description**
$\hat{L}$	-	Angular measurement
ρ	km	Distance between observer and target
fit	-	Fitness function output
μ	$\frac{km^3}{s^2}$	Standard gravitational parameter
a	km	Semimajor axis
e	-	Eccentricity
i	$^\circ$	Inclination
Ω	$^\circ$	Right ascension of the ascending node
ω	$^\circ$	Argument of perigee
ν	$^\circ$	True anomaly
u	$^\circ$	Argument of latitude
D	km	Maximum distance for the genetic algorithm initialization
r	-	Correlation coefficient
γ	$^\circ$	Angle between actual and reconstructed measurement

Chapter 1

Introduction to Orbit Determination

The enchanting charms of this sublime science reveal themselves in all their beauty only to those who have the courage to go deeply into it.

- Carl Friedrich Gauss, *Letter to Sophie Germain, 1807*

In this chapter the basics of the orbit determination are given in order to introduce the main characteristics of the problem of finding the orbit of an observed object. Different methods for angular observations will be analyzed to introduce the problem of too-short arc orbit determination.

1.1 Orbit determination concepts

The orbit determination problem deals with the problem of finding an estimation of the state vector of an observed satellite in orbit around a massive object. The classical orbit determination problem is based on the hypothesis that the two bodies are interacting only with attractive forces, no other forces are kept into account. This problem is better known as the two-body problem. The bodies are needed to be considered as having point mass, moreover the satellite has a negligible mass or, in other terms, it does not effect the dynamics of the massive object, so only the motion of the satellite will be considered.

The orbit determination can be described as the problem of finding the set of independent parameters needed to give a solution of the satellite dynamics; under the assumptions above, this case can be resolved with six independent parameters. For the satellite these parameters could be, for example, position and velocity at a certain epoch. This minimal set can be propagated through time to obtain the future state of the satellite.

The real motion of the satellite include a set of complex dynamics equations that makes the error of orbit propagation increase with the propagation time. The equations needed to describe the actual motion are not only complex but also known at a given accuracy; parameters uncertainties, or local variations as well, will cause the deviation of the predicted motion from the actual motion.

The deviation of the error of the orbit propagation makes the observations indispensable to update the estimation of the satellite orbit. The problem of orbit estimation is given for the satellite whose initial state is not known a priori, but observations can help us to obtain the set of six parameters. The observations are corrupted by sistematic and random errors, so a solution that minimizes a certain index is needed to obtain a good estimation; furthermore, usually the state vector is not directly observable but the observations are the results of non linear functions of the state vector.

1.1.1 Historical background

The problem of orbit determination always has fascinated scientists since the beginning of the celestial mechanics, see [79]. Kepler in 1607 stated that the orbit of every planet is an ellipse with the sun at a focus, so all the successive methods are based on finding the parameters of this kind of orbit, that in general is a conic section. The first method was devised from Newton and used by many astronomers; between them there was also Sir Edmond Halley. Halley in 1705 used the Newton method to calculate the orbits of all comets that had enough observations. He is famous especially for the calculation of the passage of the Halley comet, obviously named in this way to celebrate his discovery. He noted that the comets observed in 1607 and 1682 had the same orbit characteristics, so he thought that these observations could be lead to the same object, predicting its return in 1758. He was right but did not lived enough to check the prediction.

The Newton method is based on a graphical construction with successive approximations. Euler in 1744 was the first to solve analitically the problem for a parabolic orbit. Lambert in 1771 gave a generalization of the Euler technique for elliptic and hyperbolic orbits. In 1780 Laplace published a new method, followed by the method of Gauss, issued in 1800, that led to the rediscovery of the asteroid Ceres in 1801 after its lost. Gauss is also known for the use of a least squares method for the orbit estimation problem that he solved in 1809, while the first clear exposition of the least squares method was given by Laplace in 1805.

1.2 The Kepler orbit

The Kepler orbit is the orbit that a satellite is subject to when these assumptions are taken into account:

- Mass of the satellite negligible
- The motion is described in an inertial frame
- The involved bodies are considered each as a point mass
- No other forces are acting, except the gravitational ones

With these assumptions it is easy to obtain that the orbit is a conic section and it is confined in a plane:

$$\vec{h} = \vec{r} \times \vec{v} = const \tag{1.1}$$

$\vec{h}$ is called the orbital angular momentum, that is the cross product between position $\vec{r}$ and velocity $\vec{v}$. It is constant so position and velocity at a certain time define the orbital plane. The orbital energy is constant:

$$E = \frac{v^2}{2} - \frac{\mu}{r} = const = -\frac{\mu}{2a} \tag{1.2}$$

Where E is the total energy and a is knows as semimajor axis.

The shape of the orbit is defined by two quantities: a and the eccentricity e. e is defined as the modulus of the eccentricity vector:

$$\vec{e} = \frac{\vec{v} \times \vec{h}}{\mu} - \frac{\vec{r}}{r} = const \tag{1.3}$$

The eccentricity can be obtained in many ways, one easy way is in Eq. (1.3).

$$e = \frac{r_a - r_p}{r_a + r_p} \tag{1.4}$$

where r_a and r_p are respectively the distance from the focus at the apoapsis and the periapsis, these are called apogee and perigee in the case of Earth orbit, see Fig. 1.1.

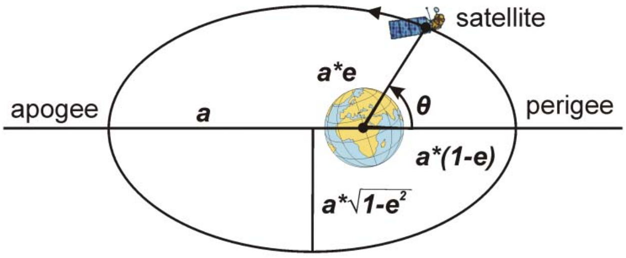


Figure 1.1: The semimajor axis

The orbital inclination is defined as:

$$cos(i) = \frac{\hat{K} \cdot \vec{h}}{|\vec{h}|} \tag{1.5}$$

Where $\hat{K}$ is the third unit vector of the inertial reference frame $\hat{I}\hat{J}\hat{K}$. $\hat{K}$ represents the Earth's rotation axis and $\hat{I}$ the vernal equinox direction. It is very useful to define also the nodal axis, or *Line of nodes*, as:

$$\vec{n} = \hat{K} \times \vec{h} \tag{1.6}$$

The nodal axis represents the intersection between the orbital plane and the equatorial plane, see Fig. 1.2. The Right Ascension of the Ascending Node (RAAN) is defined through the scalar product between $\hat{I}$ and $\vec{n}$:

$$\begin{gathered} cos(\Omega) = \frac{\hat{I} \cdot \vec{n}}{|\vec{n}|} \\ if(n_j < 0)\ then\ \Omega = 360° - \Omega \end{gathered} \tag{1.7}$$

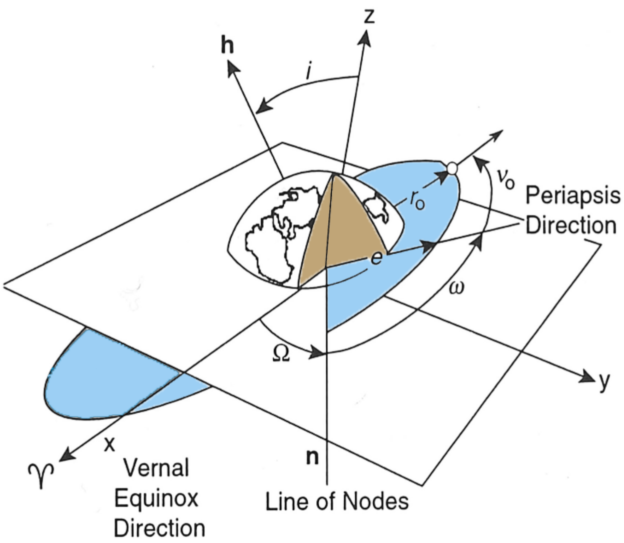


Figure 1.2: The Line of nodes

The orbital plane is so defined by these two values: i and Ω. The argument of perigee ω is the angle between the nodal axis and the periapsis, that is the angle between the nodal axis and the eccentricity vector, see Eq. (1.8).

$$\begin{gathered} cos(\omega) = \frac{\vec{n} \cdot \vec{e}}{|\vec{n}||\vec{e}|} \\ if(e_k < 0)\ then\ \omega = 360° - \omega \end{gathered} \tag{1.8}$$

The last orbital element is the true anomaly, that defines the position of the spacecraft respect to the periapsis, it is so the angle between the periapsis and the current position r:

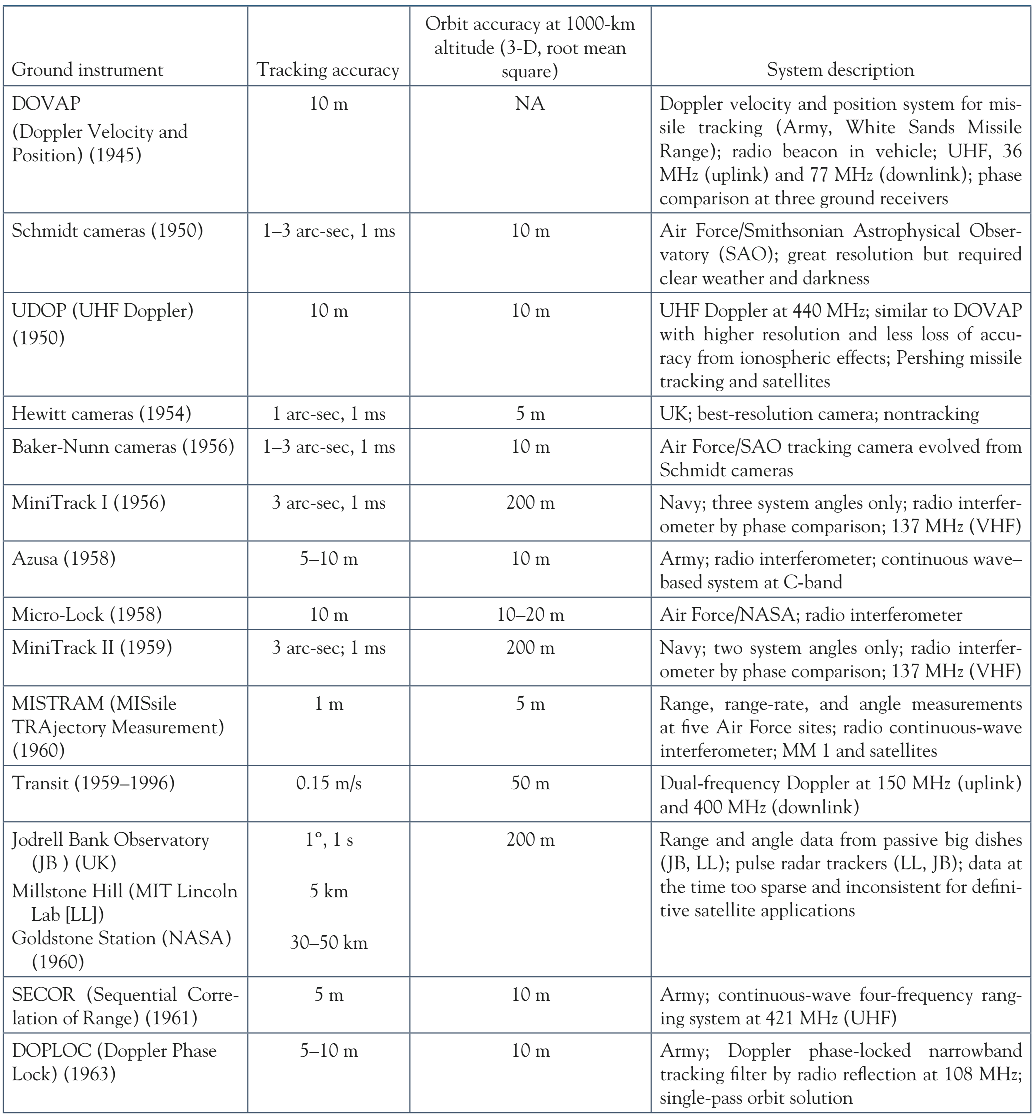
$$\begin{gathered} cos(\nu) = \frac{\vec{e} \cdot \vec{r}}{|\vec{e}||\vec{r}|} \\ if(\vec{r} \cdot \vec{v} < 0)\ then\ \nu = 360° - \nu \end{gathered} \tag{1.9}$$

The set of these six orbital parameters completely defines the orbit and the position of the satellite:

$$[a, e, i, \Omega, \omega, \nu] \tag{1.10}$$

1.3 The measurements

The discoveries obtained in the historical section 1.1.1 are based on angular measurements thanks to the telescope. Other sources of observations simply did not exist. In Fig. 1.3 the first methods for angular and range data are summarized [82], the first range methods were focusing on missile tracking, since the first satellite was launched in 1957.



Ground instrument	Tracking accuracy	Orbit accuracy at 1000-km altitude (3-D, root mean square)	System description
DOVAP (Doppler Velocity and Position) (1945)	10 m	NA	Doppler velocity and position system for missile tracking (Army, White Sands Missile Range); radio beacon in vehicle; UHF, 36 MHz (uplink) and 77 MHz (downlink); phase comparison at three ground receivers
Schmidt cameras (1950)	1–3 arc-sec, 1 ms	10 m	Air Force/Smithsonian Astrophysical Observatory (SAO); great resolution but required clear weather and darkness
UDOP (UHF Doppler) (1950)	10 m	10 m	UHF Doppler at 440 MHz; similar to DOVAP with higher resolution and less loss of accuracy from ionospheric effects; Pershing missile tracking and satellites
Hewitt cameras (1954)	1 arc-sec, 1 ms	5 m	UK; best-resolution camera; nontracking
Baker-Nunn cameras (1956)	1–3 arc-sec, 1 ms	10 m	Air Force/SAO tracking camera evolved from Schmidt cameras
MiniTrack I (1956)	3 arc-sec, 1 ms	200 m	Navy; three system angles only; radio interferometer by phase comparison; 137 MHz (VHF)
Azusa (1958)	5–10 m	10 m	Army; radio interferometer; continuous wave–based system at C-band
Micro-Lock (1958)	10 m	10–20 m	Air Force/NASA; radio interferometer
MiniTrack II (1959)	3 arc-sec; 1 ms	200 m	Navy; two system angles only; radio interferometer by phase comparison; 137 MHz (VHF)
MISTRAM (MISsile TRAjectory Measurement) (1960)	1 m	5 m	Range, range-rate, and angle measurements at five Air Force sites; radio continuous-wave interferometer; MM 1 and satellites
Transit (1959–1996)	0.15 m/s	50 m	Dual-frequency Doppler at 150 MHz (uplink) and 400 MHz (downlink)
Jodrell Bank Observatory (JB) (UK) Millstone Hill (MIT Lincoln Lab [LL]) Goldstone Station (NASA) (1960)	1°, 1 s 5 km 30–50 km	200 m	Range and angle data from passive big dishes (JB, LL); pulse radar trackers (LL, JB); data at the time too sparse and inconsistent for definitive satellite applications
SECOR (Sequential Correlation of Range) (1961)	5 m	10 m	Army; continuous-wave four-frequency ranging system at 421 MHz (UHF)
DOPLOC (Doppler Phase Lock) (1963)	5–10 m	10 m	Army; Doppler phase-locked narrowband tracking filter by radio reflection at 108 MHz; single-pass orbit solution

Figure 1.3: The first ground instruments

1.3.1 The angular measurements

The angular measurements are essentially taken from optical sensors that image the sky while the satellite is passing. When a satellite streaks over the picture, the position of the initial and the final points of the streak are compared to the positions of the known stars. One of the first cameras used for this approach was the Baker-Nunn camera shown in Fig. 1.4.

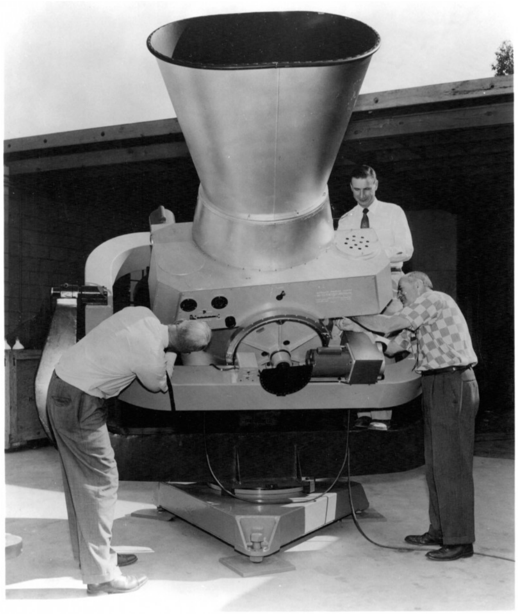


Figure 1.4: The first Baker-Nunn camera

The angular directions are so determined but without information on range. We need to determine the six independent parameters to calculate the orbit, so at least three Line Of Sight (LOS) vectors are needed, since each angular measurement is composed by two values: azimuth and elevation in topocentric coordinates.

The common Initial Orbit Determination (IOD) systems work in a geocentric inertial reference frame, the International Celestial Reference Frame (ICRF) for example; while the measurements are taken in a site with known longitude, latitude and height over sea level in a Earth-Centered Earth-Fixed (ECEF), the International Terrestrial Reference Frame (ITRF) for example. The observer site is needed to roto-translate to an inertial reference frame. Accurate roto-translations are very important to get useful observations, including precession and nutation calculations.

Each angular measurement $\hat{L}_i$ can be expressed in ICRF as:

$$\hat{L}_i = \begin{bmatrix} cos(\alpha_i)cos(\delta_i) \\ sin(\alpha_i)cos(\delta_i) \\ sin(\delta_i) \end{bmatrix} \tag{1.11}$$

Where α_i and δ_i are respectively Right Ascension (RA) and Declination (Dec) of the observation at time t_i.

For a generic instant the position of the satellite can be expressed as a function of the observer position and the angular measurement:

$$\vec{r} = \rho\hat{L} + \vec{r}_{site} \tag{1.12}$$

Eq. (1.12) refers to vectors in ICRF, it is clear that the only missing part is the range ρ, that is the scalar representing the distance between observer and target. In Fig. 1.5 three angular observations are shown, highlighting that the ranges ρ_i are needed to complete the computation of the satellite position.

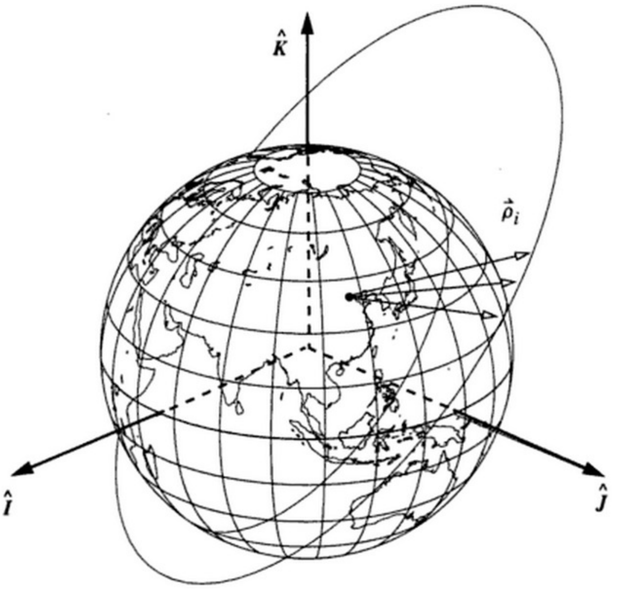
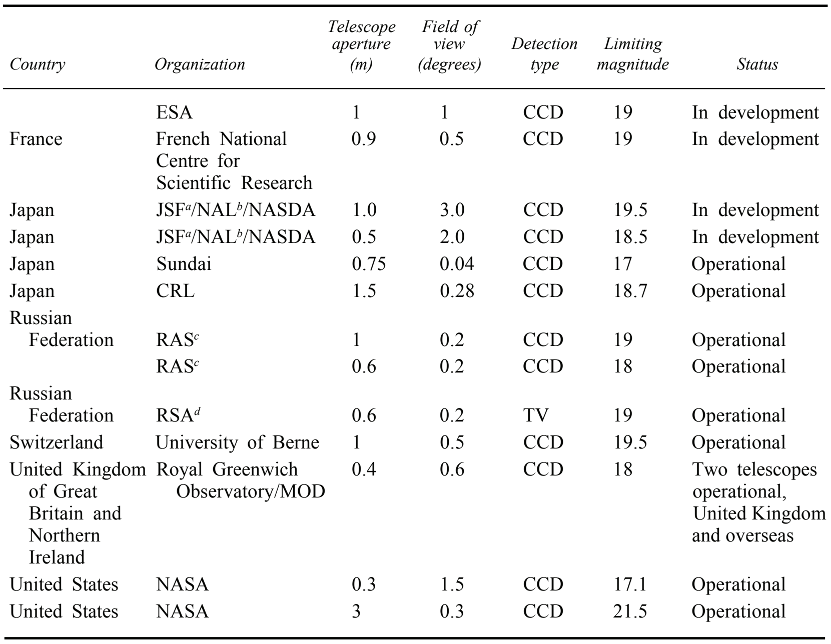


Figure 1.5: The observer-satellite distance

The angular measurements are more easy and cheap to get and are very suitable for very distant satellites, like the Geostationary Earth Orbit (GEO) satellites. Several optical facilities are used in many countries [75]. Optical observations are very useful over 5000 km, see Fig. 1.7 [39].

The European Space Agency (ESA) has a telescope located in Tenerife, Spain, where a Zeiss 1-metre telescope is used for the survey and characterisation of objects near the geostationary ring. The telescope is equipped with Ritchey-Chrtien optics, with a Field Of View (FOV) of about 0.7° and highly efficient CCD Cameras.



Country	Organization	Telescope aperture (m)	Field of view (degrees)	Detection type	Limiting magnitude	Status
	ESA	1	1	CCD	19	In development
France	French National Centre for Scientific Research	0.9	0.5	CCD	19	In development
Japan	JSF[a]/NAL[b]/NASDA	1.0	3.0	CCD	19.5	In development
Japan	JSF[a]/NAL[b]/NASDA	0.5	2.0	CCD	18.5	In development
Japan	Sundai	0.75	0.04	CCD	17	Operational
Japan	CRL	1.5	0.28	CCD	18.7	Operational
Russian Federation	RAS[c]	1	0.2	CCD	19	Operational
	RAS[c]	0.6	0.2	CCD	18	Operational
Russian Federation	RSA[d]	0.6	0.2	TV	19	Operational
Switzerland	University of Berne	1	0.5	CCD	19.5	Operational
United Kingdom of Great Britain and Northern Ireland	Royal Greenwich Observatory/MOD	0.4	0.6	CCD	18	Two telescopes operational, United Kingdom and overseas
United States	NASA	0.3	1.5	CCD	17.1	Operational
United States	NASA	3	0.3	CCD	21.5	Operational

Figure 1.6: Optical facilities for satellite observation

1.3.2 The range and range-rate measurements

All measurements of range are based on the propagation of a signal in a medium; if the signal velocity and the Time Of Flight (TOF) are known, it is possible to obtain the range information. The signal velocity is the speed of light, so it is a parameter, while the TOF is needed, here arises the problem of signal timing. The range-rate measurements are instead based on the Doppler effect.

A really simple subdivision of these measurements is treated in the following list:

- One-way range

  If c is the speed of light, t_r the time at the receiver and t_t the time at the transmitter the range ρ can be expressed as:

  $$\rho = c(t_r - t_t) \tag{1.13}$$

  For example the $(t_r - t_t)$ term is about 87 ms for a satellite orbiting at zenith at 20000 km, as the altitude of Global Positioning System (GPS) satellites. The different times are recorded by different instruments; if they are perfectly synchronized and the signal travels with constant speed, the found range is the true range.

The signal speed is unfortunately not constant if one of the instruments is on the ground, because the atmospheric delays have to be considered. With a very simple model for the clock, the measured range can be written:

$$\rho = \rho_{true} + c(a_r - a_t) + \epsilon \tag{1.14}$$

Where a_r and a_t are the constant offsets of time, ϵ is a stochastic error component. Clock drift is not considered, as other terms like nonlinear terms.

- Two-way range

  This method includes a two-way path, an uplink and a downlink path. For sake of simplicity it is assumed that the transmitter is on the ground and no retransmission delay is considered. In this case the measured range is:

  $$\rho = \rho_{true} + \epsilon \tag{1.15}$$

  The same offset has been added and subtracted, leading to the simple expression in Eq. (1.15).

  The removal of this source of error is a big advantage for this kind of measurement. Even if the clock drift is considered, the very short time of signal travel can lead to ignore the term related to the error generated by the drift.

- Range-rate

  The most important range-rate system is based on the Doppler effect. If there is a relative motion between transmitter and receiver, the received frequency will be different from the sent frequency. In a very simple model the Doppler shift can be written:

  $$f_r = f_t \left(\frac{1 + v/c}{1 - v/c} \right) \tag{1.16}$$

  Where f_r and f_t are the received and transmitted frequency, v is the relative velocity and c the speed of light. Assuming that the speed of light is much greater than the relative velocity, it possible to assume that:

$$f_D = f_r - f_t \simeq \frac{2vf_t}{c} \tag{1.17}$$

Note that the frequency shift f_D can be measured only in a certain time interval, usually a counter for number of cycles is used.

1.3.3 Radar versus Telescope observations

Radars are used to get range and range-rate information about satellites, they usually require very high power and are big and complex. On the other hand a simple camera can get useful data for the orbit determination based on angular measurements.

The limited power that radars can use limite the maximum distance that they can get, while optical telescope being passive can get to very high distances, see Fig. 1.7.

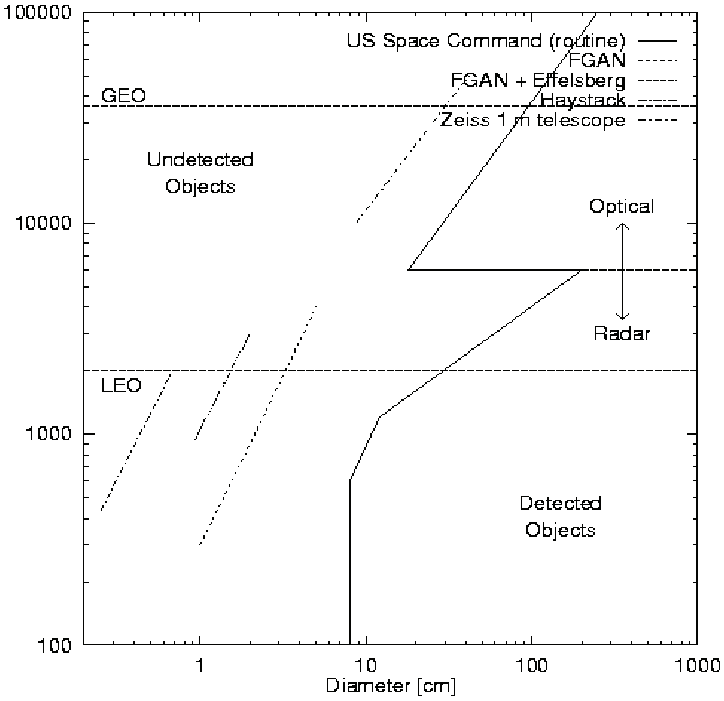


Figure 1.7: Radar versus optical observations

1.4 The algorithms for the angular measurements

A telescope produces images with stars and satellite streaks, comparing the known positions of the stars it is possible to obtain angular measurements of the satellite pass. Several algorithms can be applied, the most important are: Laplace's method, Gauss's method and Escobal's method; these methods will be briefly discussed in the following sections, for a complete overview see [79].

1.4.1 Laplace's method

This method requires three angular observations, Laplace's method estimates the middle position and velocity vectors in a span of observations data. Differentiating Eq. (1.12), Eq. (1.18) is obtained.

$$\dot{\vec{r}} = \dot{\rho}\hat{L} + \rho\dot{\hat{L}} + \dot{\vec{r}}_{site} \tag{1.18}$$

Differentiating twice:

$$\ddot{\vec{r}} = \ddot{\rho}\hat{L} + 2\dot{\rho}\dot{\hat{L}} + \rho\ddot{\hat{L}} + \ddot{\vec{r}}_{site} \tag{1.19}$$

Considering a kepler motion for the satellite, the expression for r is:

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} \tag{1.20}$$

Substituting Eq. (1.20) in Eq. (1.19) and rearraging with Eq. (1.12) leads to:

$$\ddot{\rho}\hat{L} + 2\dot{\rho}\dot{\hat{L}} + \rho\left(\ddot{\hat{L}} + \frac{\mu}{r^3}\hat{L}\right) = -\ddot{\vec{r}}_{site} - \frac{\mu}{r^3}\vec{r}_{site} \tag{1.21}$$

Note that these equations require values like $\dot{\hat{L}}$ and $\ddot{\hat{L}}$ that can be computed using the Lagrange's interpolation formula, for example there is the formula for $\dot{\hat{L}}$:

$$\dot{\hat{L}}(t) = \frac{2t - t_2 - t_3}{(t_1 - t_2)(t_1 - t_3)}\hat{L}_1 + \frac{2t - t_1 - t_3}{(t_2 - t_1)(t_2 - t_3)}\hat{L}_2 + \frac{2t - t_1 - t_2}{(t_3 - t_2)(t_3 - t_2)}\hat{L}_3 \tag{1.22}$$

Considering the middle of the interval being zero ($t = t_2 = 0$) and with equal time span: ($t_2 - t_1 = t_3 - t_2 = \Delta t$), Eq. (1.23) is easily rearranged in:

$$\dot{\hat{L}}(t) = -\frac{t_3}{2\Delta t^2}\hat{L}_1 + \frac{t_1 + t_3}{\Delta t^2}\hat{L}_2 - \frac{t_1}{2\Delta t^2}\hat{L}_3 \tag{1.23}$$

Then the Eq. (1.21) is solved for ρ using the Cramer's rule.

1.4.2 Gauss's method

The Gauss's method works best for interplanetary studies, but it is very useful also for separation angles in the order of 10° or less. Gauss's method assumes that the three position vectors at the times of observations are in the same plane, so:

$$c_1\vec{r}_1 + c_2\vec{r}_2 + c_3\vec{r}_3 = \vec{0} \tag{1.24}$$

Assuming that c_2 is not zero:

$$\vec{r}_1 \times \vec{r}_3(c_1) = \vec{r}_2 \times \vec{r}_3(-c_2) \tag{1.25}$$

$$\vec{r}_1 \times \vec{r}_3(c_3) = \vec{r}_1 \times \vec{r}_2(-c_2) \tag{1.26}$$

Now we can use the f and g functions that can express $\vec{r}_1$ and $\vec{r}_3$ as functions of position and velocity in the middle time:

$$\vec{r}_i = f_i\vec{r}_2 + g_i\vec{v}_2 \tag{1.27}$$

Substituting Eq. (1.12) in Eq. (1.24) gives us:

$$\left[\hat{L}_1|\hat{L}_2|\hat{L}_3\right]\begin{bmatrix} c_1\rho_1 \\ c_2\rho_2 \\ c_3\rho_3 \end{bmatrix} = \left[\vec{r}_{site_1}|\vec{r}_{site_2}|\vec{r}_{site_3}\right]\begin{bmatrix} -c_1 \\ -c_2 \\ -c_3 \end{bmatrix} \tag{1.28}$$

Inverting the matrix constitued by the angular measurements and through an iterative approach it is possible to find the ranges ρ_i and so the position of the satellite $\vec{r}_i$ can be easily obtained.

1.4.3 Escobal's method

The Escobal's method seems to be more efficient for observations that are far apart, even days apart. This method needs to make an initial guess of the solution, then an iterative process stars updating the solution and finally a type of differential correction determines the final solution.

This method will not be treated in detail since this thesis is focused on methods that are reliable for short arc of observations.

1.5 Short arc orbit determination

Short arc observations are very common, since many satellites are visible from a single station for just some minutes. Visibility depends on time, sky conditions, satellite illumination. Typical passes for Low Earth Orbit (LEO) satellites last some minutes, while for a GEO satellite the pass can last the all night if the satellites does not enter in eclipse, that is about 1/3 of the orbit. A Short Arc is a set of N observations that allows the orbit determination of the detected object with classical methods, but usually with relevant errors.

There is a possible dinstinction between shorter arcs of observations:

- Very Short Arc

  A Very Short Arc (VSA) is a set of N observations that does not allow the computation of the orbit through classical methods. When starting from a VSA, either Gauss' method fails, or the differential correction procedure does not converge.One of the method most used is described in [57]. VSA is recorded as a set of N observations, which means that a set of points on a straight line is what is actually detected,with deviations from alignment compatible with the random observational error.

  Thus from the VSA we can compute the straight line, either by linear regression or by other fitting procedure. Then a VSA is represented by an attributable, consisting of a reference time (just the mean of the observing times), two average angular coordinates and two corresponding angular rates for the reference time, see also [76].

- Too Short Arc

  When, as in most cases, the information contained in a VSA is not enough to compute a full, six parameters set of orbital elements, we refer to them as Too Short Arc (TSA). In such a case the problem of orbit determination must begin with the task of linkage, that is identifcation of two TSA belonging to the same physical object.

  Such a bi-identifcation is enough to allow an orbit determination, although it will be, in most cases, of very poor accuracy. From several TSAs an attributable vector can synthesize the observations, see [58].

From the attributable it is possible to compute an admissible region of orbits bound, which is a compact subset of the range, range rate half plane. The admissible region can be sampled, for example by an optimal triangulation, generating virtual objects, see [59] and [76].

Notice that having a short arc or a VSA depends on the relative dynamics between observer and target. An observation of 10 minutes for a LEO satellite seems enough to obtain some estimation of the orbit (the observed orbit is about 1/10 of the entire orbit), while for a GEO satellite the same observation time seems too short. If the observer is in orbit, observations can last longer period, even if the very high possible relative velocities between target and observer make this case quite unusual.

Notice also that the goodness of the observation depends on the curvature of the observed part of the orbit; the smaller is the curvature, the less accurate is the orbit.

If the operation of linkage between different observations fails or, if just one TSA is available, no information can be obtained. One important and new approach could be based on search algorithms that explore the space of solutions, identifying the orbit that best fits the observations data. Usually infinite orbits match the observations, this thesis is dedicated to determine the best fitting orbit and the class of orbits that could match the data.

Chapter 2

The Search Algorithms

Computers are incredibly fast, accurate and stupid; humans are incredibly slow, inaccurate and brilliant; together they are powerful beyond imagination.

- Attributed to Albert Einstein

This chapter is about search methods. A search algorithm is an algorithm that has to find a solution within a set of solutions. These solutions can be stored and limited, as in a data structure, or may be elements of a continuous search space. An important class of algorithms uses heuristic, that is an experience-based technique for problem solving, learning, and discovery.

2.1 Introduction

Search algorithms are step-by-step procedures that allow to find a solution to a given problem. One of the main differences for search algorithms regards the type of solutions space, also called feasible space. Two big branchs can be identified:

- Search in a discrete space

  The variables used in the mathematical program (or some of them) are restricted to assume only discrete values, such as the integers.

- Search in a continuous space

  The variables used in the mathematical program (or some of them) are real values, e.g., all the values from 0 to 1.

2.2 Search Algorithms for discrete problems

The objects studied in discrete mathematics (such as integers, graphs, and statements in logic) have distinct, separated values; differently from continuous problems where solutions and objects can vary smoothly.

Discrete mathematics therefore excludes topics in *continuous mathematics* such as calculus and analysis. Discrete objects can often be enumerated by integers. More formally, discrete mathematics has been characterized as the branch of mathematics dealing with countable sets, see [7].

A countable set is a set with the same number of elements as some subset of the set of natural numbers. A set that is not countable is called uncountable. The elements of a countable set can be counted one at a time, although the counting may never finish.

In any case, a universally agreed definition of *discrete mathematics* does not exist. Indeed, discrete mathematics is described less by what is included than by what is excluded: briefly no continuously varying quantities.

Several methods have been established to deal with discrete problems. In practice the majority of these methods are fitted to discrete and finite problems. Finite problems are very common in computer science, since data are stored and organized in finite memory. So here is a short list of common methods in searching algoritms:

- Brute force search

  The brute force search, also called exhaustive search, is a very simple method to solve a discrete problem. This method foresees the enumeration of all possible candidate solutions and check if they are really a solution of the problem. This search method is easy to implement but impractical in many situations, altough it always finds a solution if it exists. The cost of this method is proportional to the number of possible solutions, so it is used only when the size of the problem is very little or when it is more convenient to have a simple but very slow algorithm.

- Linear search

  The linear search is a special case of the exhaustive search. This method goes step by step enumerating and checking each solution. Linear search is a very common way to search a solution in computer science. If just one solution is needed, the algorithm stops when one single solution has been found. If all the solutions are requested, linear search is very similar to brute force search, except for the fact that in the second case the checking of the solutions is executed once all the solutions have been evaluated. The scope of this kind of search is reporting the position of the solution. See Fig. 2.1 where the requested solution 13 is found at the position 5.

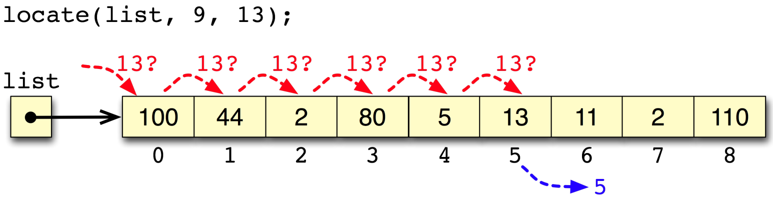


Figure 2.1: The linear search

- Binary search

  Bynary search is an algorithm used for sorted values, the method of sorting can usually be dependent on the user's need. This algorithm finds the position of a certain value in an ordered list splitting the list in two branchs in each step. For example a list of natural numbers is ordered in ascent order and the user wants to find the value 4: the algorithm checks the intermediate value and verifies that $4 < 7$ so excludes all the position that are on the right side of the value 7, see Fig. 2.2. The algorithm proceeds in an iterative way until the matching has been found with the desired value.

- Hash table

  A hash table is a efficient data structure to implement dictionaries. The most common operations in a dictionary are *insert*, *search* and *delete*;

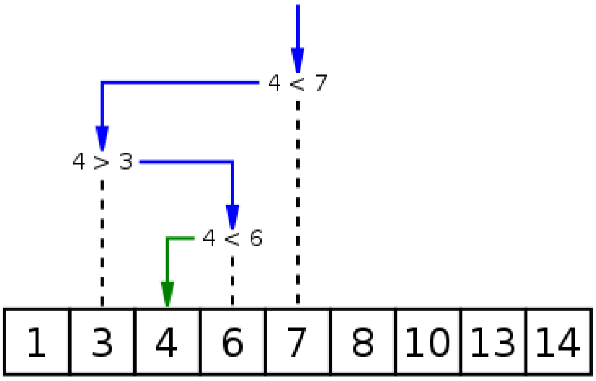


Figure 2.2: The binary search

for example, in a compiler the keys of elements are arbitrary character strings that correspond to identifiers in the language, see [12]. A hash table is a simple generalization of an array, where for each key a value can be directed to. As an example, in Fig. 2.3 there is a simple hash table where a phone number is associated to each person. The hash table is a data structure to implement associative array. Searching in a hash table is a very common problem in a database.

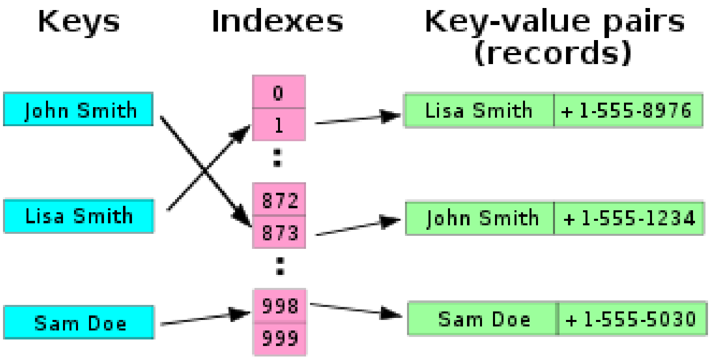


Figure 2.3: The hash table

- String searching

  String searching is very common in dictionaries and text files, very complex as pdf as well. The most fundamental operations in string searching are [71]:

  - Problem: Substring Pattern Matching

- Input: A text string t and a pattern string p
- Output: Does t contain the pattern p as a substring, and if so where?

One of the most simple methods to solve this kind of problem relies on shifting the pattern string over the text, as shown in Fig. 2.4

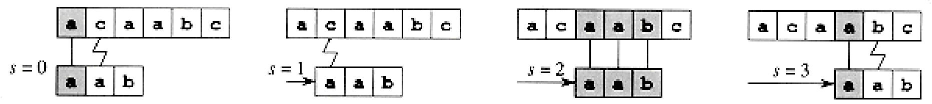


Figure 2.4: The string searching

- Tree Graph Searching

  A particular set of search problems can be dealed with graph representations. Graph teory is the study of mathematical structure used to model pairwise relations between objects [11].

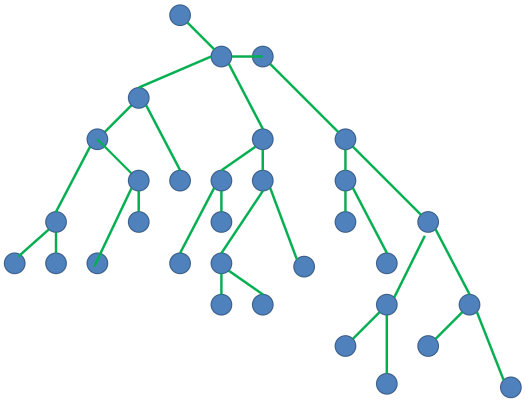


Figure 2.5: A graph example

A search tree is generated by an initial state and a function, the successor function, that define the state space. In general, we may have a search graph rather than a search tree, when the same state can be reached from multiple paths [67].

Various techniques are used to deal with graphs, some of them are listed here:

- Depth-first search always expand the deepest node. The search always proceeds to the deepest level of the search space, where no other successors exist. The order of searching is shown in Fig. 2.6. After the search reached the deepest node it backs up to explore the remaining nodes.

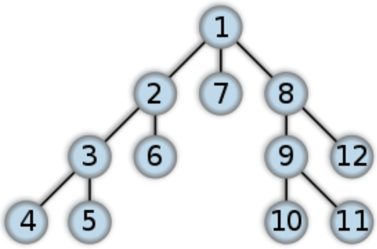


Figure 2.6: A depth-first search

– Breadth-first search is a strategy where the root is expanded first, then all the successors are expanded next, then their successors, as shown in Fig. 2.7. In general all the nodes are expandend to a certain depth, before the depth is increased.

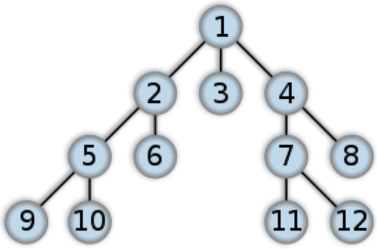


Figure 2.7: A breadth-first search

– Best-first search explores the graph expanding the most promising node chosen according to a given value, this value can be obtained from an evaluation function. This is a sort of heuristic search, where the best promising paths are explored first. Efficient selection of the best nodes is an important point of this approach.

2.3 Famous problems in discrete optimization

There are a lot of very famous problems that have also an ancient tradition. Some of them are also related to games, especially to chess. These problems inspired operations research and linear programming. Integer programming, or combinatorial optimization, can be seen like a subset of operations research dealing with integers.

- Eight queens puzzle

  The eight queens puzzle is the problem of placing eight queens on a typical chessboard so that the queens do not attack theirselves, see Fig. 2.8. So, obviously two queens can not be on the same row, column or diagonal. This problem was proposed in 1850 and has more than one solution, 92 exactly. This problem is used quite often because it is simple but not trivial. Brute-force searching is not feasible for the large number of solutions, altough it is possible to make it very simple noting the symmetries.

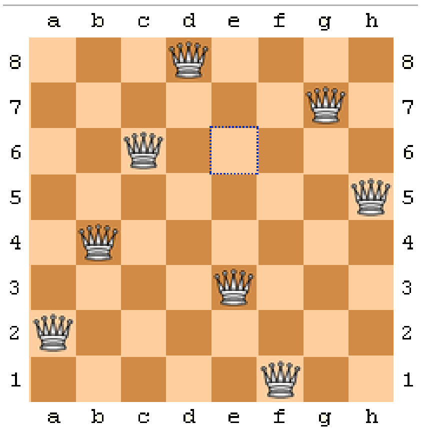


Figure 2.8: The eight queens puzzle

- Knight's tour

  A knight's tour is a sequence of moves regarding a knight that has to visit all the squares just one time. If the knight ends the tour in the begininning square, like in Fig. 2.9, the tour is closed. In this case it is possible to start from every square in the chessboard and proceeding in one direction or the other, knowing that the knight will come back to the starting position.

- The Knapsack problem

  The knapsack problem has been deeply studied since the beginning of the history of operations research, because of its intuitive applications, like in industry or finance, but also for theoretical reasons. The knapsack problem often occurs by relaxing more complex integer programming problems. The knapsack problem requires a subset of some given items to be chosen such that the corrensponding profit sum is maximixed without exceeding the capacity of the knapsack. The knapsack problem

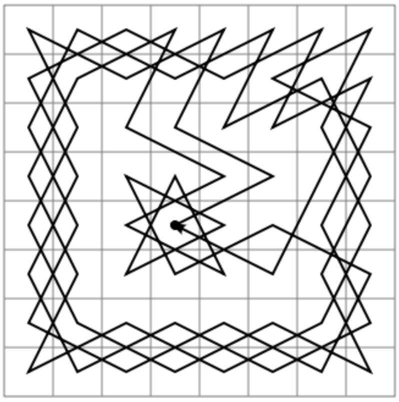


Figure 2.9: The knight's tour

is the first problem here presented that is $NP - hard$, one of the most complex class of algorithm, see Fig. 2.10

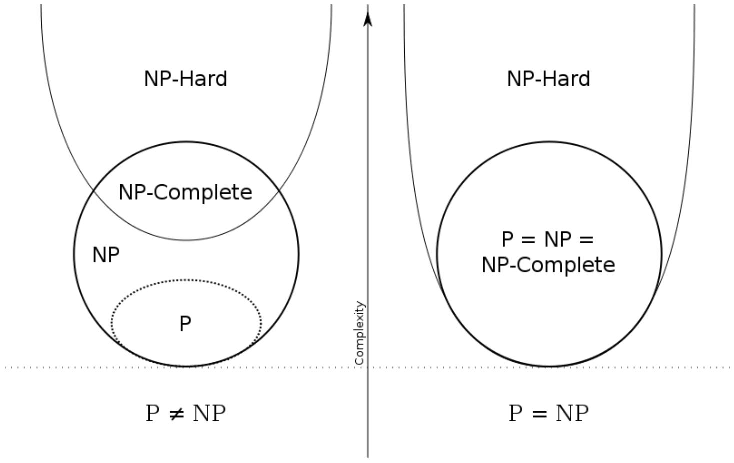


Figure 2.10: P=NP?

This thesis will not go in detail with the computer science related problems and the due formalism regarding this kind of problems, for details see [12]. Notice that in Fig. 2.10 it is highlighted the difference between $P = NP$ and $P \neq NP$, this is one of the most important open problems in computer science. This problem is in the list of the Millennium Prize Problems and can be expressed in this way: *"If the solution to a problem can be quickly verified by a computer, can the computer also solve that problem quickly?"*

There are a lot of scientist that are working in this field but the common approach is that $P \neq NP$.

"If P = NP, then the world would be a profoundly different place than we usually assume it to be. There would be no special value in 'creative leaps,' no fundamental gap between solving a problem and recognizing the solution once its found. Everyone who could appreciate a symphony would be Mozart; everyone who could follow a step-by-step argument would be Gauss..."

Scott Aaronson, MIT

- The traveling salesman problem

  This problem is about one of the most sensible field in combinatorial optimization, it is even used as a benchmark to test the goodness of an algorithm. *Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?*

  For small nets, like in Fig. 2.11, it is possible to find the optimal tour by doing an exhaustive search. For big nets, there are heuristics that will get you very close to the optimal tour [71]. This is a problem that can be related to many fields, as to logistics, planning or microchip manufacturing. Imagine the soldering points like cities and we want to minimize the path length between all the points. This will reduce the time of the execution and the cost of course. Implementation of heuristics for Travelling Salesman Problem (TSP) has led to the decreasing of the cost for chip manufacturing of about 30% [71]. The TSP approach also gave some inputs to obtain better DNA sequencing, where there are millions of DNA fragments and we want to minimize the measurement time between them.

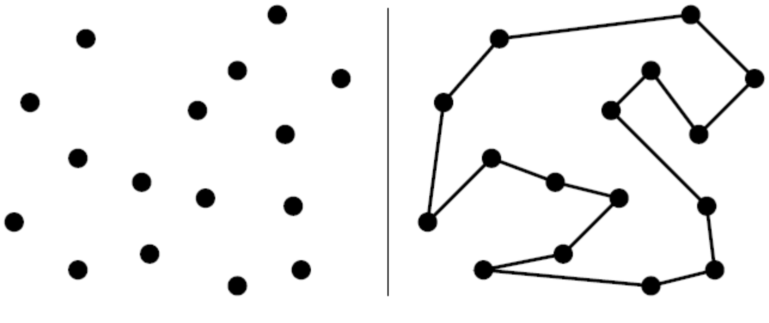


Figure 2.11: The traveling salesman problem

2.4 Search Algorithms for continuous problems

The algorithms described in this part are called heuristic algorithms, in the sense that they use heuristics. Heuristics stems from the Greek word *heuriskein* which means to find or discover. It is used in optimization, in a discrete space but especially in a continuous space, to characterize intuition-based problem solving methods.

There are several reasons to use heuristic methods [51]:

- No method for solving the problem to optimality is known
- Although there is an exact method to solve the problem, it cannot be used on the available hardware
- The heuristic method is more flexible than the exact method, allowing, for example, the incorporation of conditions that are difficult to model
- The heuristic method is used as part of a global procedure that guarantees to find the optimum solution of a problem.

A good heuristic algorithm should fulfill the following properties:

- A solution can be obtained with reasonable computational effort
- The solution should be near optimal (with high probability)
- The likelihood for obtaining a bad solution (far from optimal) should be low

There is a large number of difficult problems that are not solvable with classical methods, but these problems need to be solved efficiently. It is not important to find the best, the optimal, solution, but it is important to find a method that yields very good solutions, sub-optimal, in a given amount of time. Virtually all problems could be solved by an heuristic approach, but in any case the algorithm shall respect four properties [67]:

- Completeness: Is the algorithm guaranteed to find a solution when there is one?
- Optimality: Does the strategy find the optimal solution?
- Time complexity: How long does it take to find a solution?

- Space complexity: How much memory is needed to perform the search?

To understand what heuristics is, we report the examples in [68]:

Let us assume that someone dropped a contact lens. Here are some possibilities for search:

1. Blind search

bending down and feeling around for the lens. Such search does not guarantee a positive result.

2. Methodical search

it consists in expanding the space of research methodically and in an organized way. It always guarantees the success, but is very time-consuming.

3. Analytical search

requires the solution of a mathematical equation concerning the fall of the contact lens, taking into consideration the air resistance, wind power, gravitation. It also guarantees the success, but is impractical.

4. Lazy search

consists in finding the nearest optician and purchasing a new lens.

5. Heuristic search

we define the approximate direction of the fall and we presume how far the lens could fall and then we search the selected area. It is the most natural behavior and we most often unconsciously select this method of proceeding.

2.5 Local search

Local search algorithms use a single state, or multiple states, and move only to neighbors of this state. Usually this class of algorithms does not retain the path, so the path to a given state is not saved. This helps the speed but especially the *space complexity* because the amount of used memory is limited and usually constant. These algorithms can often find good solutions in a reasonable amount of time exploring infinite state spaces.

To understand how local search works, see Fig. 2.12. If the elevation corresponds to a goodness function, the algorithm wants to find the peak of the function, that is the global maximum. Local search algorithms explore this kind of landscape that can be very complex and in many dimensions.

2.5.1 Hill climbing

The hill climbing search technique is probably the most simple algorithm within the local search field. It is simply a loop that moves its best solution changing a single element of the best solution at each step.

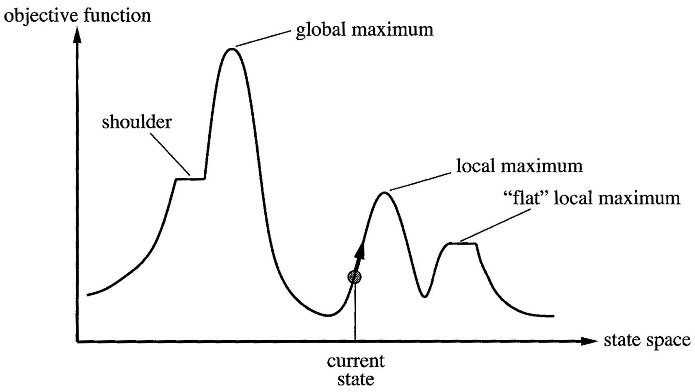


Figure 2.12: A local search

This approach makes the best solution moving towards the direction of increasing values, like climbing an hill. The simplest version of this algorithm uses a single candidate solution and does not look ahead beyond the immediate neighbors of the actual state. Usually this algorithm get stucked in a local maximum because it never moves toward worst solution.

2.5.2 Simulated annealing

Simulated annealing combines a hill climbing algorithm with a random walk. Instead of comparing only the neighbors of the current state, simulated annealing picks a random state in the solution space; if the new state gets a better solution, this will be the new starting solution. The power of the jump is decreasing, in the sense that the random picking is executed inside a local space around the current solution, this local space of picking decreases with time.

In metallurgy, simulated annealing is the process used to temper metals by heating them to high temperatures and gradually cooling them. This process foresees a higher use of energy at the beginning, high temperatures, and a slow cooling phase where the temperature decreases. This algorithm is almost always ”complete”, in the sense that it almost always find a global maximum with a good setting of parameters.

2.5.3 Tabu Search

Tabu search is another variant of the hill climbing search. If the algorithm finds good solutions without improving them for a certain number of steps, the algorithm puts these solutions in memory and they become tabu. This means that the solution can not revisit the old solutions; this approach can make effective the escaping from local maxima. Tabu search can be complete but can use a lot of memory to store the tabu list, especially in a multi-dimensional landscape.

2.6 Local search methods inspired by nature

Many search methods are inspired by nature [85] [64]. Computer scientists imitate the nature to find new methods of solution space exploration. A large class of new algorithms is developed nowadays, often with very little improvement or with some key-variants. Here there is a very short list of some between the well-known algorithms inspired by nature.

2.6.1 Ant Colony Optimization

Ant Colony Optimization (ACO) is an heuristic method in which a colony of artificial ant cooperates to find a good solution in an optimization problem. This method was initially developed for discrete problems [17] and then generalized for a continuous problem [73].

Ants initially walk randomly in the state space, if they find food they return to the nest laying pheromone trails. Other ants, randomly searching, can find this path recognizing the pheromone, so they follow the path. Over time, the pheromone on the trails tends to disappear, reducing their appeal, so it necessary, for the path survival, that periodically many ants visit the same path. This approach tend to guarantee the avoiding of local optimal solutions leading to the exploration of the entire search space.

2.6.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a method with a population of candidate solutions, or particles, that try to get to the global maximum moving in the search space [42]. The moving of the particles is guided through simple cinematic formulas regarding position and velocity. Each particle's motion is influenced by its local best known position and the global best known position in the search space. The global best known position is updated once a new better solution is found. This is a *social* algorithm that can involve thousands of particles, like a bird flock.

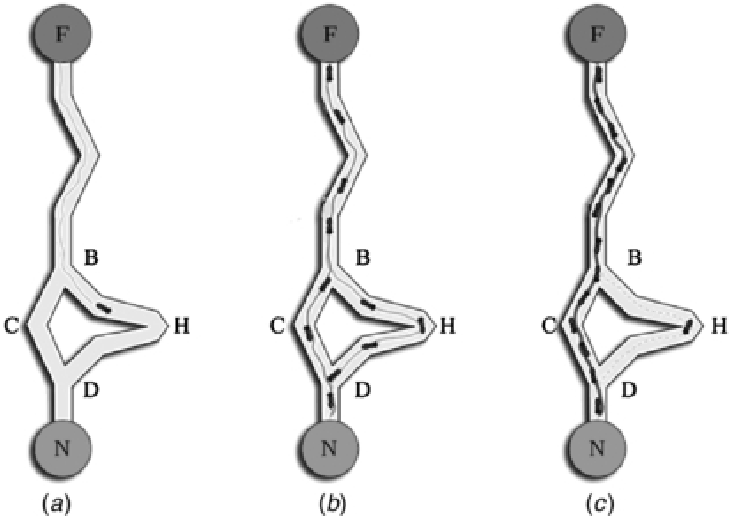


Figure 2.13: Ant colony Optimization

2.6.3 Genetic algorithm

One of the most important point is that genetic algorithms are intrinsically parallel. Like PSO, genetic algorithms can explore the solution space in multiple directions, giving so a greater chance of finding the global optimal solution. The genetic algorithm is based on an iterative process that imitates the social behaviour of a group of individuals that try to survive and generate offspring. After the initialization, a fitness function will judge the goodness of the candidates to decide who will survive and will generate the next offspring. The generation of the new offspring is yielded through genetic operator: crossover and mutation, see Fig. 2.14.

The genetic algorithm has a long tradition in computer science [60] and has set the basics for the evolutionary algorithms [19]. Evolution seems a good way to approach computational problems in many fields. Biological evolutions is a source of insipiration for searching problems. The goodness of a genetic algorithm depends on many factors:

- Set the initial population
- Decide the fitness function
- Choose who and how frequently a solution can replicate
- Choose the percentage of influence for crossover and mutation
- Choose the variance of the applied random noises

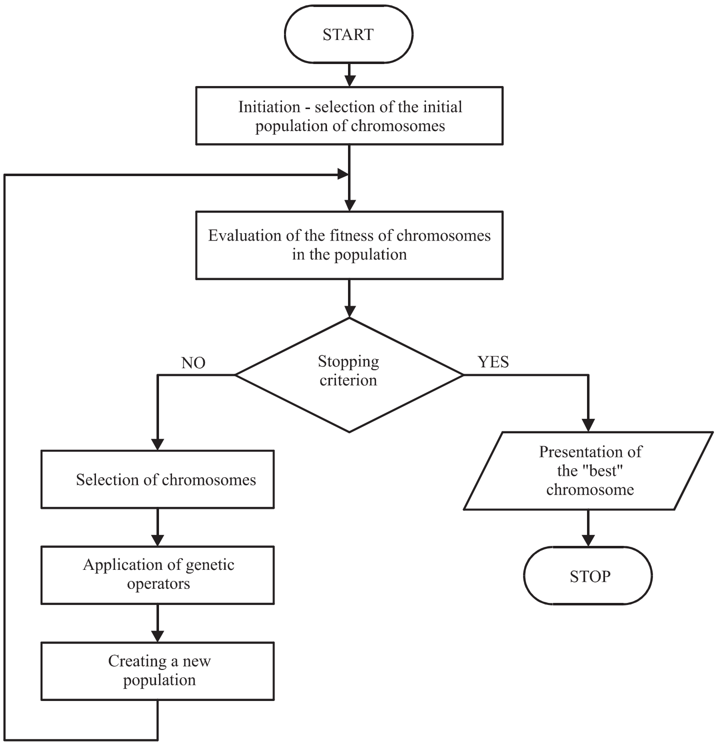

START
Initiation - selection of the initial population of chromosomes
Evaluation of the fitness of chromosomes in the population
NO
Stopping criterion
YES
Selection of chromosomes
Application of genetic operators
Creating a new population
Presentation of the "best" chromosome
STOP

Figure 2.14: The genetic algorithm flow

The random noise is very important for the genetic algorithms, see Appendix A for an overview of the methods to generate a pseudorandom noise.

2.7 Conclusions

This thesis will not go in details with the large class of heuristic or metaheuristic methods. There are some features that can help the users to choose which method could be applied to a given problem.

Thera are methods based on a population of solution, the largest class is composed by nature inspired algorithms, see Fig. 2.15. Inside the class of nature-inspired methods the evolutionary computation play a major role.

Other methods are simply based on local search, or local search with stochastic components. Some methods are memory-less.

In general it is important to have an *anytime* algorithm, that is a method that can give a solution anytime, if the algorithm runs for more time the solution gets better.

It is well-known in literature the problem of No Free Lunch (NFL) [84],

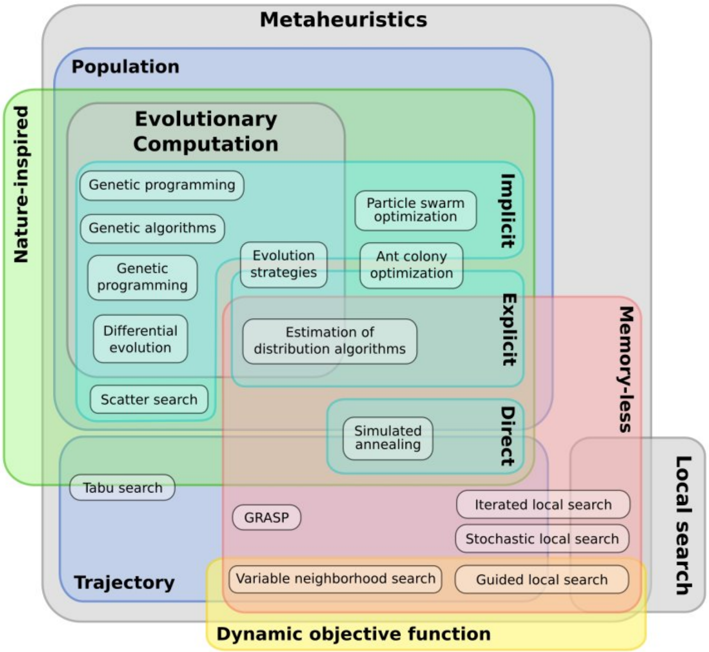


Figure 2.15: The metaheuristic algorithms

that in simple words is: "*a general-purpose universal optimization strategy is theoretically impossible, and the only way one strategy can outperform another is if it is specialized to the specific problem under consideration*" [35].

In other terms, there is no optimization method that we can use like a black-box to solve a problem. It always regards a work of parameter setting, initial population et cetera. This is the main drawback of an heuristic algorithm: virtually it can solve every problem you have but there is no method that will work off the shelf.

The metaheuristic design is based on three main aspects, see Fig. 2.16:

- Parameter setting
- Performance assessment
- Validation

The most important part is the parameter tuning that makes many solutions to a given problem tuning-dependent. Many algorithms need an hard work to make them working in an efficient way, and usually changing something in the environment or in the conditions of the problem highlights the weakness of the algorithm. Some algorithms foresee a sort of auto-tuning parameter to take into account variable conditions with an adaptive approach.

About the performance assessment, speed and precision are very important; usually the user makes a balance between performances and time. A good algorithm is needed to be checked to a very long time, with many and many iterations.

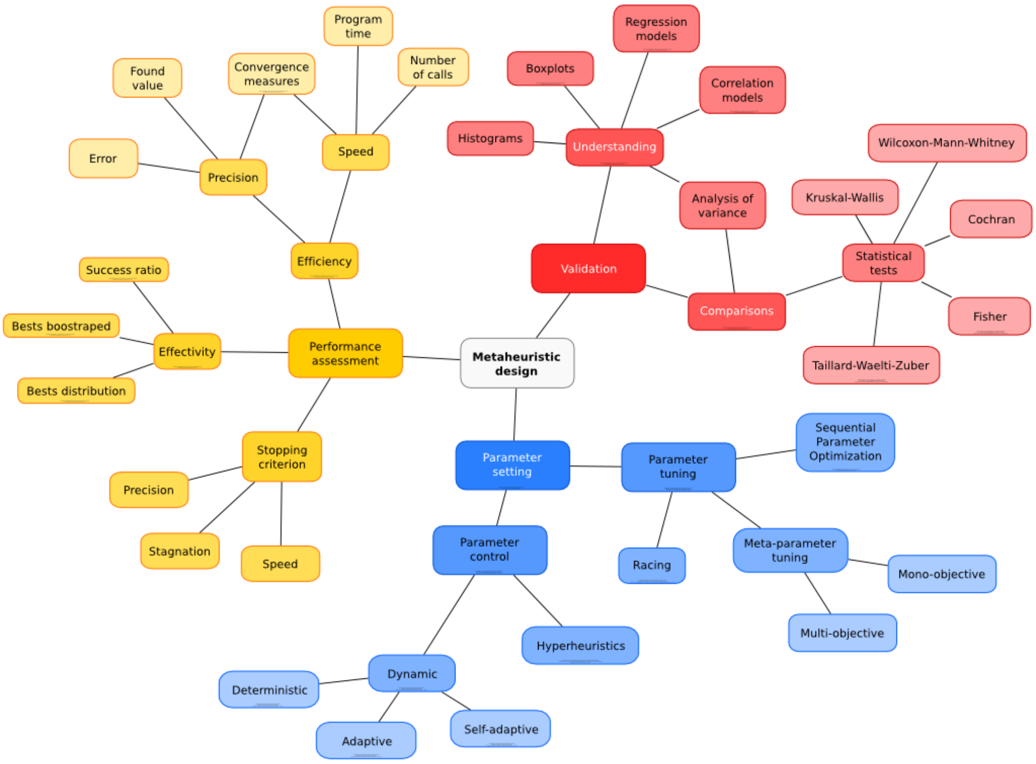


Figure 2.16: Metaheuristic design

Chapter 3

A genetic algorithm for the Initial Orbit Determination

I cant be as confident about computer science as I can about biology. Biology easily has 500 years of exciting problems to work on.

- Donald Knuth, *Computer Literacy Bookshops Interview, 1993*

In this chapter it is described the application of a genetic algorithm to an Initial Orbit Determination problem with Too Short Arc. The algorithm will be applied to a full six orbital parameters state. The initialization will be performed with and withou the NORAD database of orbiting objects.

3.1 Introduction

A search algorithm can solve the problem of Initial Orbit Determination searching in a space of six dimensions. The candidate solution, or invidividual, is so composed by six independent values. The six chosen values are the orbital elements expressed at time t_0, where $t = t_0$ indicates the initial time of the observation.

This algorithm works with no restrictions about the observer location, so it is very useful to test it with space-based observations, see Chapter 7.

Two satellites are orbiting the Earth in two different orbits; satellite illumination or visibility are not taken into account for sake of simplicity. The orbit propagation is carried out with a simple keplerian propagator; the very short time of propagation $T = (t_f - t_0)$ allows this simplification, where t_f is the final time of observation. We calculate the position and the velocity of each satellite in an inertial reference frame and then we propagate the state with a fixed step propagator.

$$\begin{pmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \nu \end{pmatrix}_{t=t_0} \rightarrow \begin{pmatrix} \vec{r} \\ \vec{v} \end{pmatrix}_{t=t_0} \rightarrow Propagation\ for\ T \tag{3.1}$$

The position of the target will be identified with r_T, while the position of the observer will be r_O. The observation vector is simply the unit difference vector between them:

$$\hat{L} = \frac{\vec{r}_T - \vec{r}_O}{|\vec{r}_T - \vec{r}_O|} \tag{3.2}$$

In order to simulate an angular measurement $\hat{L}_{meas}$, a simulated sensor noise will be applied to the true value of $\hat{L}$. The sensor noise will be simply modeled as a guassian random noise, whose standard deviation depends on the accuracy of the sensor. The angular separation between the $\hat{L}$ and $\hat{L}_{meas}$ is so simulated from a normal distribution.

To simulate the measurements we need two auxiliary vectors. The first vector is $\hat{h}$ that represents an orthogonal vector to $\hat{L}$. $\hat{h}$ is obtained through a cross vector between $\hat{L}$ and one of the vectors defining the inertial reference frame.

$$\hat{h} = \frac{\hat{L} \times \hat{k}_i}{|\hat{L} \times \hat{k}_i|} \tag{3.3}$$

It is not important which vector of the reference frame is chosen, the only concern regards the existence of the cross vector, so we have to avoid that $\hat{L}$ is parallel to the chosen vector of the reference frame. To avoid this singularity we always choose the vector $\hat{k}_i$ over which the minimum component of $\hat{L}$ is.

The second auxiliary vector is $\hat{g}$, that is obtained rotating $\hat{L}$ of an angle α from a normal distribution.

$$\hat{g} = \hat{L}\cos\alpha + (\hat{h} \times \hat{L})\sin\alpha \tag{3.4}$$

It is clear from Eq. (3.4) that:

$$\hat{g} \cdot \hat{L} = \cos\alpha \tag{3.5}$$

$\hat{g}$ represents the angular error of the measurement, but we need to rotate this vector with a uniform distribution.

The second rotation rotates the versor $\hat{g}$ of an angle β about the $\hat{L}$ axis, where β is obtained from a uniform distribution between 0 and 2π, see Eq. (3.6).

$$\hat{L}_{meas} = \hat{g}\cos\beta + (\hat{L} \times \hat{g})\sin\beta + \hat{L}(\hat{L} \cdot \hat{g})(1 - \cos\beta) \tag{3.6}$$

$\hat{L}_{meas}$ finally represents the measurement corrupted by the sensor noise. The graphical representation of these two rotations is shown in Fig. 3.1.

3.2 The setting of the genetic algorithm

The problem of orbit determination is based on finding the best orbit that fits the measurements. The problem of IOD is finding the initial orbit and then make differential corrections with the successive measurements to get a better estimation of the orbit [74].

We want to find the orbit to match the measurements with a genetic algorithm, the individual is composed by six independent values, or chromosomes, that correspond to the six orbital parameters at the starting observation time:

$$\begin{pmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \nu \end{pmatrix}_{t=t_0} \tag{3.7}$$

The orbits used to simulate the measurements are in Table 3.1 and in Table 3.2.

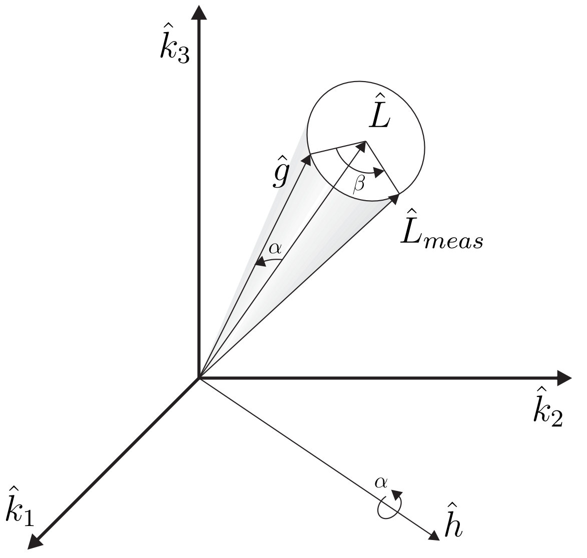


Figure 3.1: The measurement simulation

a	e	i	Ω	ω	ν
7290.20 km	0.0610	30.379°	289.042°	293.776°	66.230°

Table 3.1: The target orbit

The observer is chosen to be in a sun-synchronous orbit because it is a promising orbit to observe a huge amount of objects in different orbital regions.

a	e	i	Ω	ω	ν
7133.88 km	0.0043	98.289°	49.866°	356.768°	149.512°

Table 3.2: The observer orbit

The angular error of the measurements is set to 1 arcsec. The initial population will be composed by 10000 individuals, for each generation the *elite* population will be correspondent to the 10% of the population, so 1000 individuals. The *elite* will create the offspring with a 40% of mutation and 40% of crossover, the remaining 10% will be new individuals create randomly.

The mutation will be applied adding for each chromosome a given value from a gaussian distribution:

$$\begin{pmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \nu \end{pmatrix} \to \begin{pmatrix} a+\sigma_a \\ e+\sigma_e \\ i+\sigma_i \\ \Omega+\sigma_\Omega \\ \omega+\sigma_\omega \\ \nu+\sigma_\nu \end{pmatrix} \tag{3.8}$$

While for the crossover there will be a change between two individuals picked randomly. These two individuals will mix the chromosome, the new individual will have j chromosomes from the first individual and $6-j$ from the second individual. j is a random natural number with uniform distribution between 1 and 5, the values 0 and 6 are excluded to avoid the replication of one parent individual.

$$\begin{pmatrix} a_1 \\ e_1 \\ i_1 \\ \Omega_1 \\ \omega_1 \\ \nu_1 \end{pmatrix} \begin{pmatrix} a_2 \\ e_2 \\ i_2 \\ \Omega_2 \\ \omega_2 \\ \nu_2 \end{pmatrix} \overset{j=3}{\to} \begin{pmatrix} a_1 \\ e_1 \\ i_1 \\ \Omega_2 \\ \omega_2 \\ \nu_2 \end{pmatrix} \tag{3.9}$$

3.3 Initialization with random candidates

The initialization of the first population is crucial to the success of the algorithm. As most of the iterative algorithms, the genetic algorithm needs a good initialization to obtain an affordable convergence of the results. The range of the individuals are in Table 3.3.

Value	Minimum	Maximum
a	6600 km	46600 km
e	0	1
i	0°	180°
Ω	0°	360°
ω	0°	360°
ν	0°	360°

Table 3.3: The range of the chromosomes

The only chromosome free to exceed the range of the initialization is the semimajor axis.

The results of a uniform random initialization are shown between Fig. 3.2 and Fig. 3.4.

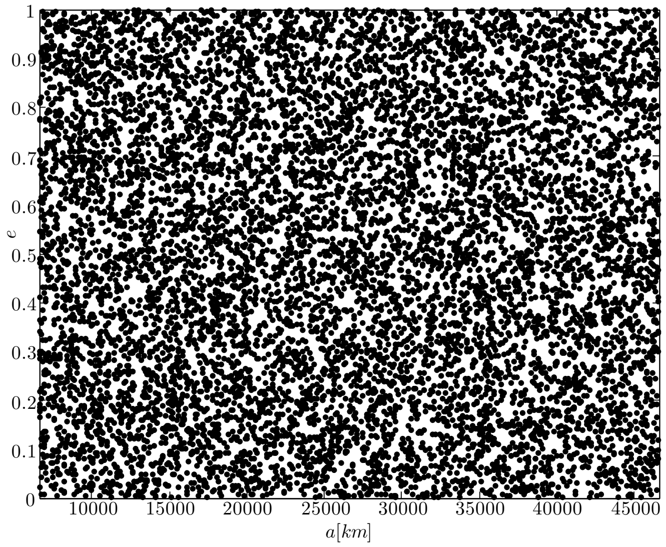


Figure 3.2: $a - e$ **of the initial population**

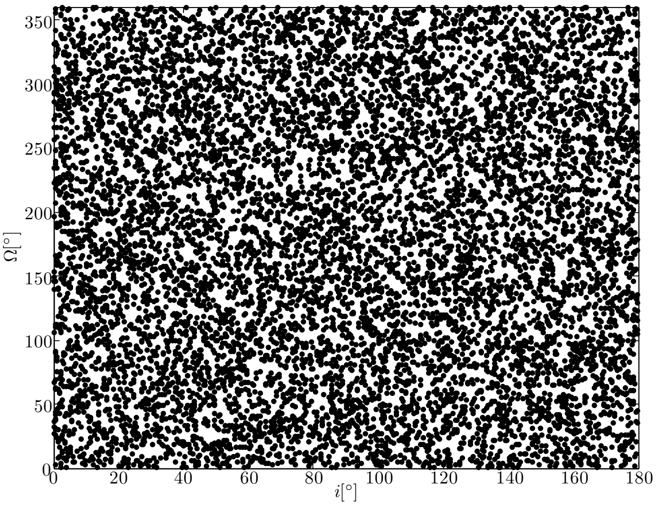


Figure 3.3: $i - \Omega$ **of the initial population**

The fitness function describes how much the individual fits the measurements, in other words how good is a solution. The fitness function gives a score to each individual; there will be a list of best individuals classified by their fitness value and only the best 10% will generate the offspring.

The fitness function uses the measurements $\hat{L}$ and the fictitious measurements $\hat{L}^*$. Each candidate is propagated for the observation time and for each

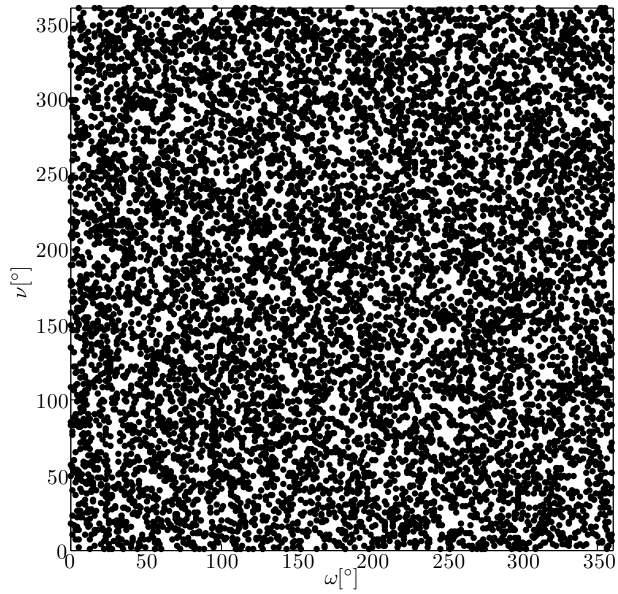


Figure 3.4: $\omega - \nu$ of the initial population

observation at a given time a fictiotious measurement will be produced. The fictiotious measurement corresponds to the measurement that a target with the orbital parameters of the candidate solution would give, see Eq. (3.10).

$$\begin{pmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \nu \end{pmatrix} \rightarrow \begin{pmatrix} \vec{r} \\ \vec{v} \end{pmatrix} \rightarrow Propagation\ for\ T \rightarrow \hat{L}^* \tag{3.10}$$

The fitness function output, called fit, is the multiplication of the dot products of actual and simulated measurements at each time of observation, the expression is in Eq. (3.11).

$$fit = \prod_{i=1}^{n_{obs}} (\hat{L}_{meas}(t_i) \cdot \hat{L}^*(t_i)) \tag{3.11}$$

The fitness function is a value between -1 and 1 but we are interested to values positive near to 1.

$$fit \leq 1, fit = 1 \rightarrow Perfect\ matching \tag{3.12}$$

Two good candidates will have a fitness value very close to 1. An efficient way to highlight the very small differences is transform the fitness value in an angular value, correspondent to the average angle between the true and the reconstructed measurement; this angle is the Equivalent Angular Error (EAE), see Eq. (3.13).

$$EAE = \arccos\left((fit)^{\frac{1}{n_{obs}}}\right) \tag{3.13}$$

The EAE is a value very similar to the Root Mean Square (RMS), the advantage of the EAE is that it is possible to obtain it directly from the fitness value. The EAE should have, more or less, the same value of the angular measurements to show that the genetic algorithm has worked in an efficient way.

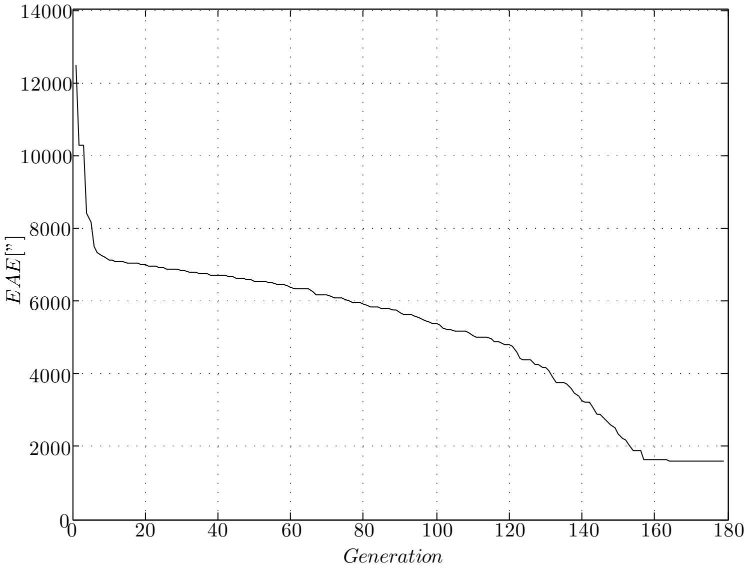


Figure 3.5: EAE with $factor_{mutation} = 400$

The result of the evolution of the EAE is shown in Fig. 3.5. It is clear that this method is not working because the EAE is much larger than the measurement error of 1 arcsec.

One of the most important parameters to set is the value associated to the mutation operator. The Fig. 3.5 refers to a sigma for each chromosome with a $factor_{mutation}$ of 400, that is a gaussian distribution with standard deviation equals to 1/400 of the search space range, see Table 3.4.

$$\sigma_{mutation} = \frac{Search\ space\ range}{factor_{mutation}} \tag{3.14}$$

Value	Search space range	$factor_{mutation}$	$\sigma_{mutation}$
a	40000 km	400	100 km
e	1	400	0.0025
i	180°	400	0.45°
Ω	360°	400	0.9°
ω	360°	400	0.9°
ν	360°	400	0.9°

Table 3.4: The $\sigma_{mutation}$ for $factor_{mutation} = 400$

In this way the algorithm explore the search space with the same $factor_{mutation}$. Notice that the number of generations reached about 180.

We can change the $factor_{mutation}$ by increasing it of a factor 10. The results are shown in Fig. 3.6 and Table 3.5.

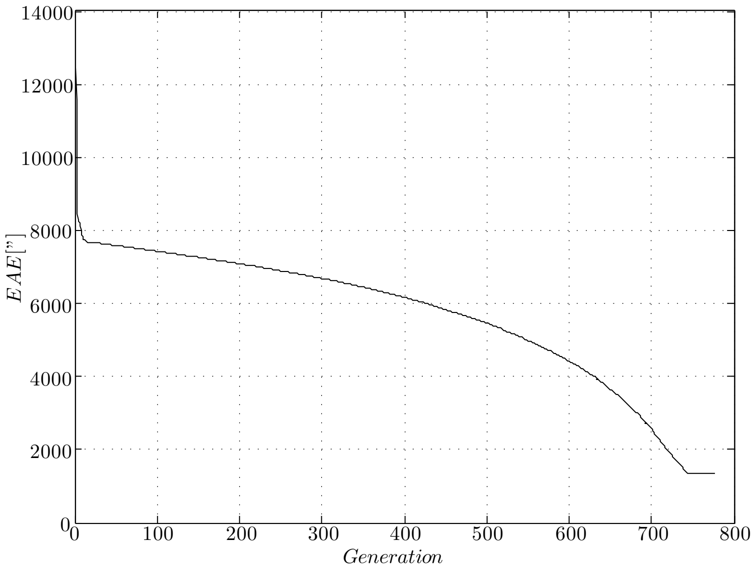


Figure 3.6: EAE with $factor_{mutation} = 4000$

Notice that the number of generations reached about 800. We can again increase the $factor_{mutation}$ of a factor 10, the results are in Fig. 3.7 and Table 3.6.

Notice that the algorithm has been stopped at 5000 generations, the convergence was very slow and the value of EAE was still too high. This behaviour is well known for the genetic algorithms [19], they tend to have a very slow convergence makin them unpractical in many fields.

Value	Search space range	$factor_{mutation}$	$\sigma_{mutation}$
a	40000 km	4000	10 km
e	1	4000	0.00025
i	180°	4000	0.045°
Ω	360°	4000	0.09°
ω	360°	4000	0.09°
ν	360°	4000	0.09°

Table 3.5: The $\sigma_{mutation}$ for $factor_{mutation} = 4000$

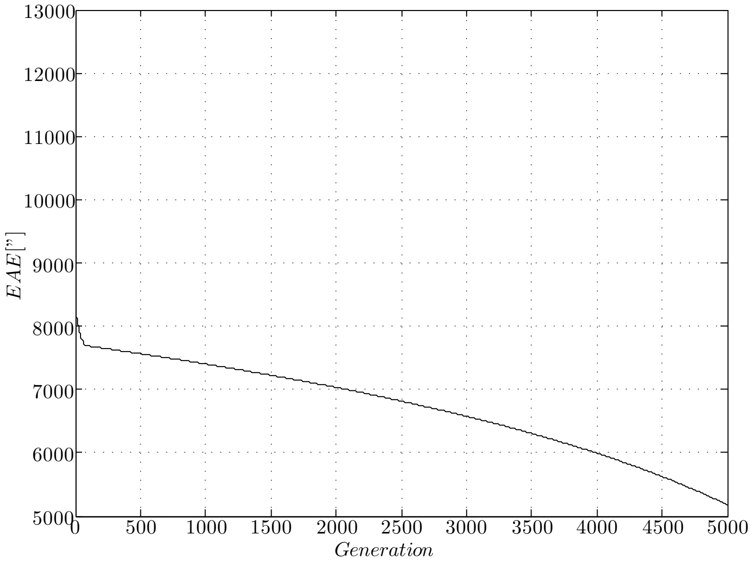


Figure 3.7: EAE with $factor_{mutation} = 40000$

3.4 The NORAD database

The North American Aerospace Defense Command (NORAD) publishes a catalog that contains the TLE of many satellites that are tracked on a daily basys.

A NORAD two-line element set consists of two 69-character lines of data which can be used together with NORAD's SGP4/SDP4 orbital model to determine the position and velocity of the associated satellite [37].

The NORAD database is an almost complete catalog of the orbiting satellites of the Earth. It could be useful to initialize the genetic algorithm with this catalog; notice that the algorithm does not look for the object in the database that best fits the data.

The distributions of the tracked object are shown between Fig. 3.8 and

Value	Search space range	$factor_{mutation}$	$\sigma_{mutation}$
a	40000 km	40000	1 km
e	1	40000	0.000025
i	180°	40000	0.0045°
Ω	360°	40000	0.009°
ω	360°	40000	0.009°
ν	360°	40000	0.009°

Table 3.6: The $\sigma_{mutation}$ for $factor_{mutation} = 40000$

Fig. 3.10. It is possible to see that the distribution is not casual.

A large part of the tracked objects are in LEO, especially in orbits with very high inclinations, the near-polar orbits. Thousands of objects are tracked from the 2007 Chinese anti-satellite missile test and the 2009 Cosmos-Iridium collisions.

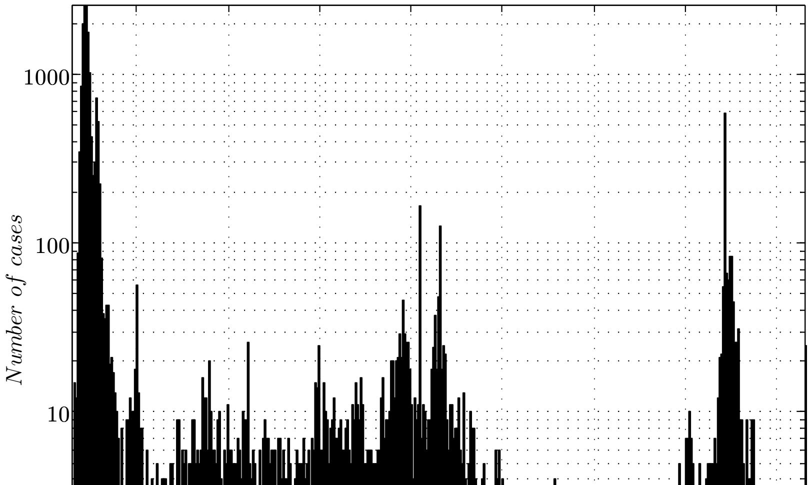


Figure 3.8: Histogram of the semimajor axis of the NORAD objects

The histograms are shown with a logarithmic scale to better interpret the results.

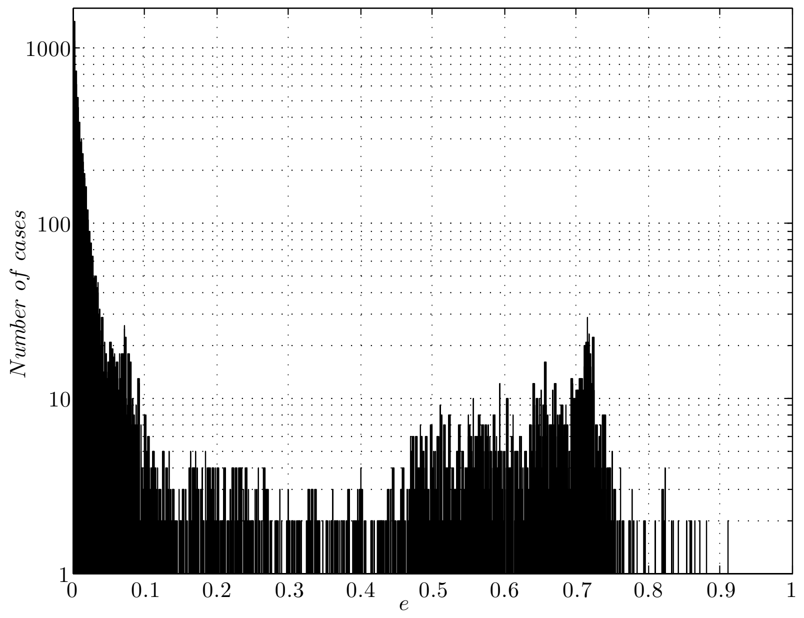


Figure 3.9: Histogram of the eccentricity of the NORAD objects

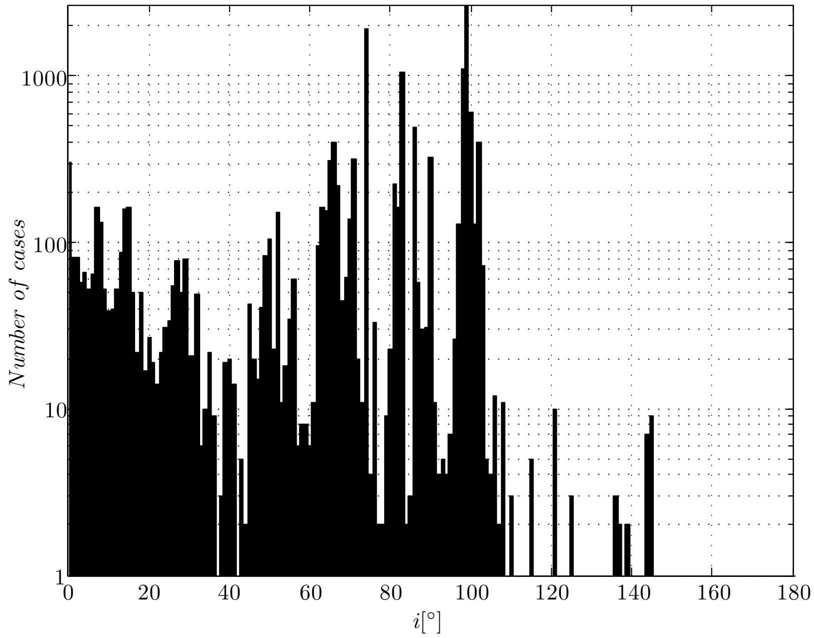


Figure 3.10: Histogram of the inclination of the NORAD objects

3.5 Initialization with the NORAD database

The NORAD database can give a good set for the initialization of the genetic algorithm because, first of all, probably the object that we have observed is tracked and catalogued; if it is not, probably it is in an orbit very close to the catalogued satellites.

There is a class of satellites, operative and not, that are very large and easy to see, also with very small telescopes, that are not catalogued. These satellites, also called spy satellites, are usually very easy to spot and it is possible to determine their orbits.

The results of the initialization are shown between Fig. 3.11 to Fig. 3.13

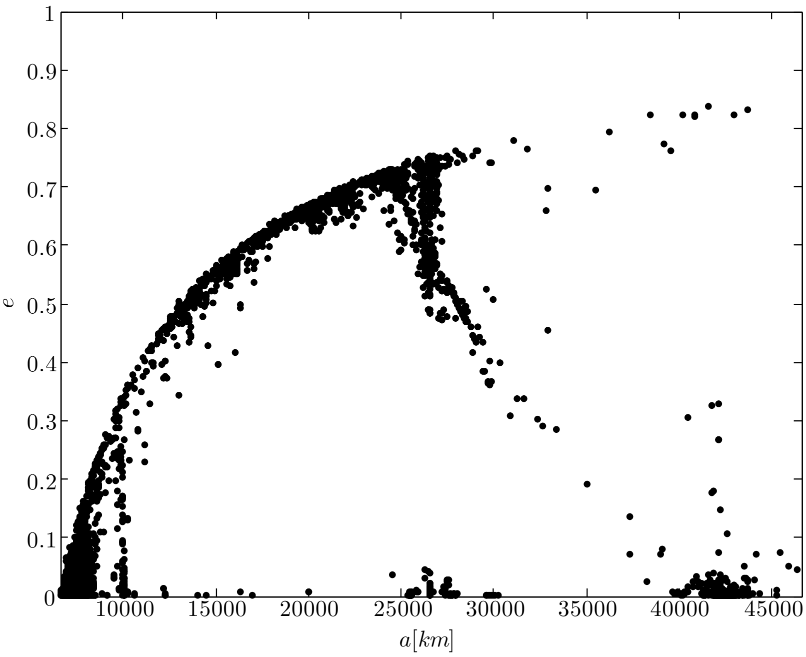


Figure 3.11: $a - e$ **of the initial population with NORAD database**

Notice that in Fig. 3.11 there is a large part of the search space that is not occupied by satellites. The dimensions of the Earth make a lot of orbits not feasible because a lot of satellites could intercept our planet.

Imposing that the distance at perigee is higher than the radius of the Earth is a smart idea to reduce the search space. Actually, it is more convenient to impose that the distance at perigee is out of the atmosphere, because we are interested in stable orbits.

In Fig. 3.14 the unfeasible region is shown in grey; notice that all the satellites tracked are out of this region.

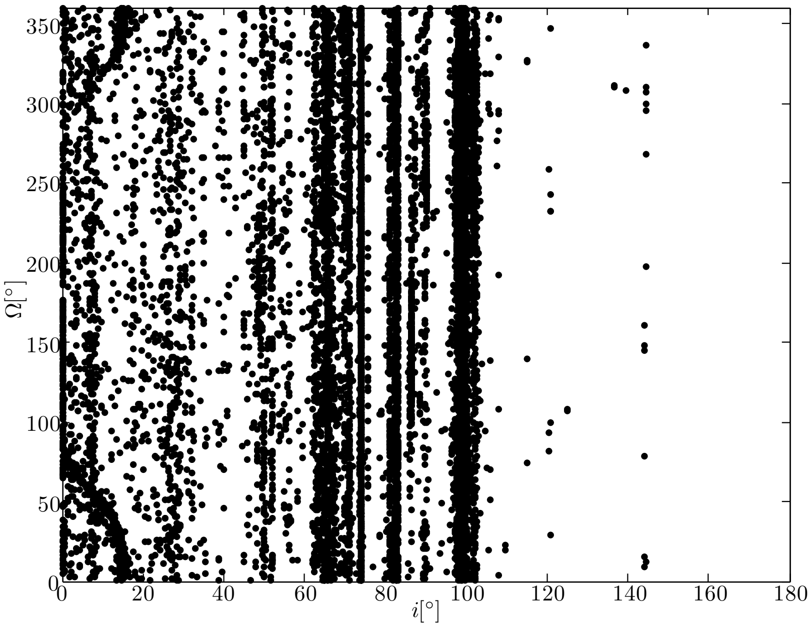


Figure 3.12: $i - \Omega$ **of the initial population with NORAD database**

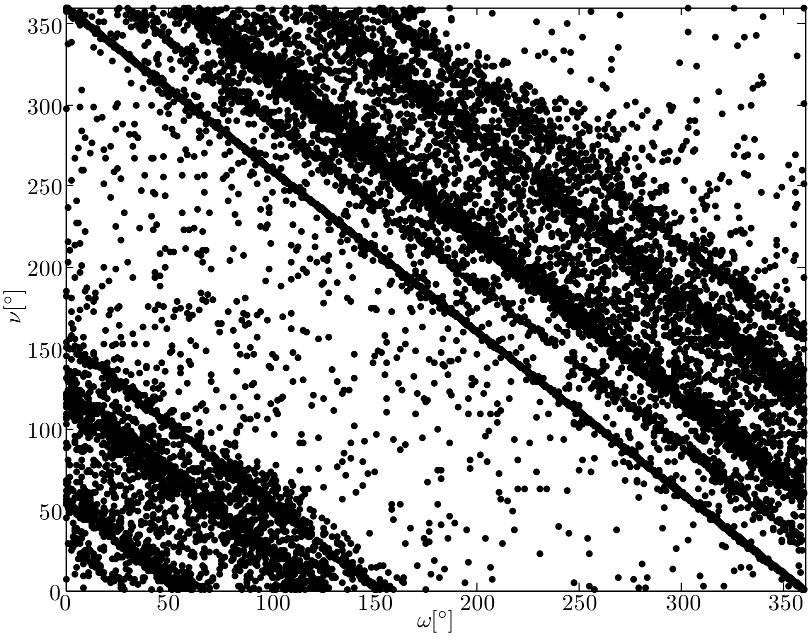


Figure 3.13: $\omega - \nu$ **of the initial population with NORAD database**

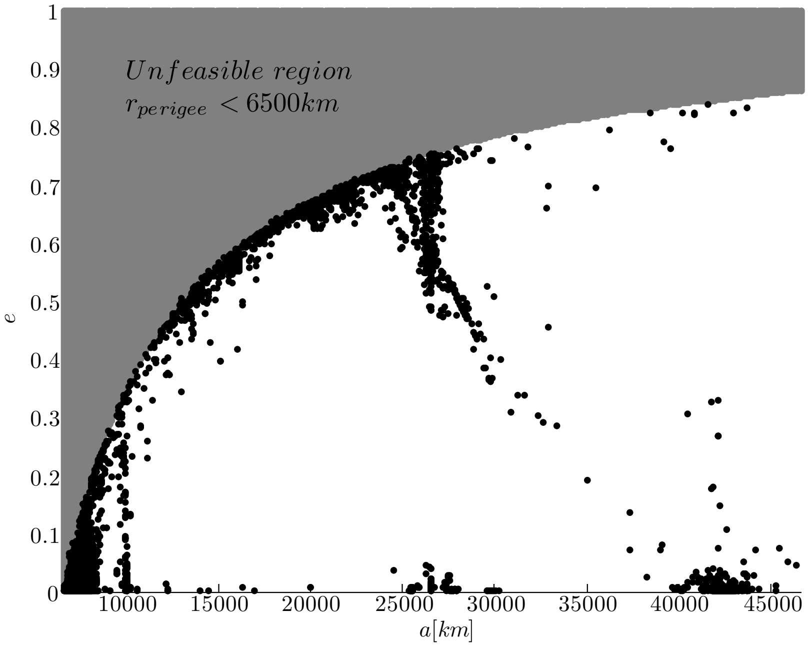


Figure 3.14: The unfeasible region in the $a - e$ plane

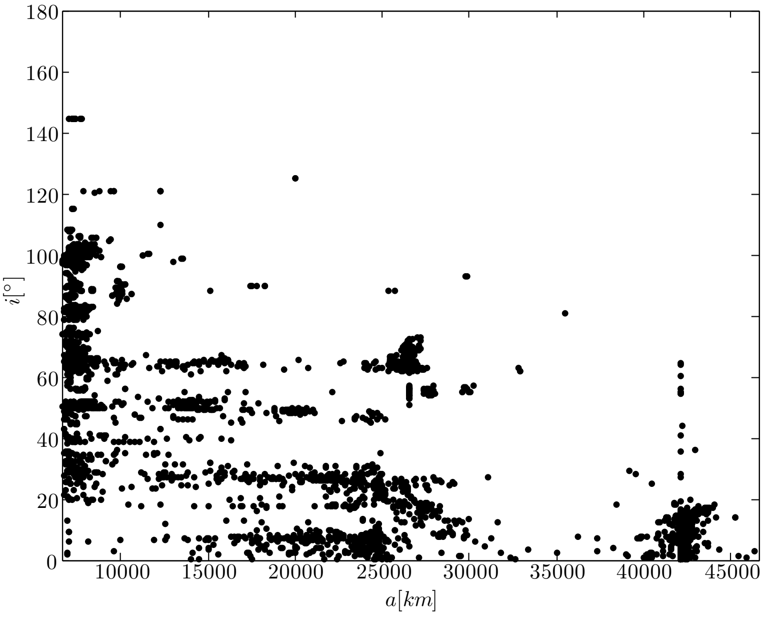


Figure 3.15: $a - i$ of the initial population with NORAD database

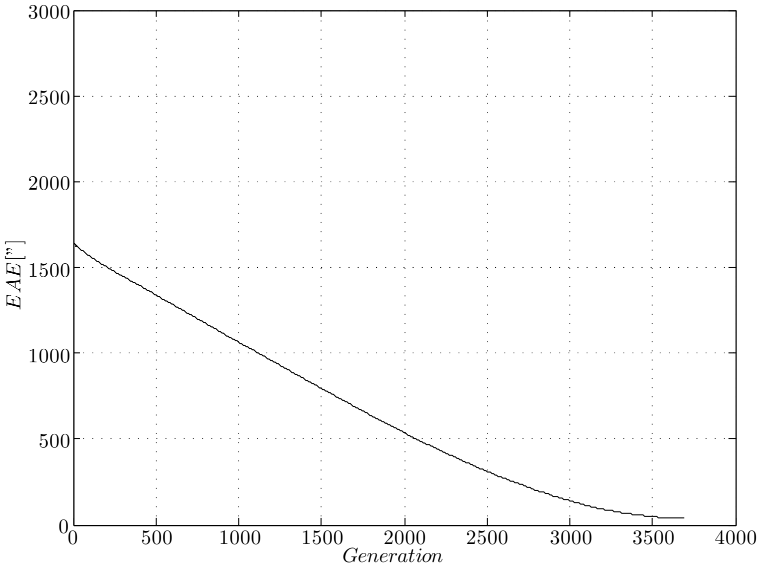


Figure 3.16: EAE with $factor_{mutation} = 40000$ **with NORAD database**

The results regarding the EAE with different $factor_{mutation}$ are shown between Fig. 3.16 and Fig. 3.18.

Notice that increasing the $factor_{mutation}$ increases the number of generations needed to converge, but giving better results.

With a $factor_{mutation}$ of 40000, the final EAE is about 40". It is possible to think about a new increase of the $factor_{mutation}$, but it would be unpractical because of the thousands of the generations needed to obtain stable results.

Notice also that this algorithm is very sensible to the seed of the random noises. In Fig. 3.18 the seed of all the random noises have been changed to see how the results are changing.

The oscillations of the results show that this algorithm is too sensible to little changes of the conditions of the stochastic components. This behaviour makes the algorithm not reliable. A different approach will be explored in the next chapter.

The result about the determination of semimajor axis, eccentricity, inclination and RAAN are shown in detail in Fig. 3.19 and Fig. 3.20.

The orbital plane is well determined; while the orbit shape is not identified, all the elite of the last generation of the algorithm does not get close to the true value. The algorithm seems to prefer a more circular orbit, where probably there is a local minimum.

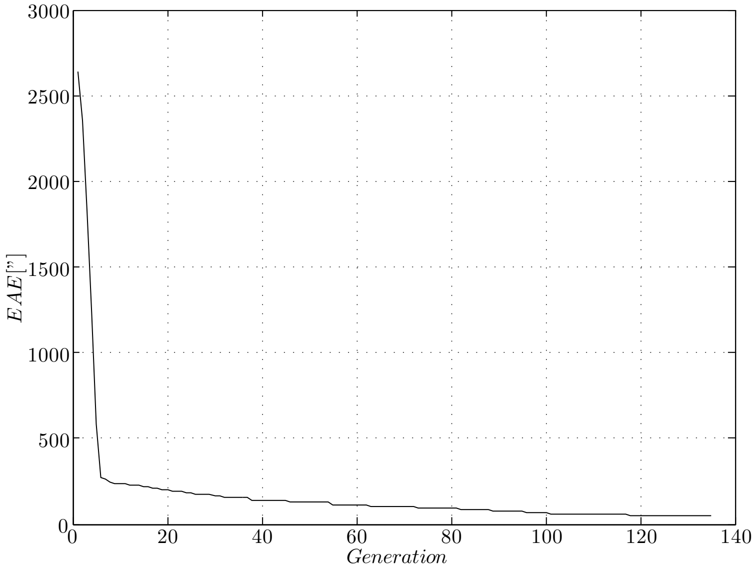


Figure 3.17: EAE with $factor_{mutation} = 4000$ with NORAD database

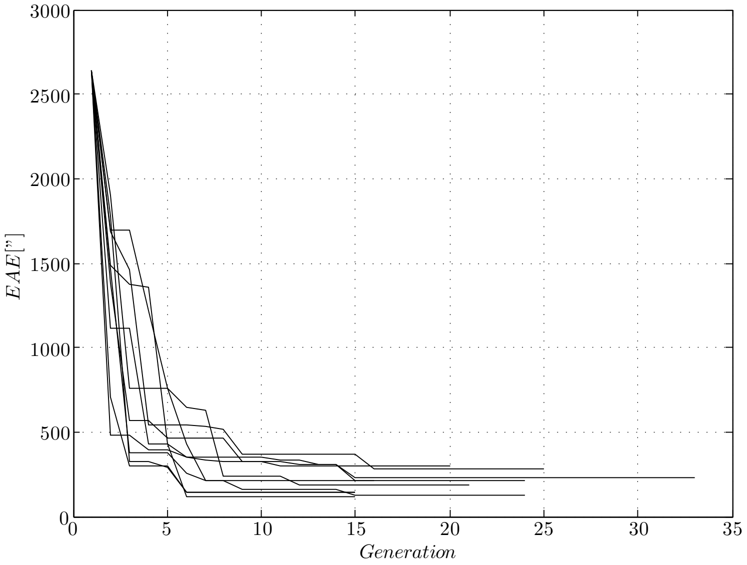


Figure 3.18: EAE with $factor_{mutation} = 400$ with NORAD database

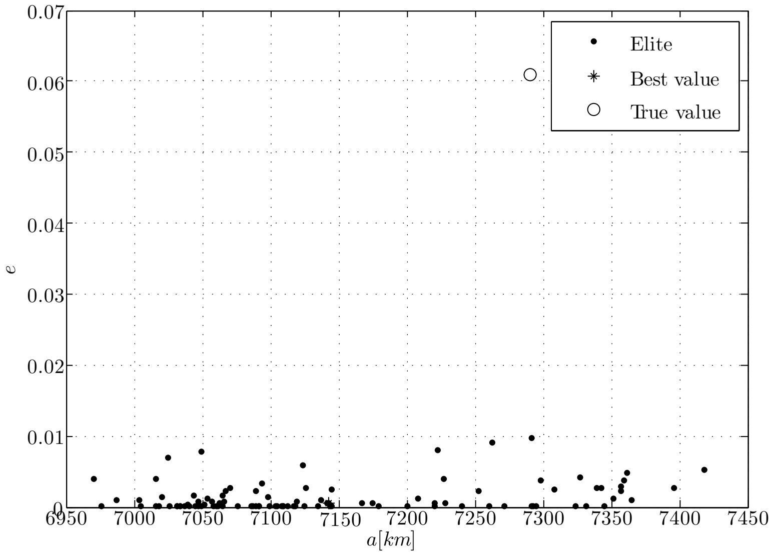


Figure 3.19: $a - e$ of the final elite

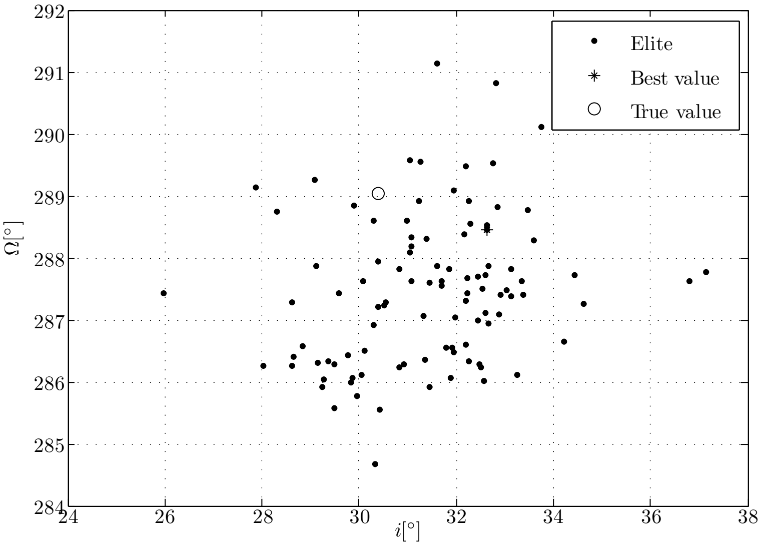


Figure 3.20: $i - \Omega$ of the final elite

Chapter 4

A Genetic algorithm in two dimensions

I use the term Struggle for Existence in a large and metaphorical sense, including dependence of one being on another, and including (which is more important) not only the life of the individual, but success in leaving progeny.

- Charles Darwin, *Origin of Species, 1859*

In this chapter the orbit determination problem will be treated in a two dimensional space, transforming the original problem to a Lambert's problem.

4.1 Introduction

The growth of objects orbiting the Earth is becoming an important issue for the use and the preservation of space. Very accurate knowledge of the orbits of active satellites and space debris is fundamental to prevent collisions and the successive increase of space debris. This requires precise determination of the orbits of these objects, and periodic measurements are needed to deal with the presence of orbital perturbations and uncertainties encountered in the orbital determination.

Many observations are too short to allow an accurate orbit determination, and therefore, creating the problem of correlating various observations to the same object to obtain a good determination; see [56] and [53]. Some works are strictly related to the problem of space debris: [76] and [22]. Furthermore, many measurements with improved telescope capabilities could find new objects with no correlation to a database or other measurements. We present a new algorithm to deal with very short passes. This genetic algorithm allows each pass to yield a candidate orbit; that is, the solution that fits best to the observations. There are no restrictions on the duration of the pass, nor on the position of the observer, therefore this algorithm could be very suitable for space-based observations.

In [40] and [72], the problem of space-based observations has been introduced. Several authors have dedicated their efforts in finding solutions for new algorithms for initial orbit determination, see [49] and [52], also with numerical methods [47] and a Multiple Shooting Method [48].

The issue of correlation of space-based observations has also been addressed before, see [28] and [14]. Some studies are focused on GEO surveillance, [86], [26], with space-based radars as well [62]. An interesting optical architecture for Space Surveillance has been proposed in [24]. Of great interest are the works of [1] and [21] that propose star trackers as sensors for space surveillance. Space-Based Surveillance has already been practiced with the MSX/SBV satellite, [70].

A more recent satellite is now in full operative mode, SBSS 10 Block of the United States Air Force, while the Canadian Department of National Defense is in the process of developing a new system, [54]. All the efforts, as well as the development of new IOD methods; see also [80], based on [33], prove the increasing interests of space based observations with new satellites. Nevertheless, there is no simple procedure to obtain an orbital estimate from a TSA with no correlations. Classical algorithms, based on Laplace, Gauss or Escobal (also known as Double-r) methods, see [79], give very poor results with TSA observations. This chapter focuses on this specific issue, presenting an algorithm that, considering the short time of observation, will yield a very good estimation of the orbital parameters. We will consider a single observation with dense data, as defined in [78], consisting in one observation per second for a period shorter than two minutes, see [2].

4.2 Problem Description

A genetic algorithm is a search algorithm that works on heuristic principles [30]. The best candidates are selected within the initial population with a fitness function; the best candidates, also called the elite, will generate the offspring through genetic operators. The new generation passes the same process creating a new population for each step until a certain limit has been reached. In this process we identify some main elements: the structure of the chromosomes, the initial population, the fitness function, the genetic operators and the stopping condition.

The chromosome has to be chosen to represent the state of the solution to our problem. We want to identify the classical six independent orbital parameters; hence, one could think of a chromosome constituted by six parameters (the orbital parameters or position and velocity); however, it is more convenient to address the problem in a minimal representation that can reduce the state dimensions.

If we consider that, using GPS, the position of the observer is usually known with very high accuracy, in the order of 1 meter, see [10] or [81], and that the angular errors of the measurements can be very low (in the order of 1 arcsecond), we can simply use two scalars to fully represent the state of the target. With these hypotheses, it is possible to solve the orbit determination issue as a Lambert's problem, using the relative distance between observer and target at two different times. The Lambert's boundary value problem consists in finding the orbit of a generic object between two positions at two different times:

$$\begin{cases} \ddot{\vec{r}}(t) = -\dfrac{\mu}{r^2(t)} \cdot \vec{r}(t) \\ \vec{r}(t_1) = \vec{r_1} \\ \vec{r}(t_2) = \vec{r_2} \end{cases} \tag{4.1}$$

where $\vec{r}$ is the object position, μ is the Earth gravitational constant, and $\vec{r_i}$ is the object position at time t_i (we assume $t_2 > t_1$), and all the vectors are expressed in an inertial reference frame.

We can express the target position as a function of the observer position using the relation:

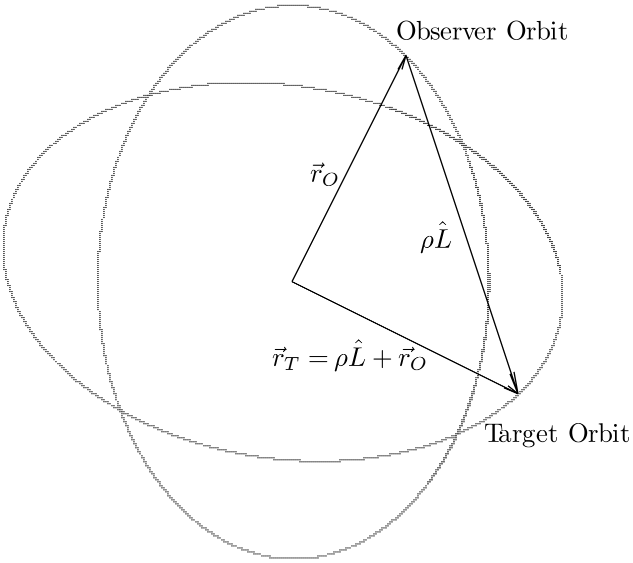


Figure 4.1: The Target orbital position respect to the Observer

$$\vec{r}_T = \rho\hat{L} + \vec{r}_O \tag{4.2}$$

where $\vec{r}_T$ is the target position, $\vec{r}_O$ is the observer position, ρ is the observer-target range and $\hat{L}$ is the unit vector from the observer to the target, as shown in Fig. 4.1.

$\hat{L}$ corresponds to an angular noiseless measurement. A random noise in a random direction has to be added to each measurement to simulate the real world. To introduce the random noise we set a strategy that needs two rotations. Let $\hat{L}$ be $\hat{L} = l_1\hat{k}_1 + l_2\hat{k}_2 + l_3\hat{k}_3$, where $\hat{k}_1$, $\hat{k}_2$ and $\hat{k}_3$ are the unit vectors of the inertial reference frame; then we select $\hat{k}_i$ such that $\hat{L} \times \hat{k}_i \neq \vec{0}$. Let α be an angle obtained from a gaussian distribution with the deviation standard of the sensor; then we rotate $\hat{L}$ about the axis $\hat{h} = (\hat{L} \times \hat{k}_i)/|\hat{L} \times \hat{k}_i|$ of an angle α. As a consequence we obtain the versor $\hat{g}$ that, using the Rodrigues' rotation formula, is given by:

$$\hat{g} = \hat{L}\cos\alpha + (\hat{h} \times \hat{L})\sin\alpha \tag{4.3}$$

The second rotation rotates the versor $\hat{g}$ of an angle β about the $\hat{L}$ axis, where β is obtained from a uniform distribution between 0 and 2π, see Eq. (4.4).

$$\hat{L}_{meas} = \hat{g}\cos\beta + (\hat{L} \times \hat{g})\sin\beta + \hat{L}(\hat{L} \cdot \hat{g})(1 - \cos\beta) \tag{4.4}$$

$\hat{L}_{meas}$ represents the measurement corrupted by the sensor noise.

In these terms to solve the Lambert's problem we simply need to determine the two distances between the observer and the target at two different instants. The Lambert's problem takes into account multiple revolutions, although the very short arc surely involves the short way between the two points with no complete revolutions. Under these assumptions, there is always a unique solution to solve the problem [79]. It is convenient to use the initial time t_0 and the final time t_f of the pass, to get the maximum arc of the observation; hence the chromosome of the genetic algorithm is constituted by ρ_0 and ρ_f, and we write:

$$\begin{cases} \ddot{\vec{r}}_T(t) = -\dfrac{\mu}{r_T^2(t)} \cdot \vec{r}_T(t) \\ \vec{r}_T(t_0) = \rho_0 \hat{L}_{meas}(t_0) + \vec{r}_O(t_0) \\ \vec{r}_T(t_f) = \rho_f \hat{L}_{meas}(t_f) + \vec{r}_O(t_f) \end{cases} \tag{4.5}$$

$\hat{L}_{meas}(t_0)$ and $\hat{L}_{meas}(t_f)$ are the angular measurements corresponding to the initial and the final times of the observation. Notice that the maximum relative velocity between two orbiting bodies multiplied by $(t_f - t_0)$ is always bigger than the maximum of $|\rho_f - \rho_0|$:

$$\max(|\rho_f - \rho_0|) < max(|v_{rel}|)(t_f - t_0) \tag{4.6}$$

The maximum velocity of the Observer Satellite in a generic elliptic orbit can be expressed as [6]:

$$\max|\vec{v}_O| = v_{O_{perigee}} = \sqrt{\frac{(1+e_O)\mu}{(1-e_O)a_O}} \tag{4.7}$$

where $\vec{v}_O$ is the velocity of the observer, e_O is the orbital eccentricity and a_O is the semi-major axis of the observer. The distance at perigee can be expressed as:

$$r_{O_{perigee}} = (1-e_O)a_O \tag{4.8}$$

The maximum relative velocity between the target and the observer occurs if they share the same orbital plane passing at their own perigee at the same time with opposite velocity. The perigee position is to be the same for both satellites, so this is the condition of a conjunction with maximum relative velocity. Imposing that $r_{T_{perigee}} = r_{O_{perigee}}$ and with simple algebraic operations using Eq. (4.7) and Eq. (4.8), we can obtain the maximum value of the target velocity:

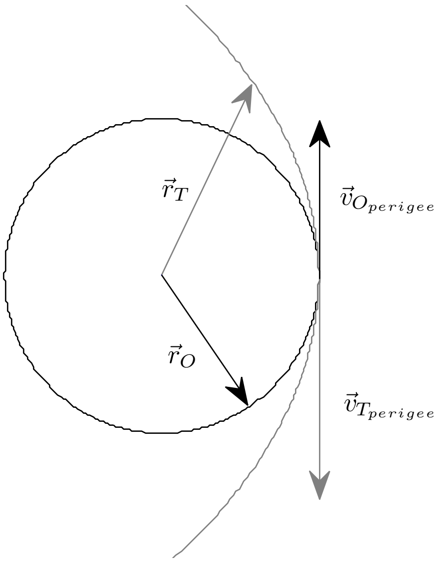


Figure 4.2: The case of maximum relative velocity

$$max(v_{T_{perigee}}) = \sqrt{\frac{2\mu}{r_{O_{perigee}}}\left(1 - \frac{r_{O_{perigee}}}{2a_T}\right)} \tag{4.9}$$

It is easy to note from Eq. (4.9) that $max(v_{T_{perigee}})$ is obtained with an infinite a_T, and such velocity corresponds to the escape velocity. For practical reasons, we can set a maximum a_T at 50000 km, because very few objects have higher values. However, a_T is free to exceed this value since it is only used for the initialization. So, $max(|v_{rel}|) = (v_{O_{perigee}} + max(v_{T_{perigee}}))$, and substituting this expression in Eq. (4.6) we obtain:

$$\max(|\rho_f - \rho_0|) < (v_{O_{perigee}} + max(v_{T_{perigee}}))(t_f - t_0) = D \tag{4.10}$$

The D parameter is useful to initialize the genetic algorithm. The sum of the scalar velocities considers a close encounter at the perigee, as depicted in Fig. 4.2. Notice that this parameter does not vary with a different target.

4.3 Genetic Algorithm for Initial Orbit Determination

4.3.1 Initialization

The genetic algorithm will use a population with a fixed number of 1000 chromosomes; each of which will be constituted by just two values, ρ_0 and ρ_f, so we can depict each solution as a point in a two dimensional space. Each chromosome determines a unique orbit, regardless of the number of observations. The initialization of the algorithm is not executed randomly. A grid of equidistant points will be generated between the space characterized by the value of D, see Eq. (4.10), so that the candidate solutions must be between the lines with equation $\rho_f = \rho_0 + D$ and $\rho_f = \rho_0 - D$, thus the region could be defined as:

$$\begin{cases} \rho_f < \rho_0 + D \\ \rho_f > \rho_0 - D \end{cases} \tag{4.11}$$

We have applied a simple rule for the selection of the best chromosomes, by simply using the initial and the final measurements. Lambert's problem is solved for each chromosome; that is, for each chromosome a unique orbit will be obtained. The solutions were obtained by applying a fast algorithm from the GTOP Database developed by ESA's Advanced Concepts Team (ACT); this is a database containing the exact definition of some global optimization spacecraft trajectory problems and their best putative solutions. The solver of the Lambert's problem, used in [38], uses a new parameterization to obtain almost immediately a reliable result with an iterative process; if the convergence of this algorithm is not reached in few steps, a more classical algorithm based on [46] and [32] will be used.

Once the orbit is determined, the vector $\vec{v}_{t_0}$ will be obtained. The position and the velocity at a certain time corresponds to a certain orbit; now it is possible to express all the orbits with the classical orbital parameters: the semi-major axis a, the eccentricity e, the inclination i, the RAAN Ω, the argument of perigee ω and the true anomaly ν. It is convenient to impose that:

$$\begin{cases} R_\oplus + 200km \leq a \leq a_{MAX} = 50000km \\ 0 \leq e \leq e_{MAX} = \left(1 - \dfrac{r_{O_{perigee}}}{a_{MAX}}\right) \end{cases} \tag{4.12}$$

with $R_\oplus$ as the Earth radius. The values used for the observer are shown in Table 4.1; in this case $r_{O_{perigee}} = 7102.85km$, $v_{O_{perigee}} = 7.51km/s$,

a	e	i	Ω	ω	ν
7133.88 km	0.0043	98.289°	49.866°	356.768°	149.512°

Table 4.1: The observer orbit

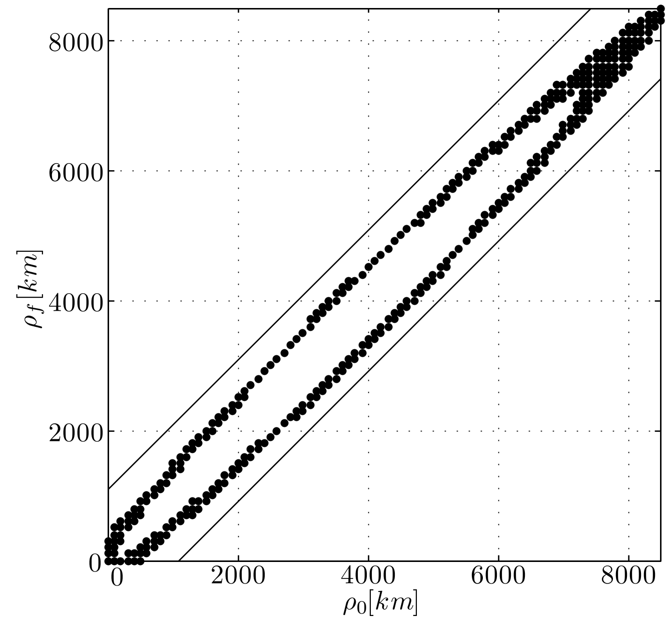


Figure 4.3: Initialization chromosomes

$max(v_{T_{perigee}}) = 10.21km/s$ yielding to $D = 1063.11km$ for $(t_f - t_0) = 60s$. The example does not consider conditions of illumination or visibility.

The points that fulfill these two conditions are shown in Fig. 4.3, such black points represent the initialization of the genetic algorithm; the shape and the extension of this figure depends on the position of the observer and on the initial and final measurements. In Fig. 4.4 the corresponding a and e are shown, these parameters represent the shape of the orbit. The corresponding i and Ω are shown as well, representing the orbital plane.

Since the algorithm runs on a fixed population, the number of the initial chromosomes could be greater or lower than the previously fixed number. In the first case, a random selection is executed until the desired number is reached, while in the second case a mutation operator will be executed to increase the population.

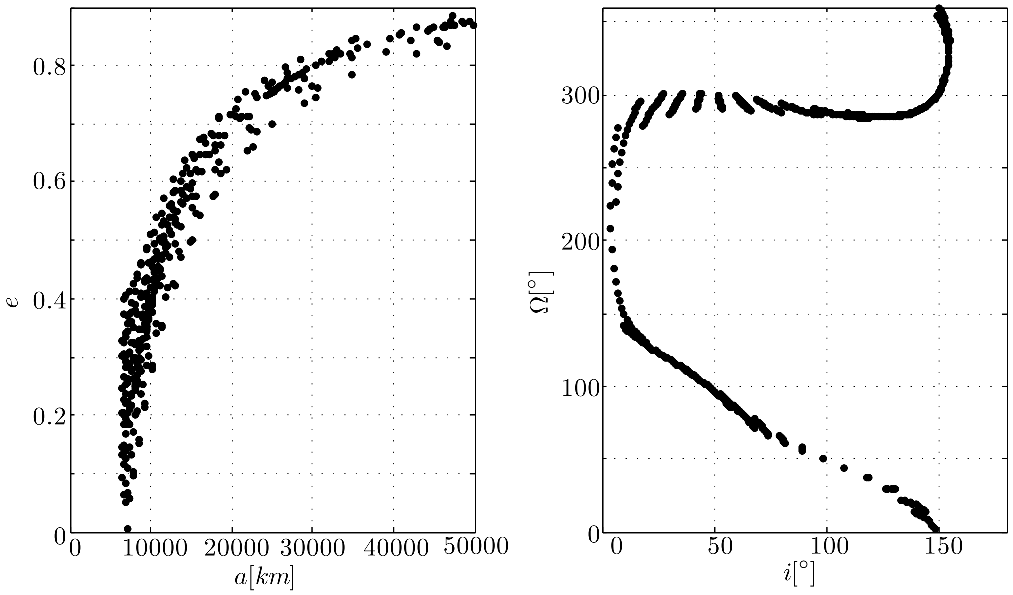


Figure 4.4: a-e and i-Ω for the initial chromosomes

4.3.2 Mutation

The mutation is a genetic operation that is used at the beginning of the algorithm and at each generation. The function takes the initial population as input, together with the number of the desired chromosomes for the output. The function needs to know the variance to apply at the randomly chosen chromosomes to spread the offspring; this could be an input or just a fixed parameter. The mutation function takes a random chromosome for each desired output, applies the variance and yields the mutated chromosome. The standard deviation used in our example case is 10 km.

$$\begin{pmatrix} \rho_0 \\ \rho_f \end{pmatrix} \rightarrow \begin{pmatrix} \rho_0 + \sigma_{\rho_0} \\ \rho_f + \sigma_{\rho_f} \end{pmatrix} \tag{4.13}$$

4.3.3 Crossover

The crossover is a genetic operation that is used in each generation. It simply takes two random chromosomes and chooses one value from the first parent and the second from the other parent. Since the chromosomes have only two genes, it is a very simple single-point crossover.

$$\begin{pmatrix} \rho_{0_1} \\ \rho_{f_1} \end{pmatrix} \begin{pmatrix} \rho_{0_2} \\ \rho_{f_2} \end{pmatrix} \rightarrow \begin{pmatrix} \rho_{0_1} \\ \rho_{f_2} \end{pmatrix} or \begin{pmatrix} \rho_{0_2} \\ \rho_{f_1} \end{pmatrix} \tag{4.14}$$

The new chromosomes must respect the maximum possible difference, such that: $|\rho_{0_1} - \rho_{f_2}| < D$; if the difference exceeds that maximum value, the new chromosome is discarded and a new one will be generated.

4.3.4 Fitness function

The fitness function is the most important part of the entire genetic algorithm. This function takes into account all the intermediate observations. The fitness value is defined as:

$$fit = \prod_{i=2}^{n_{obs}-1} \left(\hat{L}_{meas}(t_i) \cdot \hat{L}^*(t_i)\right) \tag{4.15}$$

where $\hat{L}_{meas}(t_i)$ is the actual measurement, and $\hat{L}^*(t_i)$ is the fictitious measurement computed propagating the orbit from the initial state vector obtained with the Lambert's solver, n_{obs} is the total number of observations. In our example we consider a period of 60 seconds with measurements every second starting at $t = 0$, so $n_{obs} = 61$. An error with a standard deviation of 1 arcsecond for each angular measurement has been applied. The propagation is based on the orbit found with ρ_0 and ρ_f through the Lambert's problem solution, $\vec{r}_{T_0}$ and $\vec{v}_{T_0}$ are obtained and propagated using two-body motion. The short time of propagation allows the use of a simple keplerian propagator, because the effects of perturbations are very negligible.

This fitness function represents the multiplication of the dot products of actual and simulated measurements at each time. The first and the last product obviously provide no data because the dot products are equal to 1.The maximum value of the fitness function is 1 and gives the exact solution for the target orbit.

One disadvantage is that the function works with values that are very close to 1 as the solution near. For example, if the angle between the true and the reconstructed measurements is 3 arcseconds for each observation, with 61 observations, the fitness value differs from 1 of a value of the order of 10^{-9}.

4.3.5 Selection and completion of a new generation

The selection is just a cut-off of the chromosomes, the best 10% of the initial population is selected to create the new generation, this corresponds to an elitist selection of n_{elite} values, in our example $n_{elite} = 100$. These chromosomes will be ordered in descending order giving for each chromosome an ascending value k, so that $k = 1$ corresponds to the best fitness value. The 80% of the new generation will be obtained through the mutation operator applied to the selected population. Another 5% is obtained through the crossover operator and the remaining 5% with a new set of chromosomes to ensure diversity.

4.3.6 Stop condition

Once a new elite of chromosomes is generated, some simple quantities are computed and stored: the best values (ρ_0^*, ρ_f^*), corresponding to the highest fitness value, the arithmetic mean values $(\overline{\rho}_0, \overline{\rho}_f)$ and the covariance matrix, so it is possible to obtain the standard deviations $(\sigma_{\rho_0}, \sigma_{\rho_f}, \sigma_{\rho_0\rho_f})$. The arithmetic mean values are simply obtained through:

$$(\overline{\rho}_0, \overline{\rho}_f) = \sum_{k=1}^{n_{elite}} \frac{(\rho_{0_k}, \rho_{f_k})}{n_{elite}} \tag{4.16}$$

while the covariance matrix C is simply:

$$C = \begin{pmatrix} \frac{\sum_{k=1}^{n_{elite}} (\rho_{0_k} - \overline{\rho}_0)^2}{n_{elite}} & \frac{\sum_{k=1}^{n_{elite}} (\rho_{0_k} - \overline{\rho}_0)(\rho_{f_k} - \overline{\rho}_f)}{n_{elite}} \\ \frac{\sum_{k=1}^{n_{elite}} (\rho_{0_k} - \overline{\rho}_0)(\rho_{f_k} - \overline{\rho}_f)}{n_{elite}} & \frac{\sum_{k=1}^{n_{elite}} (\rho_{f_k} - \overline{\rho}_f)^2}{n_{elite}} \end{pmatrix} \tag{4.17}$$

The genetic algorithm stops if one of the followings statements is true:

- number of generations bigger than the parameter max_{gen}
- $(\sigma_{\rho_0}, \sigma_{\rho_f}) < (\sigma_{lim_{\rho_0}}, \sigma_{lim_{\rho_f}})$, where $(\sigma_{lim_{\rho_0}}, \sigma_{lim_{\rho_f}})$ are parameters
- $(\overline{\rho}_0, \overline{\rho}_f)$ or (ρ_0^*, ρ_f^*) not varying for ψ generations, where ψ is a parameter

In our test case: $max_{gen} = 30$, $\sigma_{lim_{\rho_0}} = \sigma_{lim_{\rho_f}} = 5km$ and $\psi = 3$. Instead of showing the values of $(\overline{\rho}_0, \overline{\rho}_f)$ and (ρ_0^*, ρ_f^*) in Fig. 4.5 the errors of the mean and best solution are shown. Since each solution is represented in a plane, the error is simply obtained through the Cartesian distance from the true value. In this case, the algorithm is stopped because the best values did not change for ψ generations.

In the following figures, the points of the last elite are presented. It is important to note that the values of ρ_0 and ρ_f in the left chart of the Fig. 4.6 are very close to their linear regression. The linear regression is computed as:

$$\begin{cases} y = ax + b \\ a = \frac{\sigma_{\rho_0\rho_f}^2}{\sigma_{\rho_0}^2} \\ b = \overline{\rho}_f^{\,2} - a\overline{\rho}_0 \end{cases} \tag{4.18}$$

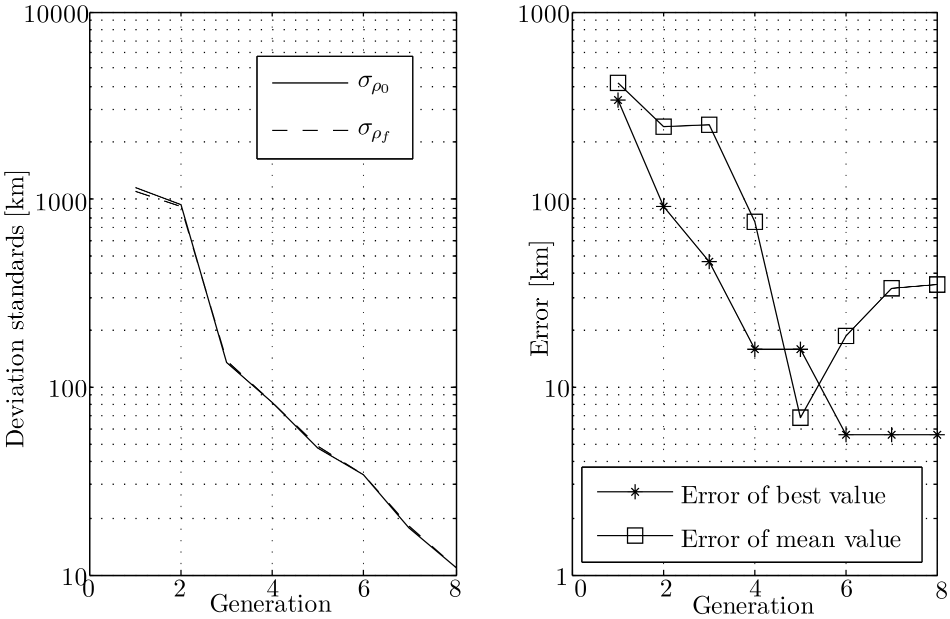


Figure 4.5: σ_{ρ_0}, σ_{ρ_f} and errors of best and mean values

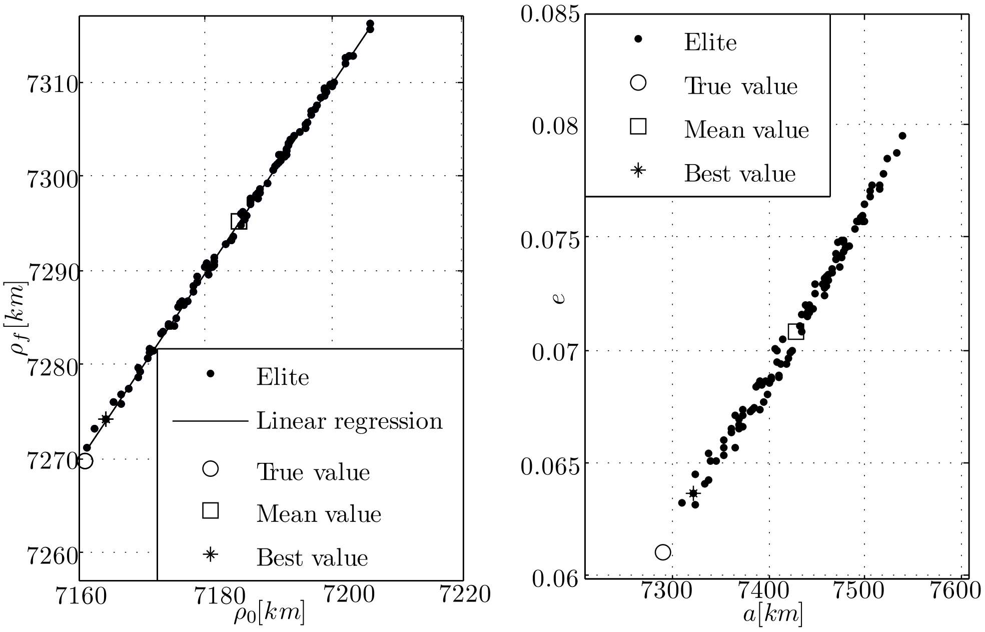


Figure 4.6: ρ_0, ρ_f and a-e for the elite

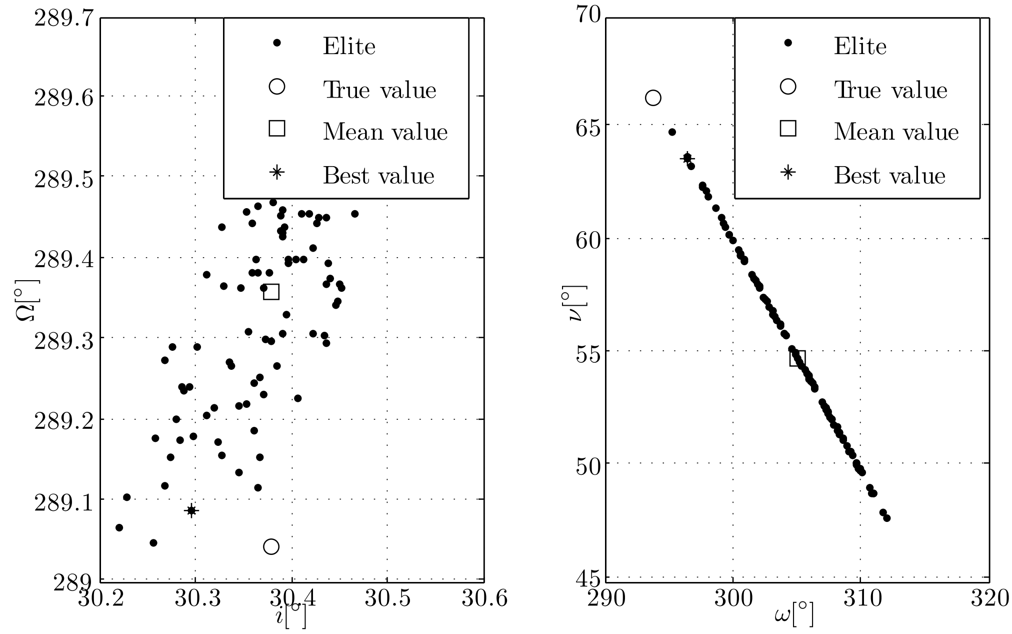


Figure 4.7: i-Ω and ω-ν for the elite

To obtain a correlation between ρ_0 and ρ_f, the Pearson correlation coefficient was computed, see [65]. This coefficient, usually denoted by r is computed as:

$$r = \frac{\sigma^2_{\rho_0 \rho_f}}{\sigma_{\rho_0}\sigma_{\rho_f}} \tag{4.19}$$

r is very close to 1 in each generation of the algorithm, as shown in Fig. 4.8. This means there is a linear dependence between ρ_0 and ρ_f. The value of r remains higher than 99.9% just from the third generation; this justifies the very low percentage used for the crossover operator.

Table 4.2 shows the values of σ_{ρ_0}, σ_{ρ_f} and $\sigma_{\rho_0\rho_f}$. These results demonstrate that the covariance matrix is nearly similar to ξI_{2x2} where ξ is a constant that decreases for each generation, that means that there is a linear dependence between the variables with a slope very close to 1.

Another interesting consideration regards the right chart in Fig. 4.7. There is a very linear dependence between ω and ν of the elite chromosomes. In fact, the Pearson correlation coefficient is equal to -1 and the slope of the linear regression is -1.02, that means that the sum of ω and ν is almost constant. The error is very large in the determination of these angles because the orbit has an eccentricity near to 0. With a near circular orbit, the argument of perigee is poorly defined. A better angle is the argument of latitude defined

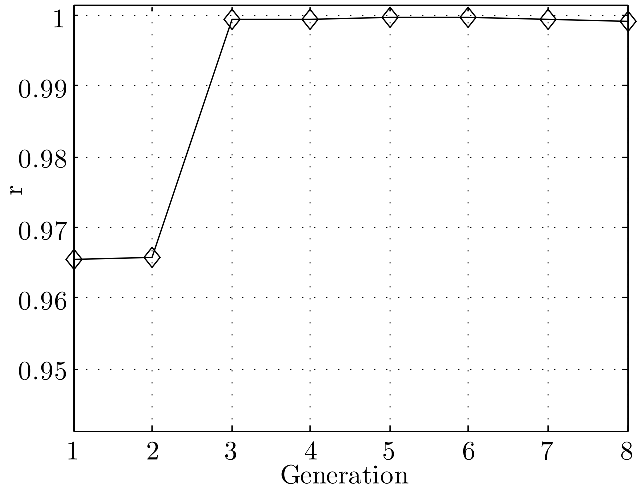


Figure 4.8: The Pearson correlation coefficient

Generation	σ_{ρ_0}	σ_{ρ_f}	$\sigma_{\rho_0 \rho_f}$
1	1158.67 km	1112.00 km	1115.33 km
2	945.06 km	905.03 km	908.90 km
3	135.34 km	137.98 km	136.62 km
4	82.06 km	82.01 km	82.51 km
5	47.64 km	48.32 km	47.97 km
6	33.79 km	34.29 km	34.03 km
7	17.80 km	18.00 km	17.89 km
8	10.78 km	10.89 km	10.83 km

Table 4.2: The standard deviations of the elite chromosomes

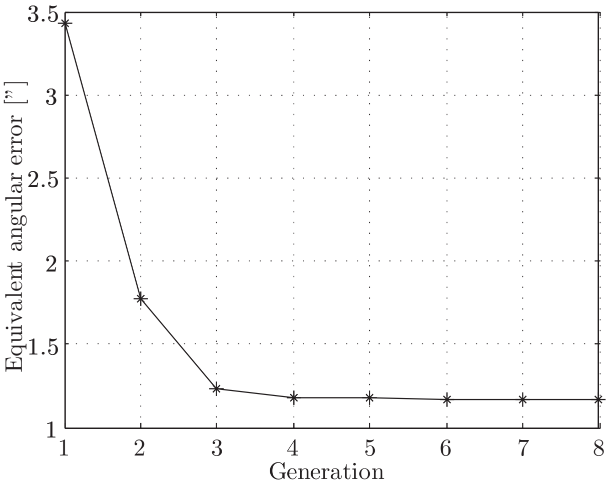


Figure 4.9: The Equivalent Angular Error

Distances	True Value	Estimated Value	Error
ρ_0	7161.20 km	7164.62 km	3.42 km
ρ_f	7269.76 km	7274.20 km	4.44 km

Table 4.3: Results of the distances estimation

as:

$$u = \omega + \nu \tag{4.20}$$

In conclusion, while there is a large error for the argument of perigee and the true anomaly, the argument of latitude (u) is well determined. In this example case u is very near to 360°, meaning that the target is passing the nodal axis. To better interpret the results from a geometrical point of view, it could be useful to introduce the EAE; this angle is defined like the average angle between the actual and the reconstructed measurements using the best chromosome for each generation, the mathematical expression is in Eq. (4.21); while the results are shown in Fig. 4.9.

$$EAE = \arccos\left((fit_{k=1})^{\frac{1}{n_{obs}-2}}\right) \tag{4.21}$$

The values obtained for ρ_0 and ρ_f are shown in Table 4.3 with the corresponding errors. The values in Table 4.4 show that there is very good determination of the orbital plane and the argument of latitude, while the biggest part of the error in ρ_0 and ρ_f affects the semi-major axis and the eccentricity.

Orbital Parameter	True Value	Estimated Value	Error	Percentage Error
a	7290.20 km	7321.30 km	31.1 km	0.43%
e	0.0610	0.0636	0.0026	4.26%
i	30.379°	30.295°	0.084°	0.28%
Ω	289.042°	289.086°	0.044°	0.02%
ω	293.776°	296.435°	2.659°	0.91%
ν	66.230°	63.541°	2.689°	4.06%
u	360.006°	359.976°	0.030°	0.01%

Table 4.4: Results of the orbital determination

Orbital Parameter	σ_{meas}=1"	σ_{meas}=2"	σ_{meas}=3"
a	31.1 km	64.38 km	102.96 km
e	0.0026	0.0057	0.0091
i	0.084°	0.180°	0.262
Ω	0.044°	0.088°	0.149
ω	2.659°	5.213°	7.949
ν	2.689°	5.273°	8.050
u	0.030°	0.061°	0.101

Table 4.5: Errors depending on the measurements

There is an important issue regarding the accuracy of the measurements; in Table 4.5 the errors of the estimated values are shown depending on the σ of the measurements. It is worthwhile noting that there is a linear dependence with the accuracy of the measurements, emphasizing the fact that a very accurate sensor is needed.

4.4 Monte Carlo Simulations

The algorithm was tested using Monte Carlo (MC) simulations. The seed of the random noise model applied to the measurements was changed for each case, showing the reliability of the algorithm. Convergence is assured, yielding a result for each case; the results presented regard a full simulation with 1000 runs. In the left part of Fig. 4.10 the EAE is shown for a sensor accuracy of 1". The mean value is approximately 1.1", while the maximum is under 1.5". Each point of the Monte Carlo results represents the best value for each case. The true values are located in the center of the results' distribution, as expected. The Monte Carlo results have roughly the same characteristics of the elite points presented in our example case of Section 4.3. We can see

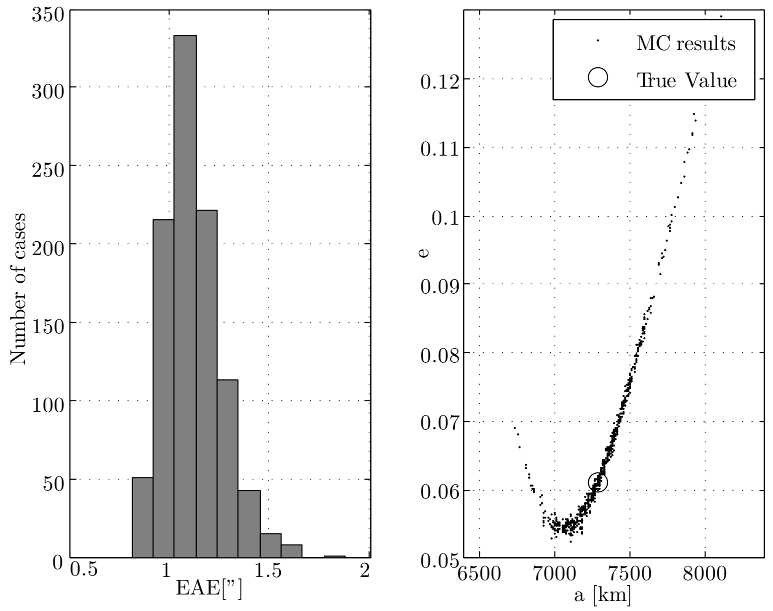


Figure 4.10: The EAE and a-e of the MC results with accuracy of 1"

Orbital Parameter	True Value	Mean Value	Standard Deviation
a	7290.20 km	7310.77 km	163.8 km
e	0.0610	0.0644	0.0089
i	30.379°	30.396°	0.149°
Ω	289.042°	289.083°	0.422°
ω	293.776°	293.418°	15.731°
ν	66.230°	66.560°	16.012°
u	360.006°	359.978°	0.285°

Table 4.6: Results of the Monte Carlo simulatio with accuracy of 1"

that the largest part of the error is spread over the semi-major axis and the eccentricity, as expected. The orbital plane is usually well determined, with a very low error for the inclination. As usual, for an almost circular orbit, the argument of perigee and the true anomaly have a large error, but we can see the linear dependence with a slope of -1, meaning that the argument of latitude is very well determined.

The results are summarized in Table 4.6, showing the true values, the mean values and the standard deviation for each parameter.

In Fig. 4.12 and Fig. 4.13 the results for the sensor accuracy of 2" are shown. The results' distributions show similar behavior to the case with accuracy of 1", but the standard deviations of the errors are greater.

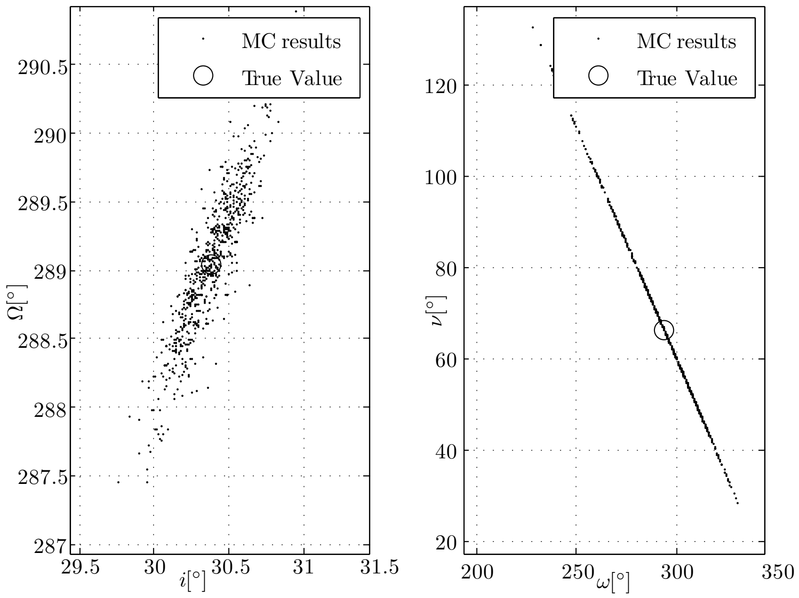


Figure 4.11: i-Ω and ω-ν of the MC with accuracy of 1"

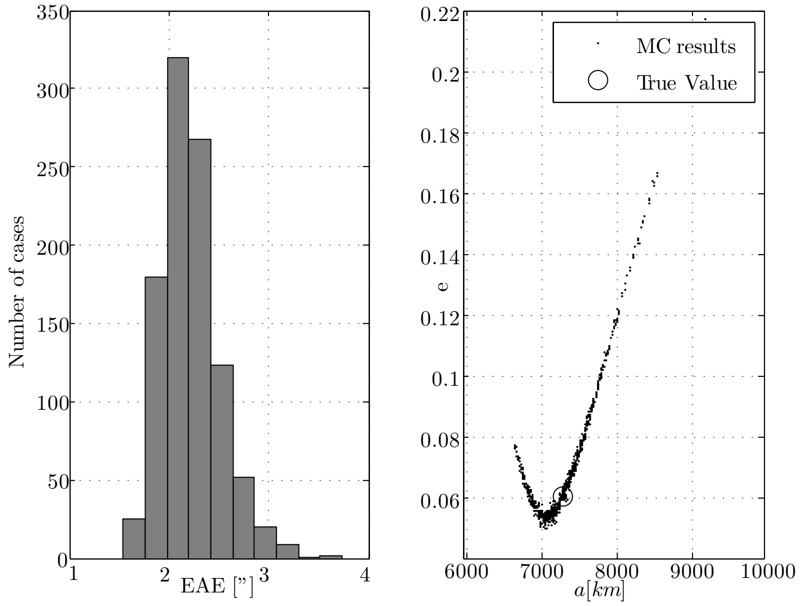


Figure 4.12: The EAE and a-e of the MC results with accuracy of 2"

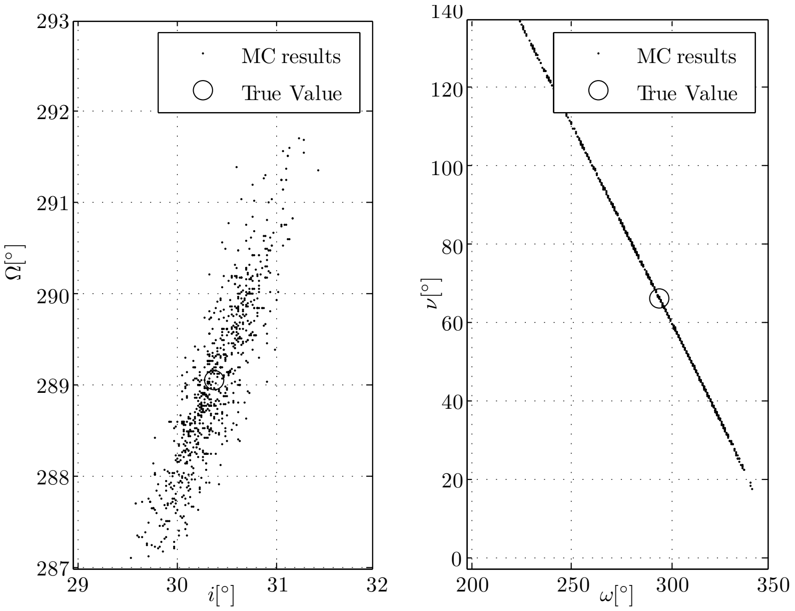


Figure 4.13: i-Ω and ω-ν of the MC results with accuracy of 2”

Fig. 4.14 and Fig. 4.15 show the results for the sensor accuracy of 3”. Also in this case the results’ distributions are similar to the previous cases, showing an increase of the standard deviations of the errors.

From the above tests it is clear that the more the sensor accuracy decreases the more the orbit determination get worse. As result, it is obvious that a high accuracy sensor is needed for very short arc observations.

4.5 Comparison with Classical Methods

In this section we compare the results of the proposed algorithm with the classical orbit determination methods. Laplace, Gauss and Double-r algorithms are applied to the same obsever-target scenario of Fig. 4.1. For the numerical test the observer orbit and the target orbit have the same parameters as in Table 4.1 and Table 4.4. Moreover, the duration of the observations is set to 60 seconds. Because these methos use three measurements only, the input data are at t_0, $(t_f - t_0)/2$ and t_f.The numerical results are shown in Table 4.7.

The classical methods give estimation values with great errors. In particular Laplace and Gauss methods show the convergence to the observer orbit instead of estimating the target orbit, while the Double-r algorithm converges to a unrealistic orbit (the semi-major axis is smaller than the Earth radius). As

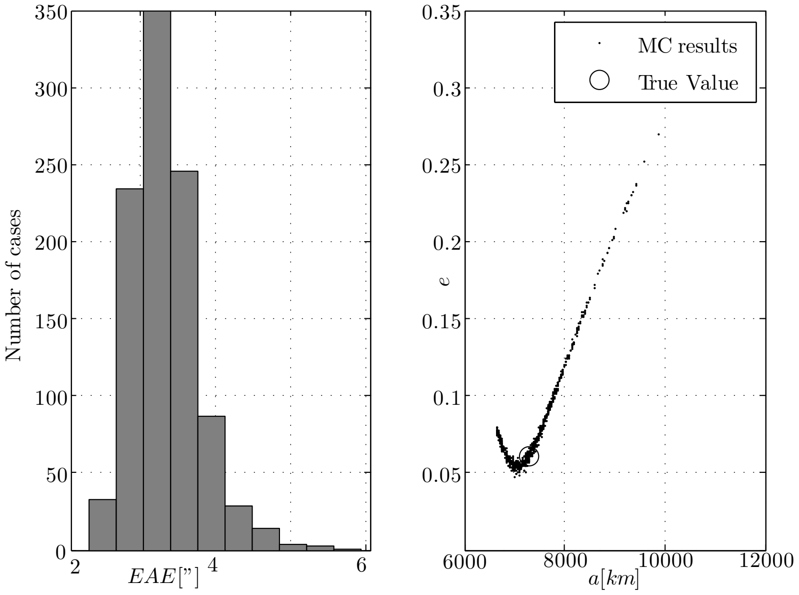


Figure 4.14: The EAE and a-e of the MC results with accuracy of 3"

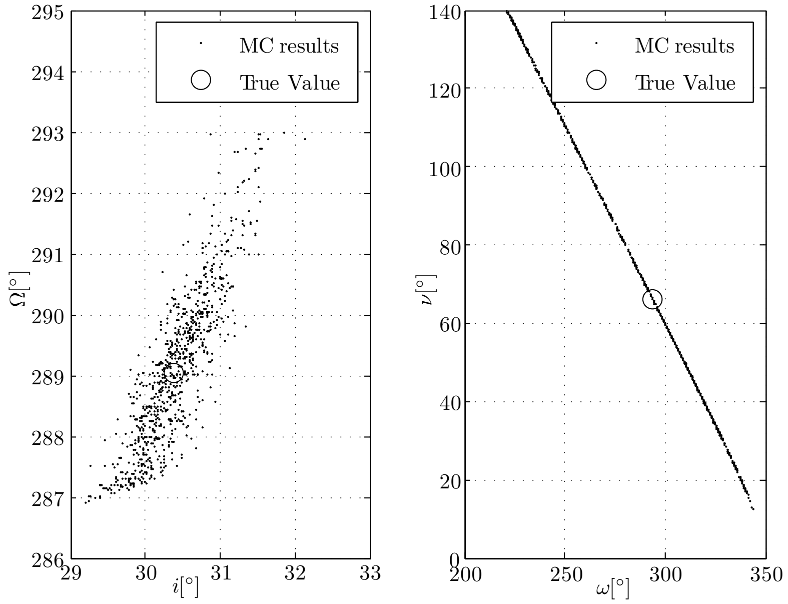


Figure 4.15: i-Ω and ω-ν of the MC results with accuracy of 3"

Orbital Parameter	True Value	Laplace	Gauss	Double-r
a	7290.20 km	7133.38 km	7124.76 km	4384.93 km
e	0.0610	0.0044	0.0053	0.4074
i	30.379°	98.288°	98.291°	147.088 °
Ω	289.042°	49.864°	49.881°	295.636 °
ω	293.776°	356.275°	350.389°	208.720°
ν	66.230°	151.792°	157.682°	185.235°
u	360.006°	148.067°	148.071°	33.955°

Table 4.7: Results of the orbital determination with classic methods

expected, the classical methods are not able to face with the problem of the orbit determination using a single TSA.

4.6 The coplanar case

Classical orbit determination methods show singularities for the coplanar case, see [20]. In order to test our algorithm for the coplanar case, Monte Carlo simulations are performed with accuracy of 1”. Fig. 4.16 and Fig. 4.17 show the results.

In the tests no singularities are encountered, but there is a huge dispersion of the determined values in the semi-major axis and the eccentricity. On the other hand the orbital plane and the argument of latitude ($\omega + \nu$) are well determined, see Fig. 4.17.

4.7 Conclusions

A new genetic algorithm for Initial Orbit Determination was developed to deal with very short passes. The performed study was focused on space-based observer in LEO and in the cases of no-coplanar and coplanar observed orbiting object. The study was conducted to test the algorithm for short arc optical observations of duration of just 60 seconds. The algorithm was tested with 1000 runs of Monte Carlo simulations in which each simulation was performed with different seed in the random noise model. Moreover, the Monte Carlo analysis was performed for sensors with accuracies of 1”, 2” and 3”. The results show good performances for such short period of observation time. The coplanar case shows no singularities, but the initial orbital determination is less accurate with respect to the no-coplanar case.

The running time is of the order of a minute on a laptop with a dual-core

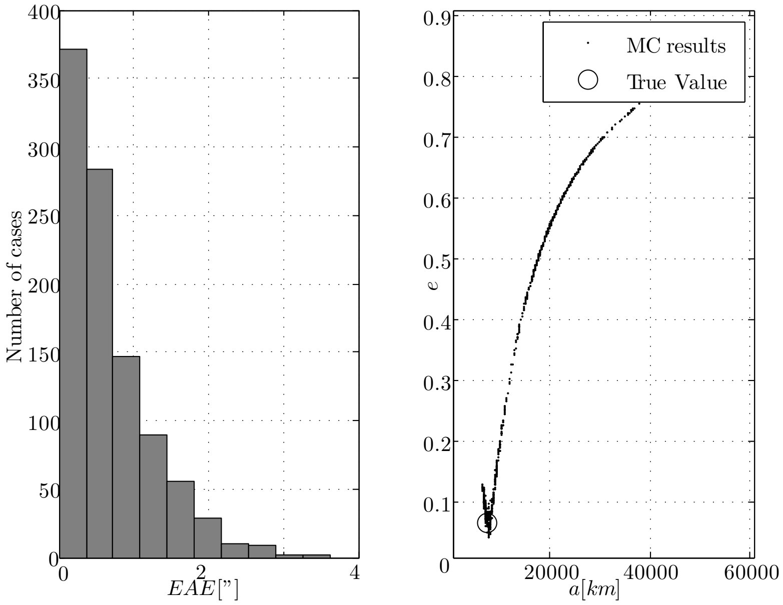


Figure 4.16: The EAE and a-e of the MC results with accuracy of 1" for the coplanar case

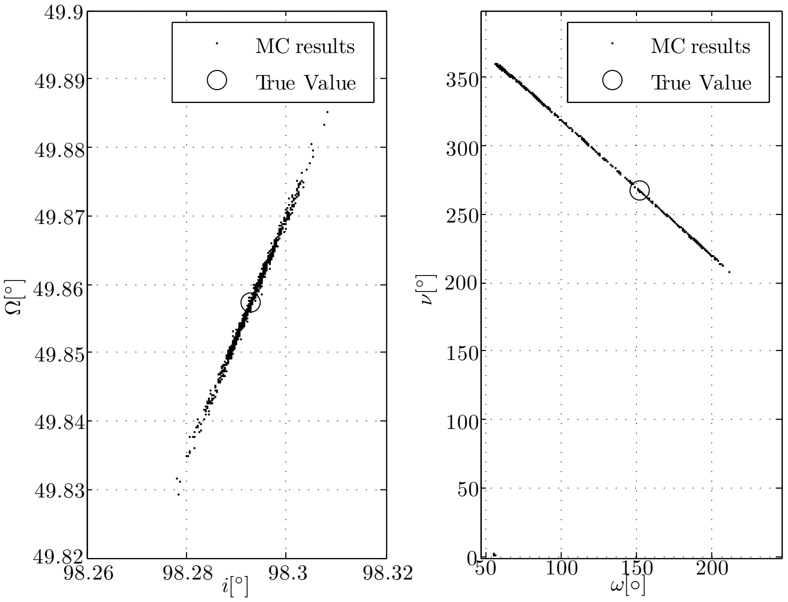


Figure 4.17: i-Ω and ω-ν of the MC results with accuracy of 1" for the coplanar case

processor and 2 GB of Random Access Memory (RAM), even if, the software has not yet been optimized for the execution speed.

As a future work, the algorithm will be applied to observations acquired by a ground-based telescope, in order to obtain results from a real case. Another future improvement will be the use of a statistical approach in order to correlate the observed object to the satellite catalog provided by NORAD. The aim will be to identify the orbiting object in order to improve the orbit determination accuracy.

Chapter 5

Performance Assessment

Those that are most slow in making a promise are the most faithful in the performance of it.

- Attributed to Jean-Jacques Rousseau

In this chapter the results of the genetic algorithm wil be analyzed varying some key parameters, such the observation time or noise seed. Each parameter is changed keeping all the other ones fixed to highlight the influence of that parameter. This analysis also shows the reliability of the algorithm.

5.1 Introduction

One of the main concerns about genetic algorithms is about the performance assessment. In this chapter several test will be reported maintaining the same scenario. First of all, the results of the explained procedure for the search algorithm in two dimensions will be used in a six dimensions space. The number of the observations are not so important for the results of the orbit determination process. The same scenario will be presented using only three observations for one minute of observation time.

The initialization of the genetic algorithm will be deeply analyzed considering a modified grid for the initialization of the search for the initial and the final distance.

The most important key factor to obtain better results in orbit determination is the observation period. Results for different periods will be shown and analyzed.

The last part of this chapter will be about the sensibility to the random noise. Different seeds will be applied to test the stability of the algorithm and to see how the results are affected.

5.2 Varying the number of the observations

The results of the algorithm do not change so much varying the number of the observations. It is deducible that increasing the number of the observations for such short period of time does not decrease considerably the error of the results.

The observation with *dense data*, see [78], does not produce results very different from an observation with just three measurements. The next results are obtained with the same procedure of the past chapter, but with just three measurements at 0, 30 and 60 seconds.

The standard deviations of the elite for each generation and the error for the best and the mean value of the elite are shown in Fig. 5.1.

Notice that the standard deviations for ρ_0 and ρ_f of the elite reach a good convergence just after 10 generations, both maintaining a value under 10 km.

The final error of the best result is about 10 km, that is near to the value of 6 km found in Fig. 4.5.

The final values of the elite in the left graph of Fig. 5.2 show that the best chromosomes of ρ_0 and ρ_f always spread out in a linear dependence, see the values of r in Fig. 5.4 as well.

The final results of the orbit determination are shown in Table 5.1. Notice that these values are very similar to the ones in Table 4.5

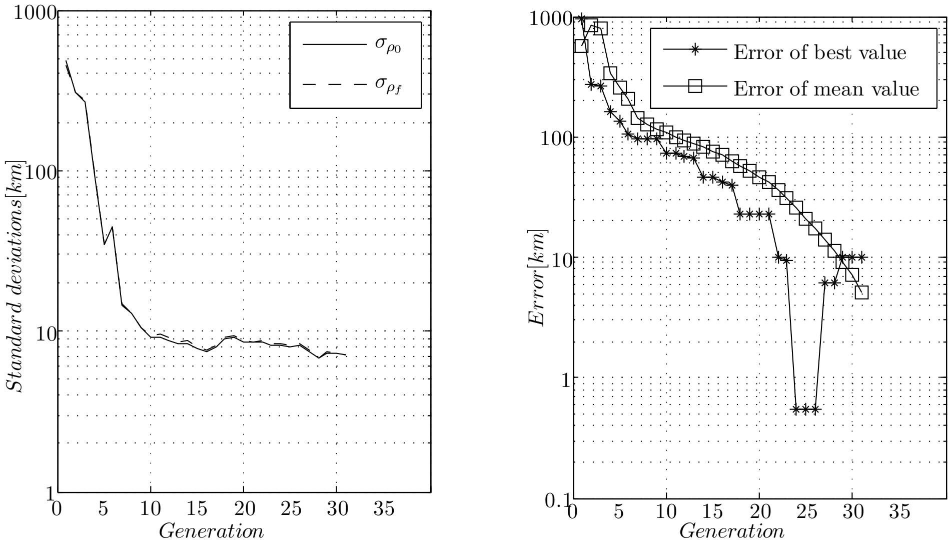


Figure 5.1: σ_{ρ_0}, σ_{ρ_f} and errors of best and mean values, $n_{obs} = 3$

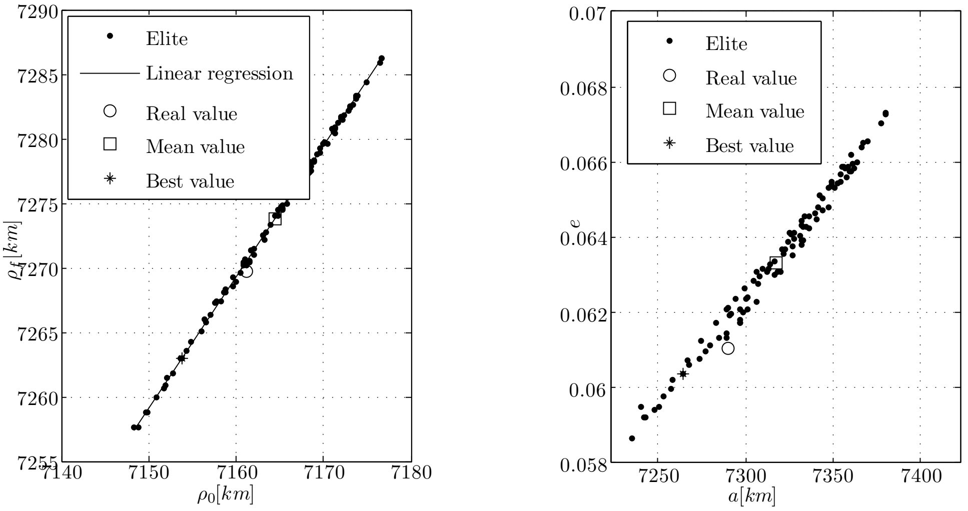


Figure 5.2: ρ_0, ρ_f and a-e for the elite, $n_{obs} = 3$

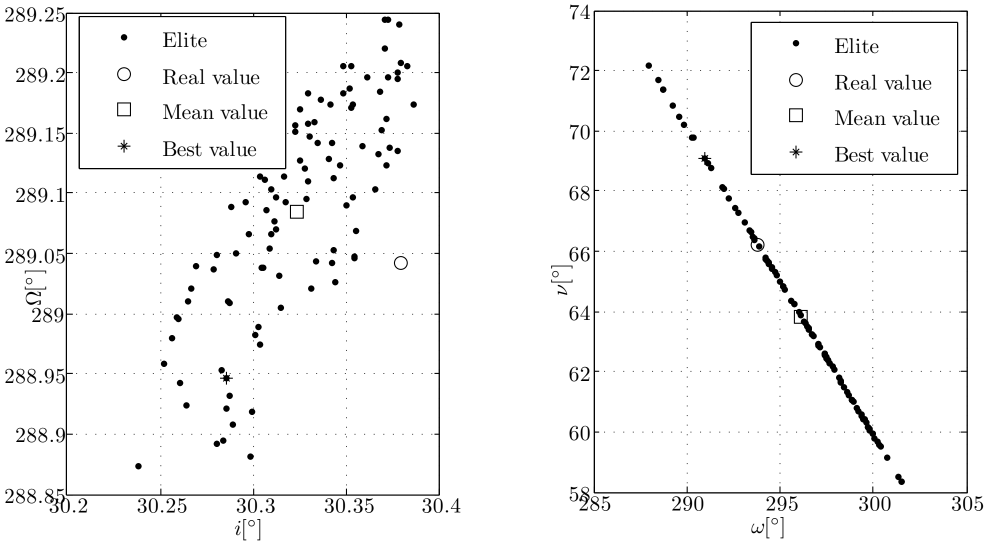


Figure 5.3: i-Ω **and** ω-ν **for the elite,** $n_{obs} = 3$

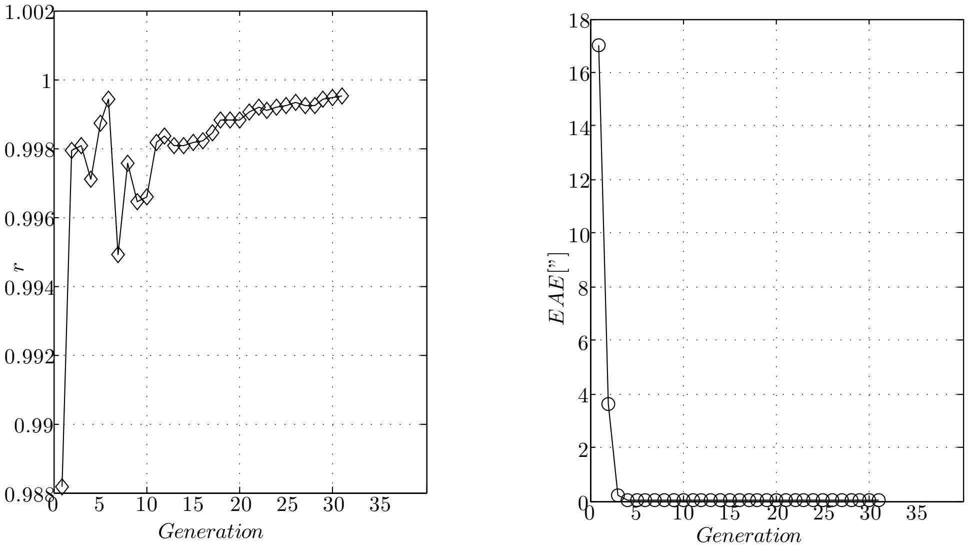


Figure 5.4: r **and** EAE, $n_{obs} = 3$

Orbital Parameter	True Value	Estimated Value	Error	Percentage Error
a	7290.20 km	7264.15 km	26.05 km	0.36%
e	0.0610	0.0604	0.0006	0.98%
i	30.379°	30.290°	0.089°	0.29%
Ω	289.042°	288.903°	0.139°	0.05%
ω	293.776°	291.521°	2.255°	0.77%
ν	66.230°	69.091°	2.861°	4.32%
u	360.006°	360.612°	0.606°	0.17%

Table 5.1: Results of the orbital determination, $n_{obs} = 3$

5.3 Varying the initialization

The initialization of the algorithm in two dimensions is made through an equispaced grid in the plane identified by ρ_0 and ρ_f. Each point has to satisfy the following equations in Eq. (5.1).

$$\begin{cases} \rho_f < \rho_0 + D \\ \rho_f > \rho_0 - D \\ R_\oplus + 200km \leq a \leq a_{MAX} = 50000km \\ 0 \leq e \leq e_{MAX} = \left(1 - \dfrac{r_{O_{perigee}}}{a_{MAX}}\right) \end{cases} \tag{5.1}$$

with $R_\oplus = 6371km$, $D = 1063.11km$ and $e_{MAX} = 0.86$.

The results with different grid spacing are presented in Table 5.2.

Grid Spacing	Initial EAE	Final EAE	Time for initialization	Total time
$200km$	17”	0.04”	4.6s	79.2s
$100km$	3.5”	0.04”	13.9s	90.7s
$50km$	7.4”	0.01”	55.1s	86.2s
$25km$	9.3”	0.05”	207.7s	232.7s
$12.5km$	2.6”	0.01”	854.9s	881.2s

Table 5.2: EAE **and execution times for different grid spacing**

The initialization is really time consuming, so we want a good matching between time and performances. The best grid seems to be the one with the spacing of 50 km. The initial EAE does not vary in a intuitive way because the high number of points for initialization often exceeds the fixed number for the genetic algorithm that runs with a constant number of individuals.

If the number of the initial points exceeds the fixed number of the genetic algorithm, a random choice is executed to pick up only the pre-fixed number of desired individuals. No evaluation on the fitness function is still performed, so it is possible that some of the best individuals are discarded because of the random choice between the initial individuals.

The initialization with a grid spacing of 200 km is presented in Fig. 5.5 and Fig. 5.6.

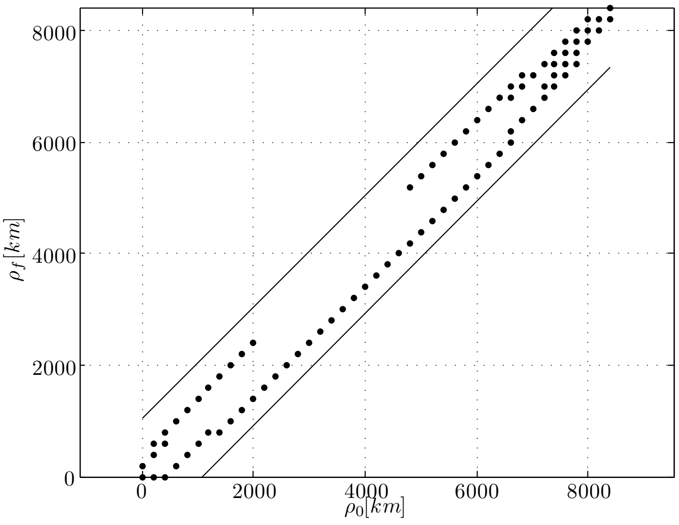


Figure 5.5: Initialization chromosomes, grid spacing = 200km

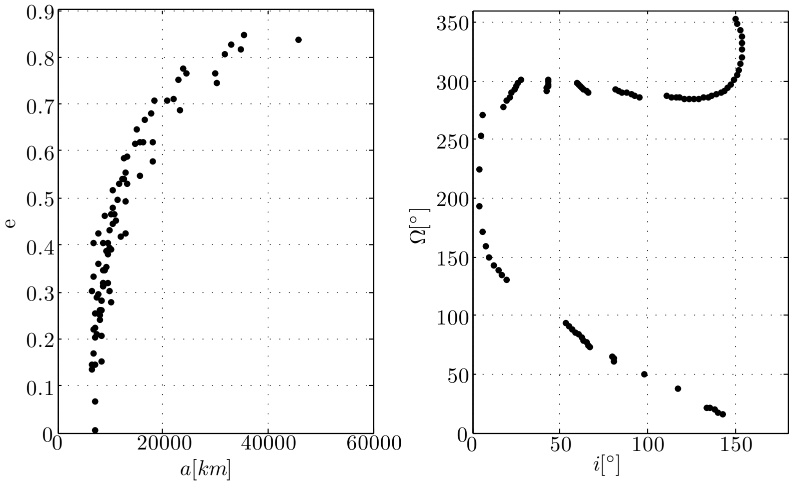


Figure 5.6: a-e and i-Ω for the initial chromosomes, grid spacing = 200km

The initialization with a grid spacing of 12.5 km is presented in Fig. 5.7 and Fig. 5.8.

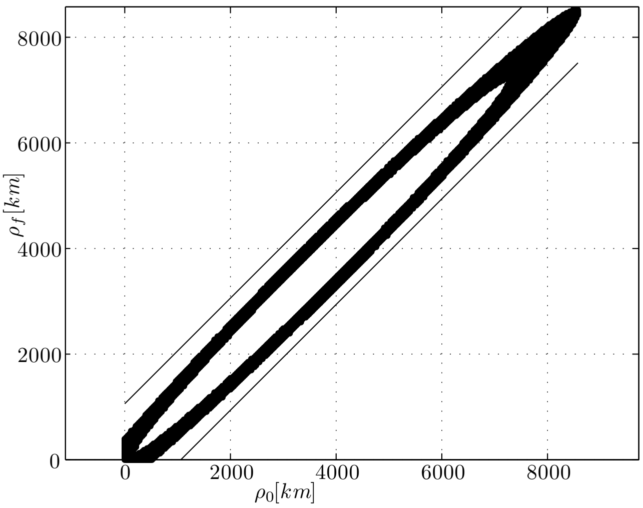


Figure 5.7: Initialization chromosomes, grid spacing = 12.5km

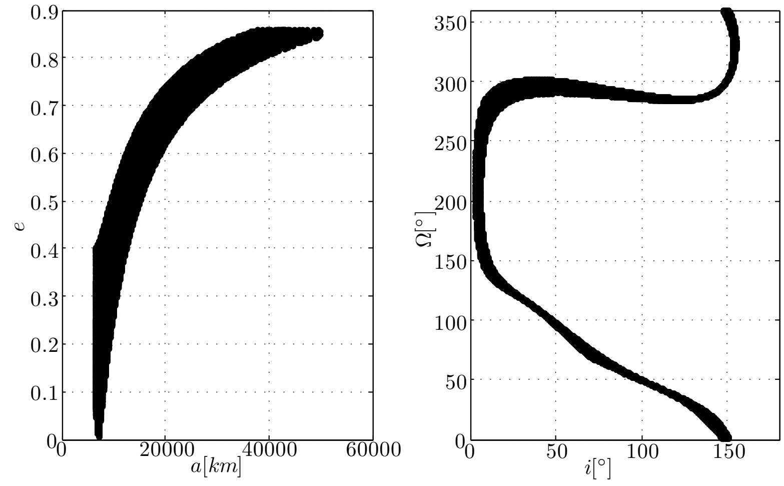


Figure 5.8: a-e and i-Ω for the initial chromosomes, grid spacing = 12.5km

The shape that the points create are perfectly the same, obviously the number of the points increases if the grid spacing decreases. Notice that the initialization in Fig. 4.3 and Fig. 4.4 are referring to a grid with spacing of 100 km.

5.4 Varying the observation time

The observation time is the fundamental parameter that the orbit determination error is dependent of. A longer observation time allows a better estimation, as with the classical orbit determination algorithms.

The presented test have been conducted using $n_{obs} = 3$ but increasing and decreasing the observation time. Under 15 s the estimation is no more significant.

Observation Period	Error of a	Error of e	Error of i	Error of Ω
15s	$240.7km$	0.0038	1.39°	1.04°
30s	$77.3km$	0.0009	0.36°	0.30°
60s	$26.1km$	0.0006	0.09°	0.14°
120s	$7.3km$	0.0001	0.02°	0.02°
240s	$0.2km$	0.00003	0.001°	0.001°

Table 5.3: Results of the orbital determination with several observation periods

In Table 5.3 the results for observation times between 15 s and 240 s are shown. Notice that the error of the estimation is decreasing for each of the shown orbital elements presented.

Only the orbital elements regarding the orbit shape and the orbit plane are shown for sake of simplicity. The error of the measurements has a standard deviation of 1". A longer observation time does not seem necessary because of the very good results obtained with periods shorter than 240s.

The results for the elite of the last generation are shown in Fig. 5.9 for an observation of 15 s. Notice the dispersion of the points and that some results approach very low level of semimajor axis. It seems that the algorithm has found two different minima because the elite are not concentred around a point.

The results for an observation of 240 s are shown in Fig. 5.10. Notice that the estimated values are really close to the real ones.

The mean values of the last elite are positioned very close to the real values as well. This means that, probably, further iterations could bring better results. The convergence for each setup is very good showing a EAE always in the order of 0.01".

An observation time of 240 s is still feasible also with a ground based observer and a good pass of a LEO satellite.

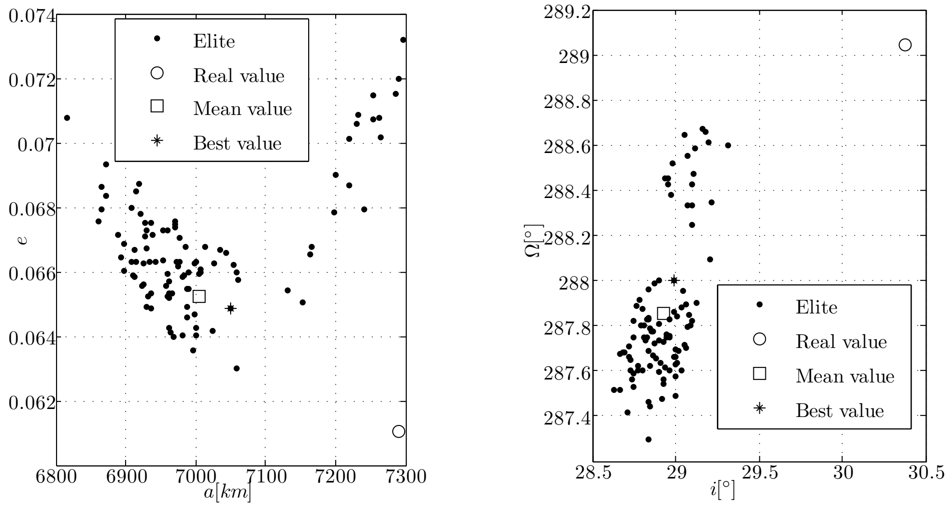


Figure 5.9: Final elite for the observation time of 15s

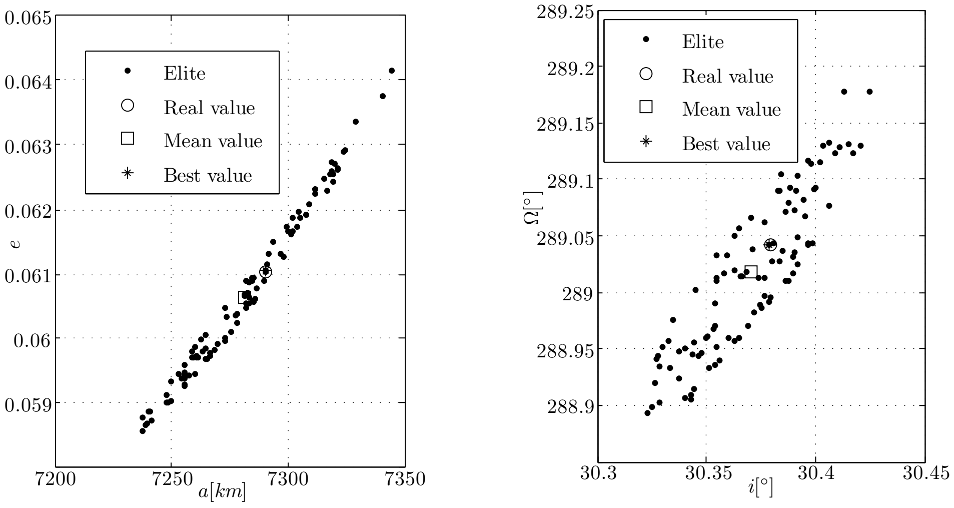


Figure 5.10: Final elite for the observation time of 240s

5.5 Varying the noise seed

The noise has a fundamental role in a stochastic algorithm, an in-depth analysis is in Appendix A. A good algorithm should be independent on the way the random noise is generated.

To test the reliability of the algorithm, we run the entire process with different noise seeds, maitaining the initial measurements. This method will highlight the strong dependence that exists between noise and stochastic algorithms.

In Fig. 5.11 and Fig. 5.12 the results of the best value for each run are depicted with stars, while the true value still remains depicted with a circle.

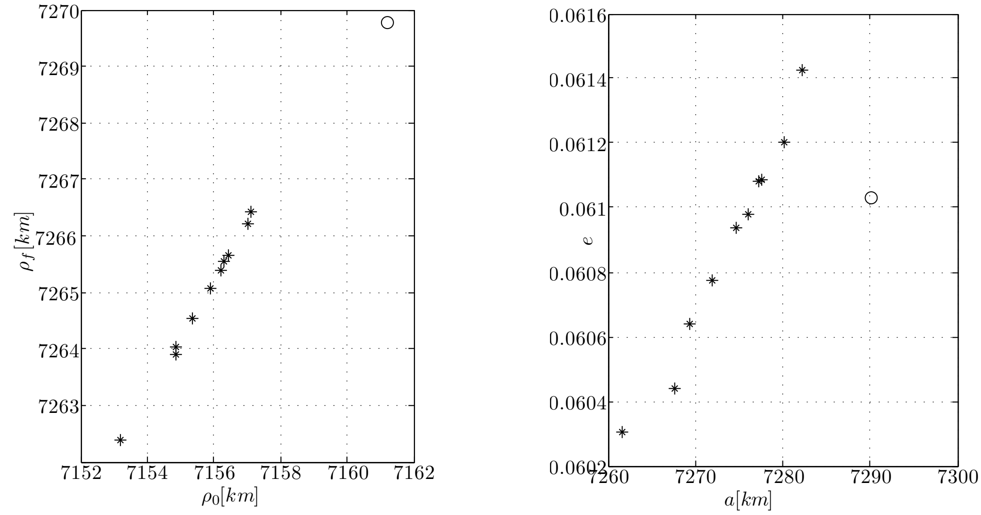


Figure 5.11: Best ρ_0, ρ_f and a-e for several noise seeds

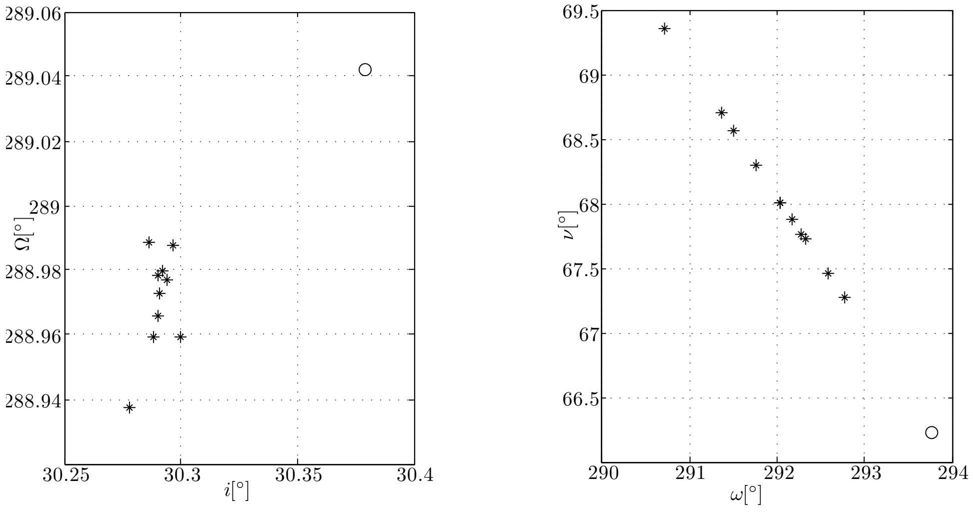


Figure 5.12: Best i-Ω and ω-ν for several noise seeds

Notice that the results are close to the true values, but they present a slight spreading. Probably these results have been obtained stopping the algorithm before reaching a true convergence.

There is another option to explain the spreading of the results, it is possible that the algorithm was truly converged but it was incapable of reaching the same value of EAE with different noise seeds.

In Fig. 5.13 the values of the final EAE are showed depending on the generation they reached.

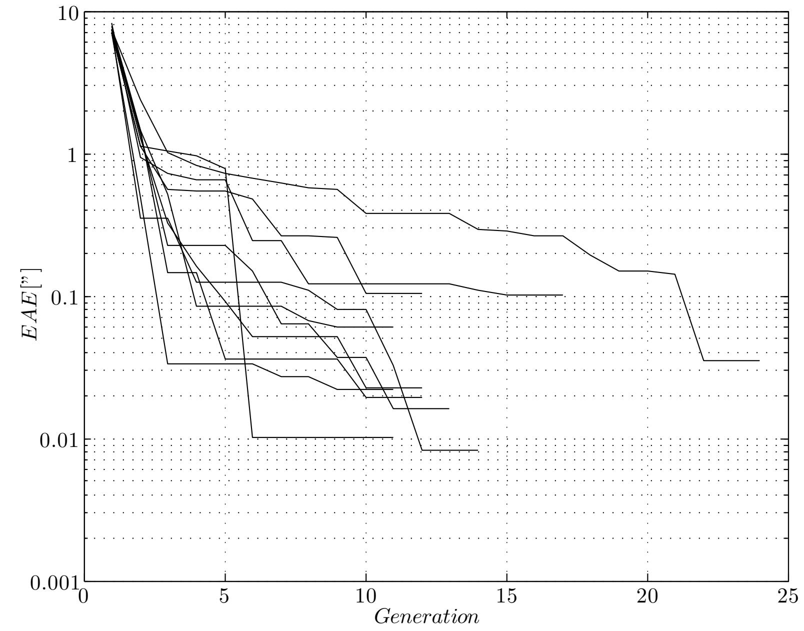


Figure 5.13: The *EAE* with several noise seeds

Notice that there is a large difference, both in the reached generation and the final EAE. Only few results have the EAE lower than 0.025", so we want to test the algorithm to force the convergence under this value.

The stopping condition has been changed, now it is forced to stop only if the EAE $\leq$ 0.025". In this way the algorithm is forced to continue the iterations, even if it remains with the same performance values unchanged for tens of iterations.

The results are shown in Fig. 5.14. Notice that all the runs with different noise seeds have reached the goal value. Another test with a goal of EAE < 0.01" showed that only 50% of the runs reached the goal. The other cases have been trapped in some local minima difficult to escape. The maximum number of generations has been fixed with 100 generations.

Notice in Fig. 5.14 that the best value of EAE has been reached with the minimum number of iterations, showing that the random number generation is very influential as expected.

The EAE is equal to the RMS in case of we have just a measurement to fit, i.e. $n_{obs} = 3$.

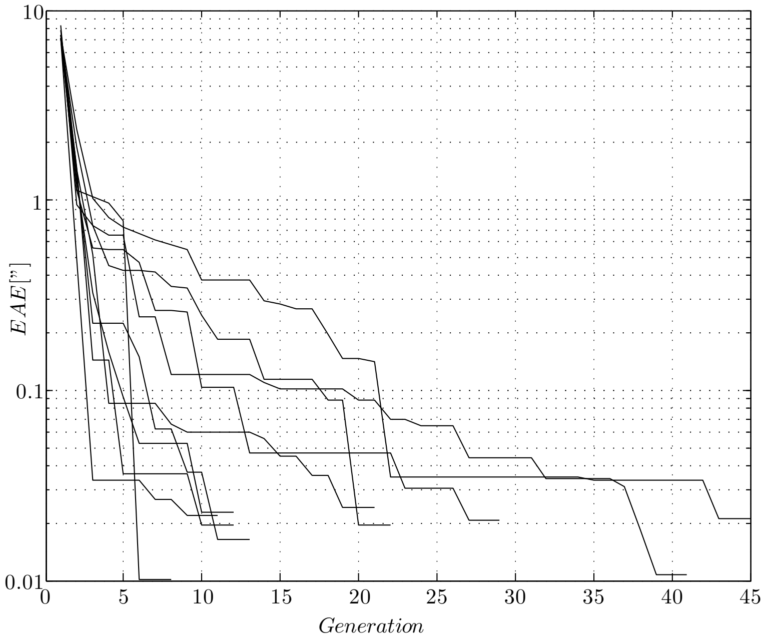


Figure 5.14: The EAE with several noise seeds and a fixed EAE goal

$$EAE = \arccos\left((fit_{k=1})^{\frac{1}{n_{obs}-2}}\right) \tag{5.2}$$

$$EAE = \arccos(fit_{k=1}) = \arccos(\hat{L}_{meas}(t_2) \cdot \hat{L}^*(t_2)) = \gamma_2 \tag{5.3}$$

Being γ_i the angle between the generic actual and reconstructed measurement. In this case the EAE is simply the angle between the intermediate actual and reconstructed measurement.

The RMS equation is in

$$RMS = \sqrt{\frac{1}{n_{obs}-2}\sum_{i=2}^{n_{obs}-1}(\gamma_i^2)} \tag{5.4}$$

If $n_{obs} = 3$ the Eq. (5.4) becomes:

$$RMS = \sqrt{\gamma_2^2} = \gamma_2 = EAE \tag{5.5}$$

The RMS evaluation is a typical method to show how well the reconstructed orbit of an asteroid fits the observation data.

Chapter 6

Test of the algorithm with real images

Macte nova virtute, puer: sic itur ad astra, dis genite et geniture deos.

- Publius Vergilius Maro, *Aeneis, Liber IX, 19 BC*

In this chapter the genetic algorithm is tested with real image. The measurement extraction is treated in detail with emphasis on astrometric calibration. A test with single image is presented to show how it is possible to correlate a single streak to an object in the NORAD catalog.

6.1 Introduction

Each method of starts with the assumption of topocentric observations. The angular observations are always described as ordered pairs of angles, usually RA and Dec and at a certain time:

t_1	RA_1	Dec_1
t_2	RA_2	Dec_2
...	...	...
t_n	RA_n	Dec_n

Table 6.1: The measurements set

The most common way of obtaining these angles is through pictures. The satellite images are taken and then the star background is compared to a catalog. Once the background is recognized, the image is calibrated; that means that we can assign to each point of the image a value of RA and Dec.

It is not possible to trust the pointing data of the telescope, or the camera, because the pointing error is usually higher than the accuracy we want to obtain.

6.2 The observatory

The observations have been kindly offered by the Franco Fuligni Observatory, managed by the Associazione Tuscolana di Astronomia (ATA), `http://nuke.ataonweb.it/`. The Association was founded in 1995 by a group of amateur astronomers, scientists and astrophysicists and has been named after a great Italian astrophysicist, Livio Gratton, formerly President of the Italian Astronomer Union.

ATA is the registered address of the Unione Astrofili Italiani (UAI) General Secretary and is often location for speeches devoted to public understanding of science held by renowned scientific popularisers. Currently the members of the association are almost 300, of which around 15 elements are integral part of the research team.

The observatory site is quite close to Rome (35 Km from the center) but far enough to have dark skies, especially due to the near hills shielding from the capital lights.

The current setting, used by the ATA research team mainly, but not exclusively, for asteroids observation, includes a Meade LX200-ACF 14 in Schmidt-Cassegrain configuration on a fix mount GM2000 (10 micron manufacturing), see Fig. 6.1. The whole system is controlled from a control room inside the building housing the Association.

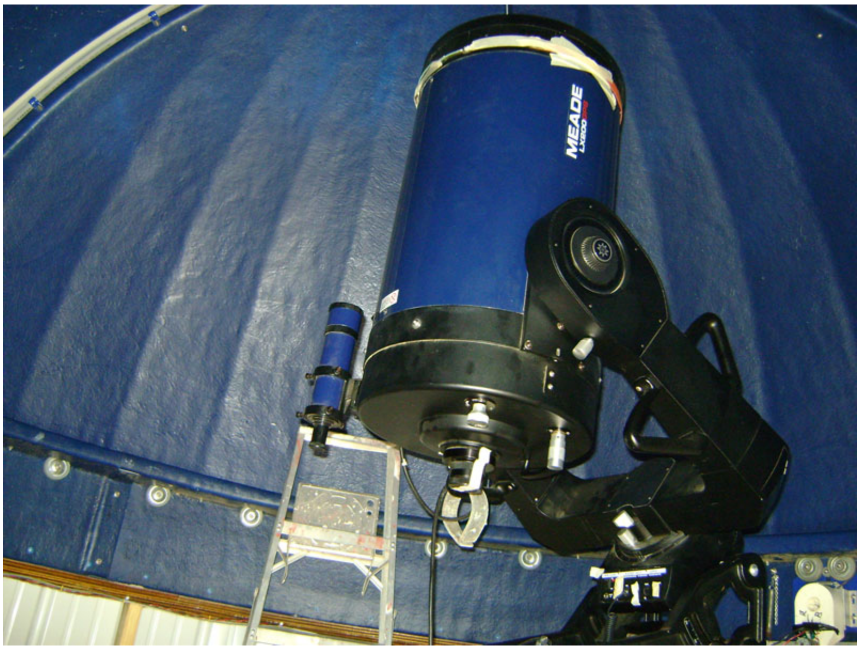


Figure 6.1: The Meade LX200 14”

The available cameras are a Santa Barbara Instrument Group (SBIG) ST-8 XME and a SBIG ST-9 XE. All the associations devices have been sponsored by its members or by private foundations that occasionally finance the Association projects. ATA is trying to obtain a funding also from public actors in such a way to integrate the contributions for a more ambitious project.

The research team has been recently reorganized and especially thanks to the purchase of the above-mentioned telescope, started an asteroid observation campaign expected to evolve in a most structured Near Earth Object (NEO) observation campaign. The main subjects studied by the team are:

- Minor Planet study
- Supernovae and Variable Stars observation
- Sun behavior and variability
- Satellite and Space Debris survey

Latitude	41.750085°
Longitude	12.770200°
Altitude	578m
MPC code	D06
FOV	13.8'x9.2'

Table 6.2: The observatory data

The following table shows the main data of the observatory, including the Minor Planet Center (MPC) code:

The dimensions of the FOV are useful to check what catalog can be applied to the image that we will obtain. If the FOV is so little, we need a very dense catalog to find stars to correctly locate the telescope pointing.

6.3 The star catalogs

A star catalog is basically a list of stars with some related properties. The basic catalog data contain position, magnitude, proper motion and an identificator. Many catalogs have been written during the years, having a great tradition also in ancient times, where only the visible stars were listed.

Modern catalogs contain data obtained by ground telescope and optical satellite that allow a very accurate determination of the telescope pointing. The most known catalogs are: Hipparcos and Tycho-2:

- HIPPARCOS

  The word Hipparcos is an acronym for *High precision parallax collecting satellite*. The satellite Hipparcos, which operated for four years, returned high quality scientific data from November 1989 to March 1993. ESA's Hipparcos space astrometry mission was a pioneering European project which pinpointed the positions of more than one hundred thousand stars with high precision, exactly 118218 stars, see [77]. Median precision of the five astrometric parameters (magnitude$\leq$9) exceeded the original mission goals, and are between 0.61.0 mas. The spacecraft carried a single all-reflective, eccentric Schmidt telescope, with an aperture of 29 cm (11.4 in). A special beam-combining mirror superimposed two fields of view, 58 degrees apart, into the common focal plane. This complex mirror consisted of two mirrors tilted in opposite directions, each occupying half of the rectangular entrance pupil, and providing an unvignetted field of view of about 11. The telescope used a system of grids, at the focal surface, composed of 2688 alternate opaque and transparent bands, with a period of 1.208 arc-sec (8.2 micrometre).

- TYCHO-2

  The Tycho-2 Catalogue is an astrometric reference catalog containing positions and proper motions as well as two-colour photometric data for the 2.5 million brightest stars in the sky, see Fig. 6.2 [36]. The Tycho-2 positions and magnitudes are based on precisely the same observations as the original Tycho Catalogue (hereafter Tycho-1; see CDS Cat. I/239) collected by the star mapper of the ESA Hipparcos satellite, but Tycho-2 is much bigger and slightly more precise, owing to a more advanced reduction technique. Components of double stars with separations down to 0.8 arcsec are included. Proper motions precise to about 2.5 mas/yr are given as derived from a comparison with the Astrographic Catalogue and 143 other ground-based astrometric catalogs, all reduced to the Hipparcos celestial coordinate system. Tycho-2 supersedes in most applications Tycho-1, as well as the ACT (CDS Cat. I/246) and the TRC (CDS Cat. I/250) catalogs based on Tycho-1.

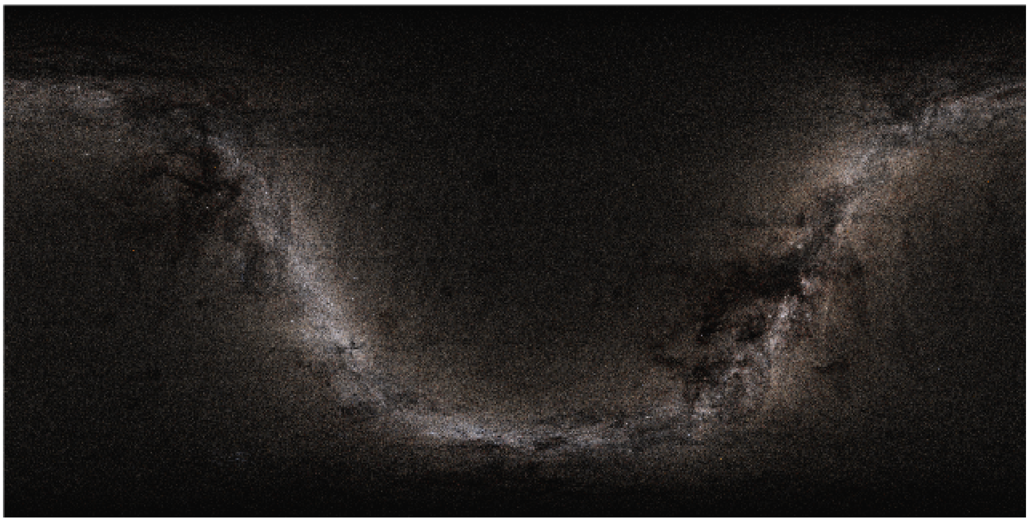


Figure 6.2: The Tycho-2 stars

Notice in Fig. 6.2 that the density of the stars is absolutely not constant, so the amount of stars that will be seen is highly dependent on the portion of the sky we are pointing to.

These catalogs are usually not enough complete to calibrate telescope images with little FOV. Catalogs with more stars are needed to be sure that each portion of the sky can be calibrated, because the stars density is not constant.

The calibration is an hard procedure usually made by a software. There are many software that can calibrate images, free as well. The results here presented have been obtained through Astrometrica, that is a interactive software tool for scientific grade astrometric data reduction of Charge Coupled Device (CCD) images.

The following catalogs can be accessed from the software Astrometrica `http://www.astrometrica.at/`:

- UCAC 4

  Observations for the USNO CCD Astrograph Catalog (UCAC) started in early 1998 and were completed in 2004. With these observations, the UCAC is the first modern high-density, full-sky star catalog that is not based on photographic images of the sky, but on recent CCD observations. After two intermediate releases, which did not cover the whole sky, the first complete catalog (UCAC 3) was released in 2009, followed by the final release of UCAC 4 in 2012. The UCAC 3 includes positions, proper motions and magnitudes for 113,780,093 objects. Reference star positions in UCAC 4 are accurate to about 0.02" for brighter stars (10mag to 14mag), and a precision better than 0.1" is expected at the limiting magnitude of 16mag. Native magnitudes have been measured in one single, non-standard color, but the catalog includes five-band photometry (B,V,g,r,i) from the AAVSO Photometric All-Sky Survey (APASS) (American Association of Variable Star Observers (AAVSO) Photometric All-Sky Survey) for over 50 million stars.

  Astrometrica can access a local copy of the UCAC 3, or query VizieR to download reference star data.

- PPMXL

  PPMXL is a combination of the data from the United States Naval Observatory (USNO)-B1.0 and the infradred 2MASS catalog. It aims to be complete from the brightest stars down to about magnitude V=20 full-sky. PPMXL contains about 910,468,710 objects, and therefore is the largest collection of International Celestial Reference System (ICRS) positions and proper motions at present. The catalog includes astrometric positions with an accuracy (at epoch J2000.0) 0.08" to 0.12" where 2MASS astrometry was available (that is, for 410 million objects), and 0.15" to 0.30" where no data from 2MASS could be used. Proper motions are absolute in the ICRS reference frame (not relative, as in the

USNO-B1.0). Magnitudes in the PPMXL were taken from USNO-B1.0, so there is no improvement in the photometric calibration compared to the USNO catalog.

Astrometrica will query VizieR to download reference star data from PPMXL.

- NOMAD

  The Naval Observatory Merged Astrometric Dataset (NOMAD) is a merged catalog, with positions and magnitudes for 1.1 billion stars from several source catalogs, including Hipparcos, Tycho-2, UCAC 2, and USNO-B 1.0 (see below). For each star the presumably best astrometric and photometric data were chosen from one of the source catalogs. Positions are mainly from the Hipparcos for brighter stars (down to 10mag), from UCAC 2 for stars to down to 16mag (in those parts of the sky covered by the UCAC 2), and from USNO-B1, down to the limiting magnitude of the latter (21mag). Positional errors is about 0.015" for stars from Hipparcos, 0.07" for stars from UCAC, and 0.2" fro stars from USNO-B 1.0 (not taking into account any systematic offsets between these catalogs). By setting the lower magnitude limit for the reference stars in the program settings of Astrometrica accordingly, users may control the balance between reference stars with data from the UCAC and the USNO-B.

  Due to it's enormous size (100 Gigabytes), the catalog has never been distributed on CD or DVD or some other media, but Astrometrica will automatically query VizieR to download NOMAD reference star data.

The following table summarizes for each catalog the number of stars listed:

Catalog	Number of stars	Sensor Medium Density	Sensor Lowest Density
Hipparcos	$10^6 \cdot 0.12$	0.1 stars	0 stars
Tycho-2	$10^6 \cdot 2.50$	2.1 stars	0.2 stars
UCAC 4	$10^6 \cdot 114$	97.5 stars	8.1 stars
PPXML	$10^6 \cdot 910$	777 stars	64.8 stars
NOMAD	$10^6 \cdot 1100$	940 stars	78.3 stars

Table 6.3: The number of stars for catalog

The Sensor medium density has been considered for the size of the sensor of about 1/28 square degrees; while the lowest density has been simply calculated as 1/12 of the medium density. A good calibration can be conducted using at least a ten or so stars, so the UCAC 4 catalog does not have enough stars for our images.

We use the PPXML or NOMAD catalog to calibrate, but consider that a lot of stars will not be seen because of the limiting magnitude of the telescope, that is lower than 16; this is to explain that a very good catalog is needed and very few stars will be seen, so the calibration will be a tough process.

6.4 A sample image

A sample image of the telescope is shown in Fig. 6.3: very few stars are visible, the background is quite noisy but three streaks are easily identified.

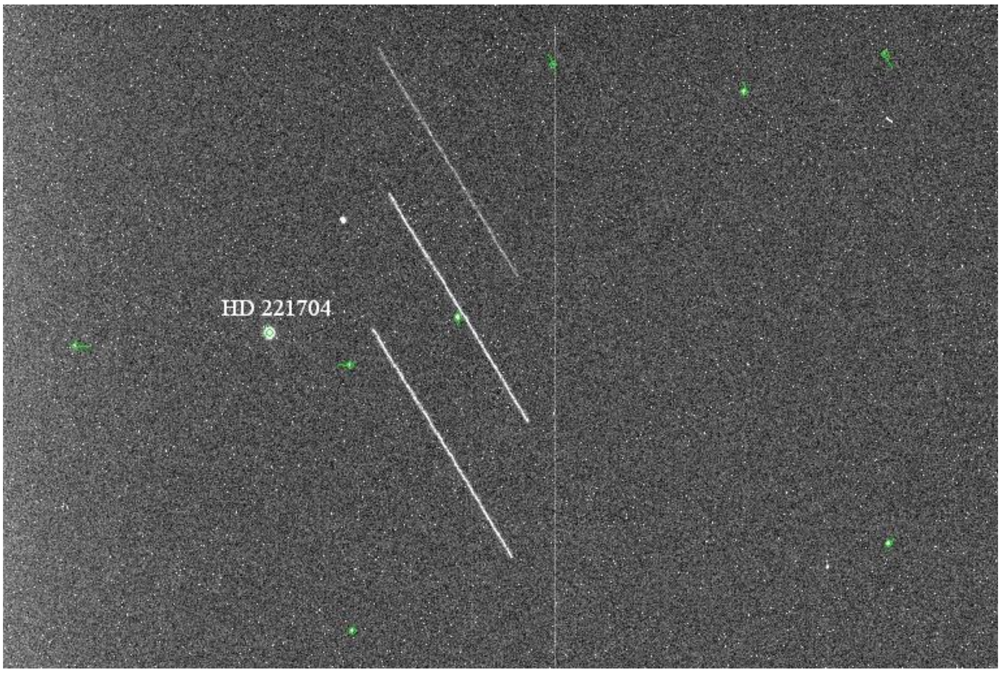


Figure 6.3: The sample image with exposition of 15 seconds

The streaks are related to the satellites in Table 6.4. These objects are colocated geostationary satellites. The picture has been taken with automatic star tracking mode, so the stars are fixed and the satellites are moving against the background.

The satellites are all active and mantained in an imaginary 150 km cube in space. The satellites are controlled to avoid collisions and to maintain the orbital position respect to the ground. Usually the satellites have a distance between each other of at least 5 km, see [18].

NORAD ID	First name	Name at observation epoch
27499	Hot Bird 6	Eutelsat Hot Bird 13A
29270	Hot Bird 8	Eutelsat Hot Bird 13B
33459	Hot Bird 9	Eutelsat Hot Bird 13C

Table 6.4: The spotted satellites

The data of the image are in Table 6.5, the Coordinated Universal Time (UTC) time is from the GPS module.

RA	23h34m20s
Dec	-06°26'7.9"
Rotation angle	58.61°
Focal Length	3431 mm
FOV	13.8'x9.2'
Pixel size	0.54"x0.54"
Color	16 bit
Binning	1
Day	19/10/2012
UTC Time	20:47:51.588
Exposure time	15 s
CCD temperature	11.2°C
Telescope elevation	41.8°
Airmass	1.49

Table 6.5: The image data

The data extrapolated from the TLE of the three satellites for the month of October 2012 are shown in Fig. 6.4, Fig. 6.5 and Fig. 6.6.

The satellites do not show relevant maneuvers for the entire month, the maximum variation of the semimajor axis is under 3 km.

In Fig. 6.7 the relative size of the picture is shown with the Moon.

6.4.1 The results of the astrometric calibration

The calibration has been conducted with the NOMAD catalog, finding 9 stars as references, all these 9 stars have been correlated to catalogued stars. The final estimated error of the image calibration is: $dRA = 0.16", dDec = 0.12"$ with $dmag = 0.05mag$. The pixel size is $0.54"x0.54"$, so we have obtained a calibration with a subpixel accuracy, under 1/3 of the pixel size.

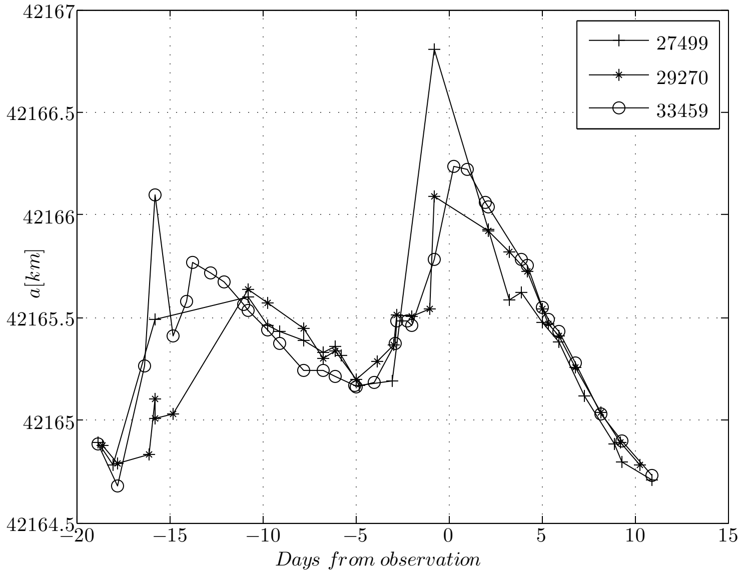


Figure 6.4: The semimajor axis from the TLEs

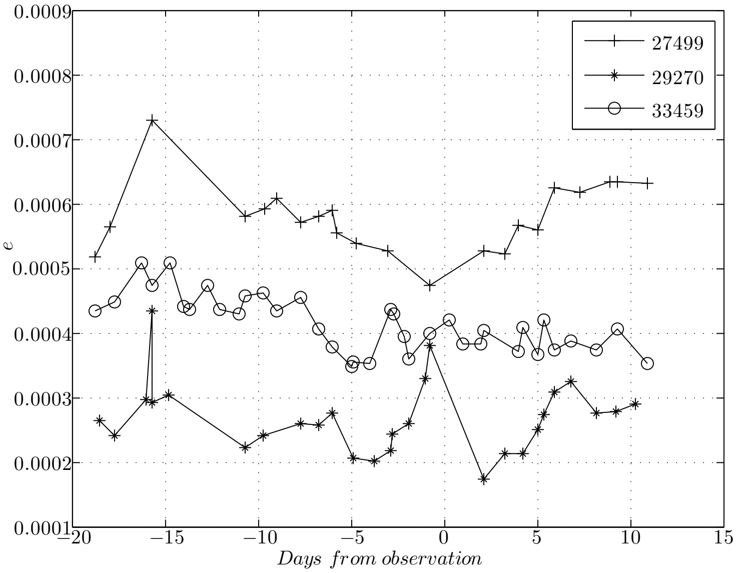


Figure 6.5: The eccentricity from the TLEs

The calibrated pointing is: $RA = 23h34m20.00s, Dec = -06°26'07.90$”, while the commanded pointing was: $RA = 23h36m06.59, Dec = -06°30'09.95$”. The error between the two vectors is about 30’.

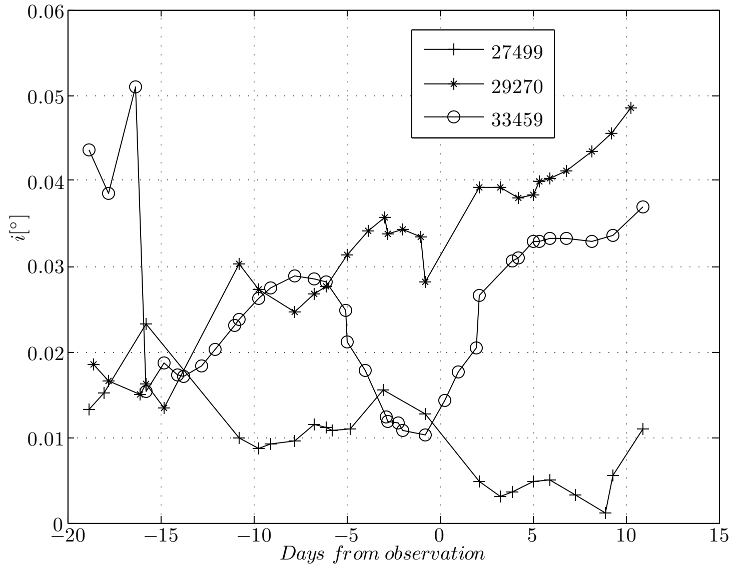


Figure 6.6: The inclination from the TLEs

The errors in RA and Dec for each identified star are shown in Fig. 6.8, while the error in magnitude is in Fig. 6.9.

The software output for the brightest star is shown in Fig. 6.10. The star is $HD221704$ with a visual magnitude of 9.2. In the right part of the image the Point Spread Function (PSF)-fit is shown, this star has a very good Signal Noise Ratio (SNR) equal to 105.7. The coordinates of the star have been calculated with the centroid of the bright pixels.

6.5 The observer position

We need to locate the observer position in the same reference frame of the observations. The observed RA and Dec of the satellites are in J2000, while the station coordinates are taken from GPS, so they are in World Geodetic System 1984 (WGS84).

WGS84 is the reference for the GPS, the yielded latitude and height refer to the reference ellipsoid. Notice that the height from GPS does not correspond to the height over Mean Sea Level (MSL), see Fig. 6.11. The geoid is the geometric figure that approximates the MSL; the geoid is the shape that the surface of the oceans would take under the influence of Earth's gravitation and rotation alone, in the absence of any other influences. The geoid does not represent the exact sea level because of ocean circulation patterns, salinity,

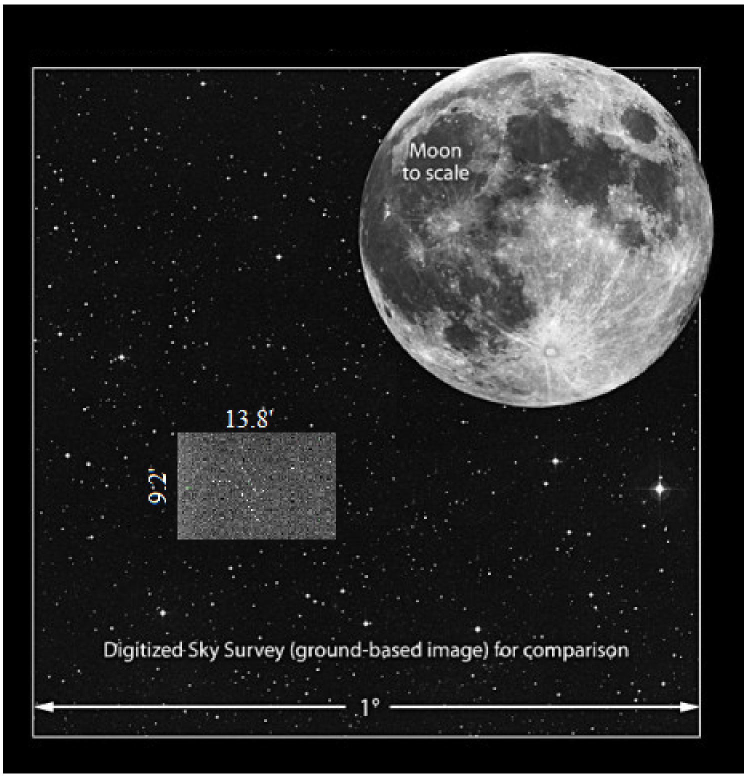


Figure 6.7: The relative size of the sample image

temperature differences, wind and tides.

The latitude data in Table 6.2 refers to the geodetic latitude, while the altitude refers to the ellipsoidal height. These data are referring to the reference ellipsoid of WGS84, whose main data are the semimajor axis of the ellipsoid $a = R_{\oplus} = 6378.137km$ and the flattening $f = 1/298.257223563$. The flattening is defined in Eq. (6.1).

$$f = \frac{a-b}{a} \tag{6.1}$$

Where b is the semiminor axis of the ellipsoid, also called the polar axis. Actually, b is a derived quantity from a and f:

$$b = a(1-f) = 6356.752km \tag{6.2}$$

The difference $(a - b)$ is about 21 km.

While the flattening is the most known quantity to identify the compression of a sphere, the eccentricity of the Earth $e_{\oplus}$, see Eq. (6.3), is used to obtain geocentric coordinates from geodetic coordinates.

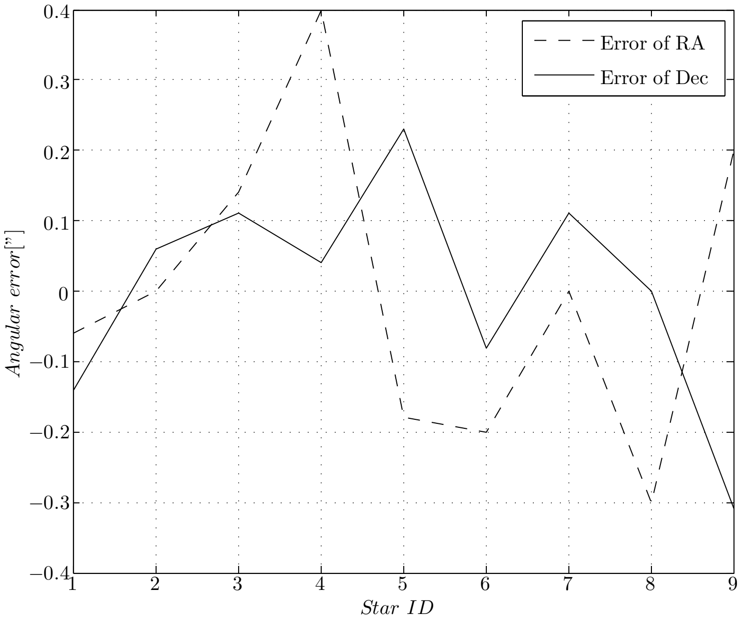


Figure 6.8: The angular error for each detected star

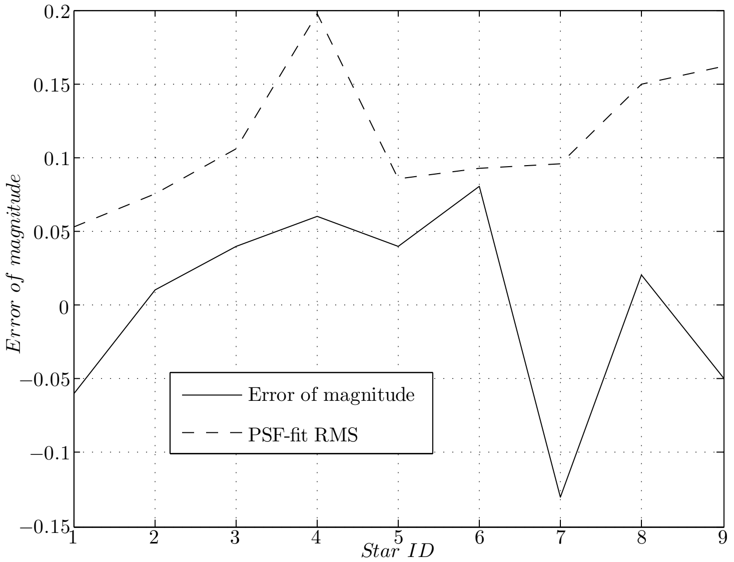
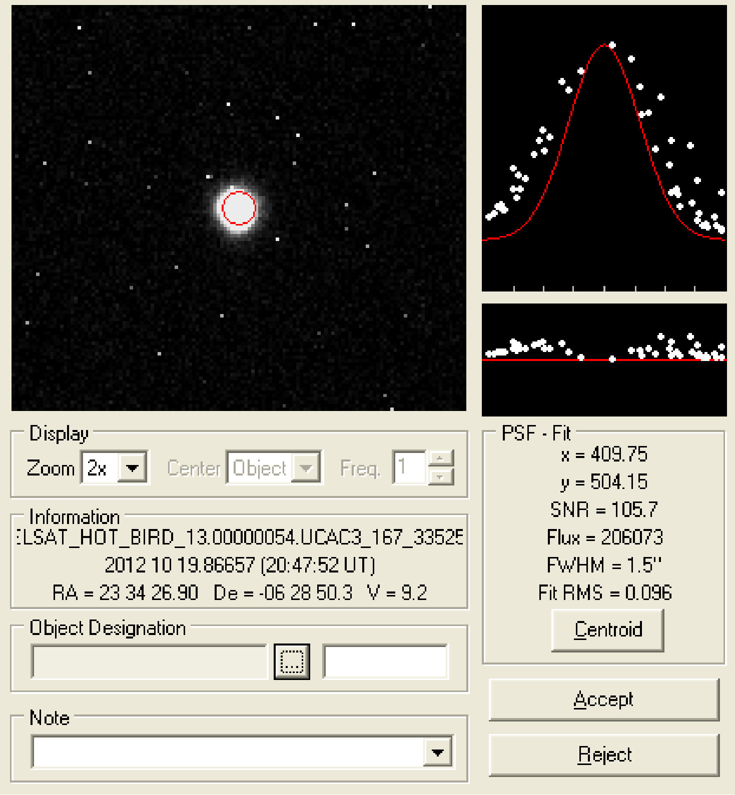


Figure 6.9: The magnitude errors for each detected star

$$e_{\oplus} = \sqrt{1-(b/a)^2} = \sqrt{2f-f^2} = 0.081819191 \tag{6.3}$$



Zoom 2x

Center Object

Freq. 1

:LSAT_HOT_BIRD_13.00000054.UCAC3_167_33525
2012 10 19.86657 (20:47:52 UT)
RA = 23 34 26.90 De = -06 28 50.3 V = 9.2

x = 409.75
y = 504.15
SNR = 105.7
Flux = 206073
FWHM = 1.5"
Fit RMS = 0.096

Figure 6.10: The brightest star in the image: HD 221704

Notice that $e_\oplus$ refers to the planet geometry, not to the planet orbit.

We can obtain two auxiliary quantities, obtained through the geometric properties of the ellipse:

$$C_\oplus = \frac{R_\oplus}{\sqrt{1 - e_\oplus^2 \sin^2(\phi_{gd})}} \tag{6.4}$$

$$S_\oplus = \frac{R_\oplus(1 - e_\oplus^2)}{\sqrt{1 - e_\oplus^2 \sin^2(\phi_{gd})}} \tag{6.5}$$

$C_\oplus$ is usually known as the radius of curvature in the meridian.

Eq. (6.12) shows the geometry of the problem of finding the observer site in a geocentric reference frame [79].

$$\vec{r}_{site_{ITRF}} = \begin{bmatrix} (C_\oplus + h_{ellp})\cos(\phi_{gd})\cos(\lambda) \\ (C_\oplus + h_{ellp})\cos(\phi_{gd})\sin(\lambda) \\ (S_\oplus + h_{ellp})\sin(\lambda) \end{bmatrix} \tag{6.6}$$

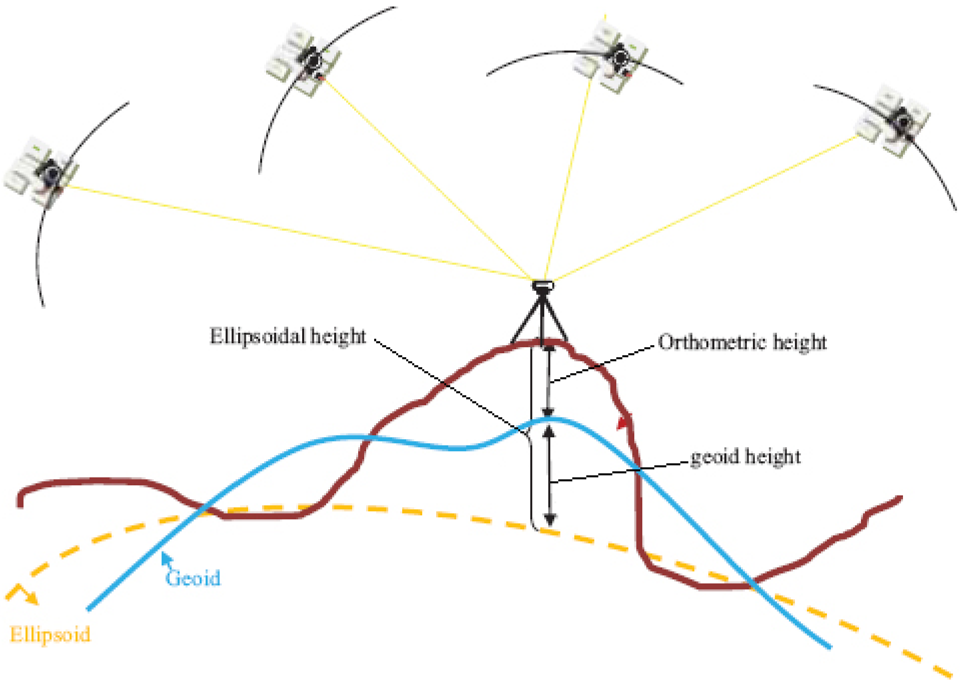


Figure 6.11: The ellipsoidal height

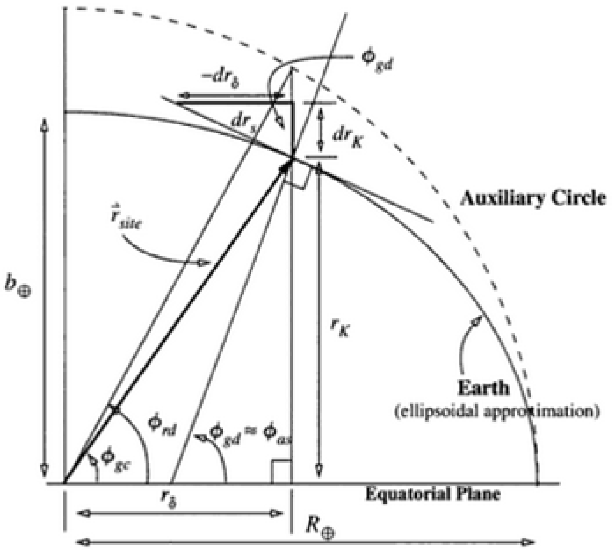


Figure 6.12: The geodetic latitude

The output of Eq. (6.6) is in ITRF, the next step is to rotate this position in ICRF, we will not make any differences between ICRF and J2000, that can be considered parallel; actually the difference of some tens of *mas* is negligible for our application.

The rotation from ITRF to ICRF is obtained from the International Earth Rotation Service (IERS) site: `http://www.iers.org/`. This rotation matrix includes the effect of [8]:

- Precession
- Nutation
- Sidereal time
- Celestial pole offset
- UT1-UTC
- Polar motion
- Ocean tides

The observer position in ITRF is:

$$\vec{r}_{site_{ITRF}} = \begin{bmatrix} 4648.1 \\ 1056.5 \\ 4225.3 \end{bmatrix} km \tag{6.7}$$

While the position at the start of the observation time t_0 is:

$$\vec{r}_{site}(t_0) = \begin{bmatrix} 4738.7 \\ -556.5 \\ 4219.3 \end{bmatrix} km \tag{6.8}$$

The position at $t = t_f = t_0 + 15$ is:

$$\vec{r}_{site}(t_f) = \begin{bmatrix} 4739.3 \\ -551.3 \\ 4219.3 \end{bmatrix} km \tag{6.9}$$

The observer has changed its position of about 5.2 km in 15 seconds. We can find the same displacement applying the formula in Eq. (6.10)

$$|\vec{r}_{site}(t_f) - \vec{r}_{site}(t_0)| = \omega_{\oplus} r_{\delta}(t_f - t_0) = 5.2km \tag{6.10}$$

with $\omega_{\oplus} = 7.292115 \cdot 10^{-}5$ rad/s the angular velocity of the Earth and r_{δ} the distance of the observer location from the rotation axis of the Earth:

$$r_\delta = (C_\oplus + h_{ellp})\cos(\phi_{gd}) \tag{6.11}$$

The maximum displacement over the Earth surface is at the equator, where there is a linear velocity of about 465 m/s, in 15 seconds the displacement is about 7 km.

6.6 Object correlation with the NORAD database

The measurements extracted from the calibrated images are:

$$\hat{L}(t_0) = \begin{bmatrix} 23h34m9.23s \\ -6°27'35.8" \end{bmatrix} = \begin{bmatrix} 0.9873 \\ -0.1118 \\ -0.1125 \end{bmatrix} \tag{6.12}$$

$$\hat{L}(t_f) = \begin{bmatrix} 23h34m24s \\ -6°27'35.1" \end{bmatrix} = \begin{bmatrix} 0.9875 \\ -0.1108 \\ -0.1125 \end{bmatrix} \tag{6.13}$$

Notice that the angle between the two measurements is just about 3.7$'$.

Proceeding with the iniatilization in the plane $[\rho_0, \rho_f]$ we can find the points in Fig. 6.13.

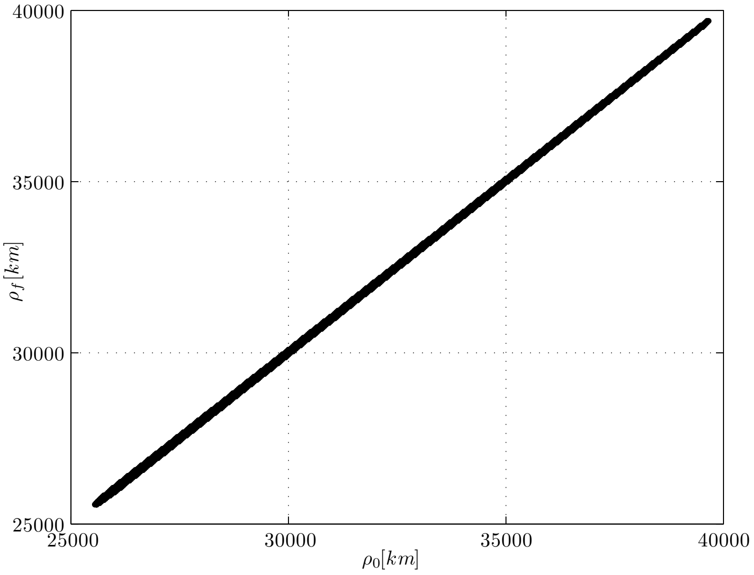


Figure 6.13: The initialization for ρ_0 and ρ_f

These points represent the orbits whose first orbital parameters are in Fig. 6.14.

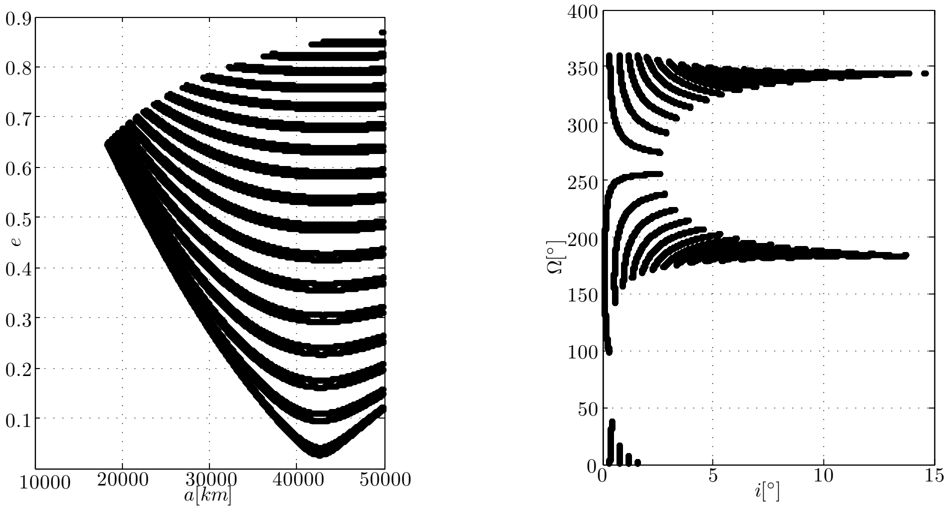


Figure 6.14: a-e and i-Ω of the initial points

Imposing that the distance at the initial time is the same at the final time $\rho_0 = \rho_f$ with a step of 100 km, we obtain the results shown in Fig. 6.15.

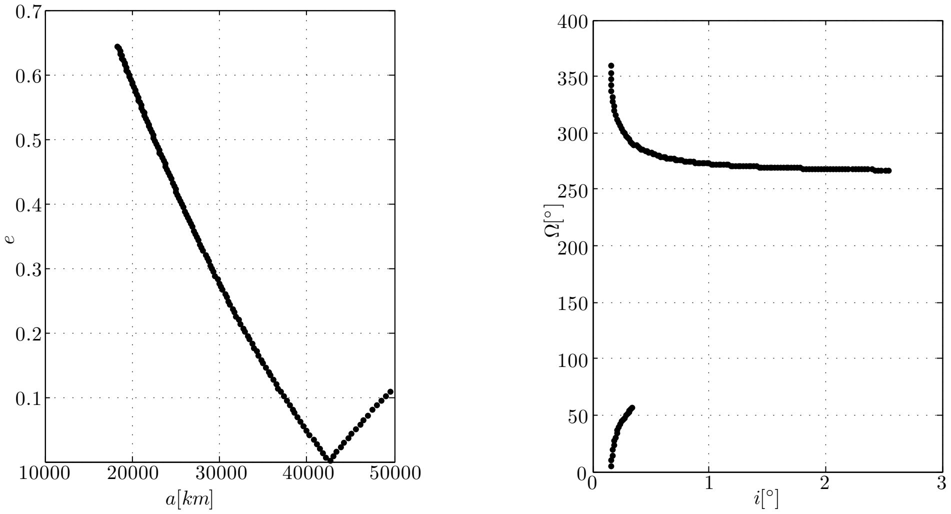


Figure 6.15: a-e and i-Ω for $\rho_0 = \rho_f$

The orbits that represent these found orbital elements are shown in Fig. 6.16 and Fig. 6.17.

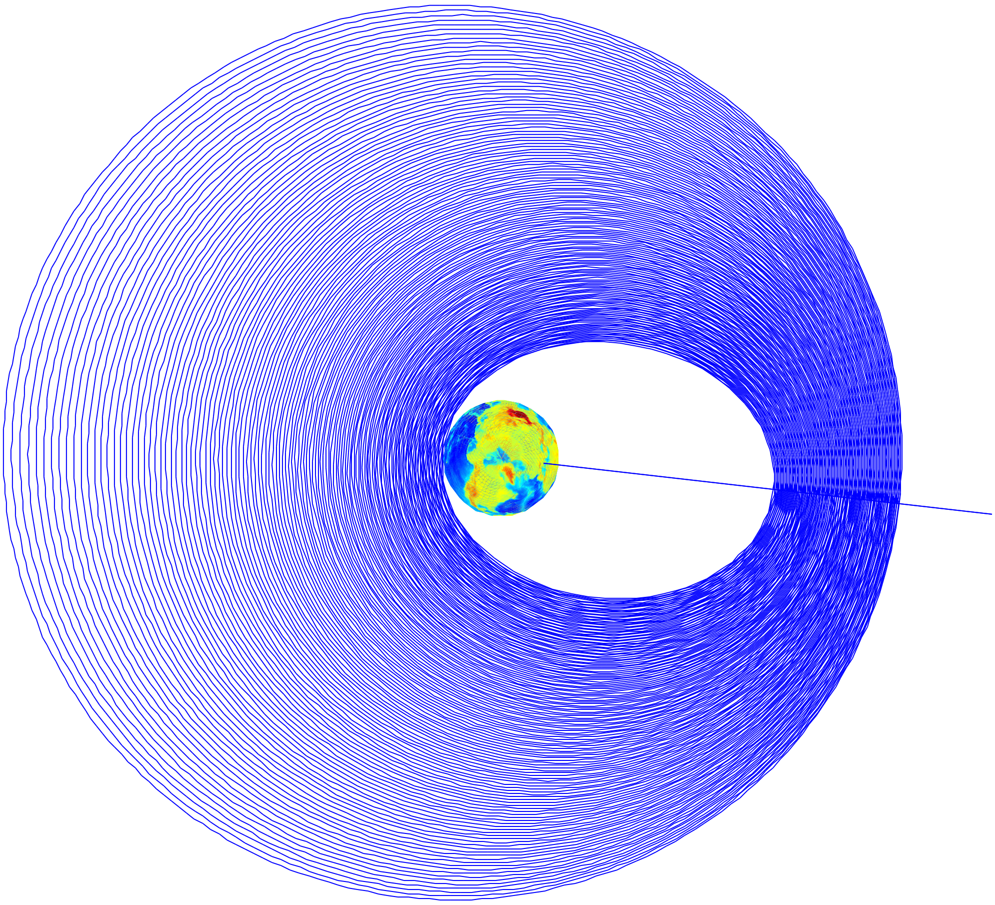


Figure 6.16: The orbits for $\rho_0 = \rho_f$, x-y plane

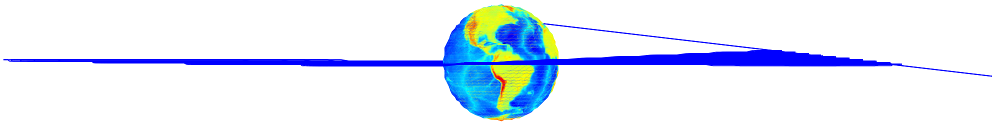


Figure 6.17: The orbits for $\rho_0 = \rho_f$, x-z plane

A first selection has been made limiting the NORAD catalog for the objects with inclination lower than the maximum inclination of the found points, about 14 degrees.

The inclination is the only constant angular parameter that is always defined. The RAAN is not defined for orbit with zero inclination, while the argument of perigee is not defined for zero eccentricity orbits.

The semimajor axis has been used as well to decrease the number of objects to be propagated with a SGP4 program.

All the objects with $17000 \leq a \leq 50000 km$ and $i \leq 14°$ have been propagated considering the correct propagation time issued for each object from the TLE. For each object the γ angle has been computed at the initial and the final time. Only the results with values under ten times the minimum found value are shown in Fig. 6.18.

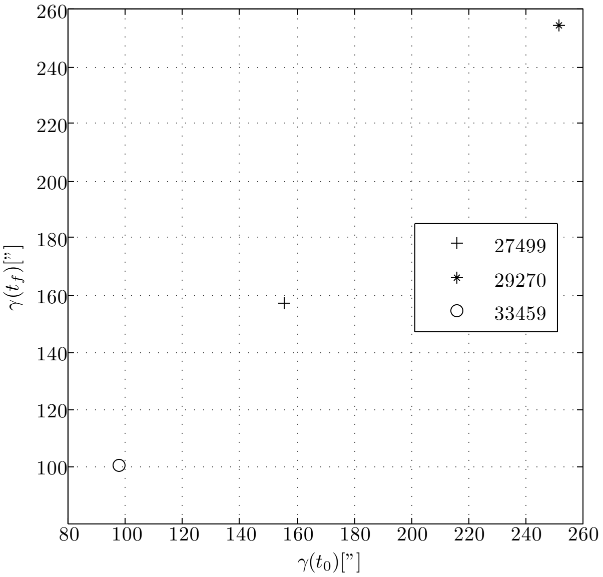


Figure 6.18: The γ angles for the NORAD satellites

These satellites correspond to the three satellites that we are supposed to observe. The propagated states at the time of the the observations are shown in Fig. 6.19 with the actual observations.

The TLEs have been propagated through a Simplified General Perturbations 4 (SGP4) program; actually this propagator (freely distributed in many languages) switches to a different kind of propagator, the Simplified Deep-space Perturbations 4 (SDP4), for satellites with periods greater than 225 minutes.

Notice that the TLEs are in True Equator Mean Equinox (TEME) while we are looking for the J2000/ICRF reference frame, so we need to rotate the position vectors propagated (the J2000 is a Mean Equator Mean Equinox (MEME) reference frame). This rotation approximates the effects of precession

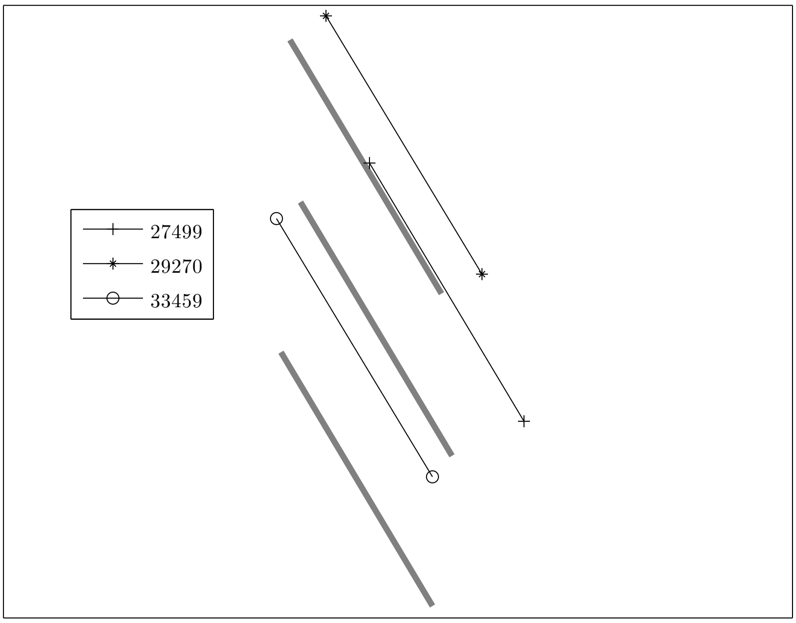


Figure 6.19: The recontructed and the actual measurements

and nutation.

6.7 The aberration of light

The aberration of light takes into account the relativity theory. The light speed c is not infinite, we know that:

$$c = 299792458 \frac{m}{s} \tag{6.14}$$

The distance of the GEO satellites, about 38000 km, implies that a delay of about 125 ms has to be considered. This very known delay, think to communication issues, does not seem to be so high when considering a process of orbit determination; however it is very important because it has a significant effect.

The satellite is moving with a speed of about 3 km/s, in 125 ms it moves of about 400 m. Notice that the pixel resolution is 0.54”, so the distance between pixels is about 100 m for an object 38000 km far. We can quantify the relativistic effect in terms of 4 pixels, so with an angular error of about 2”, that is absolutely unacceptable for our purposes.

The observer also is moving in an inertial frame during the time of light traveling, but due to the minor velocity, it moves of just about 50 m.

6.8 Conclusions

The process to identify the satellite from a single image is not complicated, unless we have colocated satellites as in the presented case. Also notice that we can not make any hypotheses on the direction that the satellites have in the image.

In this case it was easy because no correlation has been found for satellites with large semimajor axis and inclination near to 180 degrees.

The accuracy of the TLEs for GEO satellite has been proved to be in the order of 1/25 of degrees, [27]: about 150”, for an error of 25 km along track and 10 km in cross track.

With these errors we can not identify correctly which streak belongs to which satellite. We can assume that magnitude can help us with this correlation.

27499 has been constructed by Alcatel Space based on the Spacebus-3000B3 satellite bus, with about 3900 kg. 29270 and 33459 have the same Astrium bus, based on Eurostar-3000, with about 4900 kg.

So we can assume that 27499 is streaking in the upper part of the image, and probably 29270 is the medium streak and, as a consequence, 33459 is the lower part of the image. Notice that for a good correlation, or linkage, it is crucial to model the error of the TLEs that, unfortunately, are published without any error information.

Chapter 7

A proposal for a space-based camera for objects tracking

Satellite collisions will produce a number of fragments, some of which may be capable of fragmenting another satellite upon collision, creating even more fragments.

- Donald J. Kessler, *Journal of Geophysical Research, 1978*

In this chapter the arising problem of space debris is briefly introduced. The need of a program for Space Situational Awareness is common for every space agency. Two different solutions for a space-based camera are presented: one involving a micro-satellite and one related to a payload on the ISS. A pan-tilt system is proposed for the second solution; different camera modes are taken into account explaining the image processing.

7.1 Introduction

Currently, space missions must take into account a relatively new threat, which is represented by space debris, i.e. the orbital residuals of past artificial satellites and rockets. This problem has arisen in the last 25 years and requires specific strategies for mitigation, with the main intent of avoiding collisions between orbital debris and spacecraft. Space debris monitoring and orbit determination is an essential premise to this task, [4]. Thousands of object are tracked to estimate their orbits, moreover the total number of object is increasing, see Fig. 7.1.

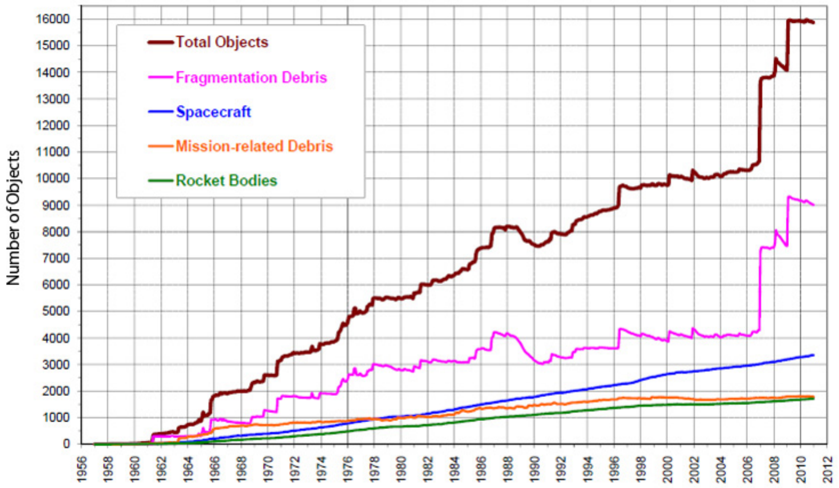


Figure 7.1: The increasing number of tracked objects

Space Situational Awareness (SSA) represents an important political and military concern and is related to a multitude of aspects, such as safety of spacecraft operations, safety for ground facilities (prediction of reentering objects and related hazards), basic information on the space debris population. In addition, for space missions an appropriate knowledge of debris environment allows realizing corrective maneuvers to prevent collisions.

Space debris observation campaigns are aimed at preventing orbiting spacecraft against impacts. Ground-based optical telescopes and radar can be used for the detection and orbit determination of space debris. The differences between the detection capability of these two systems is basically due to the fact that the power of the signal received by an optical system is proportional to $1/(d^2)$ (where d is the distance from the orbiting object) whereas for a radar it is proportionally to $1/(d^4)$. For this reason radars are mainly used for the surveillance of LEO objects, whereas optical systems allow detecting more distant objects.

With a ground-based radar system it is possible to detect LEO objects with size down to 1 centimeter; these objects cannot be seen with a ground-based optical system. On the other hand, optical systems have several advantages with respect to radars, such as the possibility of monitoring GEO regions, as well as the capability of identifying the shape and the attitude of an orbiting object. Another important feature of optical systems is the possibility to compare different images from distinct observation sites, thus allowing an improved orbital determination.

In situ space debris monitoring systems are based on impact sensors and on the analysis of surfaces returned from space, but these methods do not allow preventing the possible collisions. Otherwise, in situ measuring of space debris can be exploited using a microsatellite equipped with an optical system [16] or radar based system. A possible solution of an optical system is given by an Elphel camera, see Fig. 7.2.

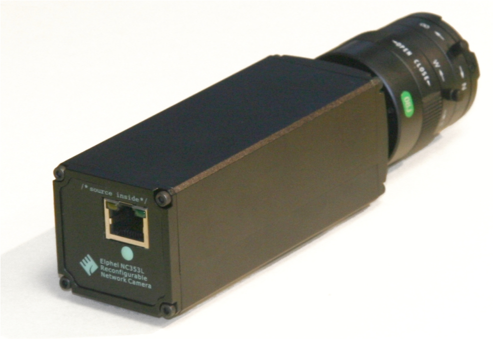


Figure 7.2: The Elphel NC353L camera

A radar system is not suitable for university microsatellite due to the large amount of required power. Another advantage of the use of an orbiting optical system is the possibility to have a better resolution and to observe smaller objects in better illumination conditions.

7.2 An ISS payload for space debris monitoring

The increasing number of space debris is becoming an important issue for all the active satellites. The LEO population is constantly increasing due to launches, orbit collisions and antisatellite missile tests. The Inter-Agency Space Debris Coordination Committee (IADC) has developed some mitigation guidelines to preserve the space environment; however, the current scenario forces Space agencies to take into account the presence of space debris. Satellite collisions have been observed, and thousands of fragments have been generated, as foreseen in [44].

The mission analysis and the mission planning are affected by the operations in a hazardous environment, where there is a risk of an unintentional collision. In this frame there is the need of being able to determine precisely and periodically the orbits of the space debris. Ground-based optical and radar observations are usually used to determine the orbit of the satellites and of the largest space debris. Space-based observations are another source of optical measurements. It is possible to use some techniques by which it is possible to determine the orbital parameters of an orbiting object using the images acquired by a camera mounted on the International Space Station (ISS), see [3]. The ISS offers the possibility of installing a payload, without the need of launching into space a dedicated satellite to acquire images for orbit determination. The pan-tilt system with the camera should be installed on the external surface of the ISS. One possible site could be the Columbus-External Payload Facility (CEPF), see Fig. 7.3.

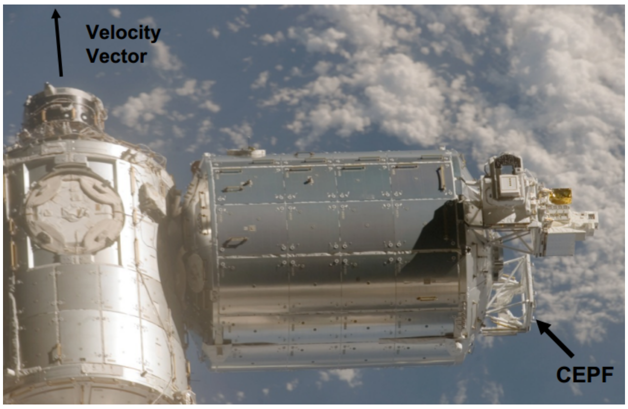


Figure 7.3: The CEPF

The CEPF provides four powered external attachment sites for scientific payloads or facilities: one nadir site (platform faces Earth), one zenith site (platform faces up), and two starboard sites (platforms face to the right of Columbus as viewed along its line of flight). Each payload may have a mass of up to 290 kg. The camera system for orbit determination will have a mass of about 5 kilograms, so it is suitable to be included in a multiple accommodation payload. A system of this class already flew on ISS and it was called European Technology Exposure Facility (EuTEF) [15].

EuTEF is a programmable, fully automated, multi-user facility with modular and flexible accommodation for a variety of technology payloads. EuTEF returned on Earth in 2009 with the Shuttle STS-128 mission. While one site on the CEPF is already occupied [69], other two are expected to be occupied in 2015 by Atomic Clock Ensemble in Space (ACES) [13] and Atmosphere-Space Interactions Monitor (ASIM) [61]. One starboard site has no programmed payloads; this could be a suitable place for our payload.

The ISS nominal attitude is the XVV ($X - axis$ in the Velocity Vector) as shown in Fig. 7.4; the cone representing the field of view of the camera has been placed on the CEPF. The $Z - axis$, obtained with the right hand rule, is nadir pointing, this direction identifies the portion of the sky occupied by the Earth. The negative $Y - axis$ view will be obstructed by the structure of the ISS, so the favorite directions of observation will be inside the octant with X positive, Y positive, Z negative, in this region only the solar panels could yield a limitation in the field of view. The solar panels have a pre-configured motion, so it is possible to manage the scheduling of observations to avoid the presence of the panels in the field of view.

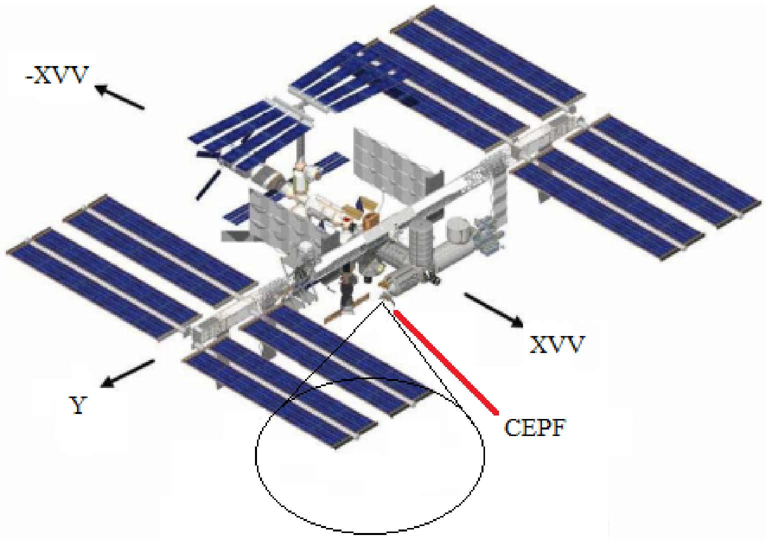


Figure 7.4: The XVV Direction

7.3 The camera modes and the image processing

The camera will operate continuously and autonomously in a *scanning mode*. The pan-tilt dispositive will be used to search objects in areas of space avoiding the presence in the FOV of Earth, Moon and parts of the ISS. The pan-tilt will be operated scanning the available region of the sky, pointing with near inertial attitude for little periods. The ISS attitude is not inertial, so the pan-tilt will compensate the slow drift keeping stars fixed. In the Fig. 7.5 a pan-tilt camera is shown as example.

Figure 7.5: A pan-tilt camera

Once an object is detected, the camera will go into *tracking mode*, trying to maintain the object in the field of view. A test will be performed to control that the moving point is always related to the same object. The object has to appear moving along a near straight line, with a coherent angular velocity. Once the object is recognized as a possible target, an evaluation of the angular velocity can indicate whether it is possible to track the object within the angular velocities of the pan-tilt structure. When the object is no longer visible, the camera will restart with the *scanning mode*.

It will be possible to set the camera for an *on demand mode* so that the camera will point in a certain direction at a given time to track a determined object. It will be possible to uplink the data and the times of the *on demand mode* from the ground. The camera can operate in an open loop control, with a set of time-tagged pre-computed instructions, or once the object is in the

FOV, the camera could switch in a closed loop control, like in the *tracking mode*.

The Fig. 7.7 shows the output of a star tracker mounted on Mango of the PRISMA mission, see [29]. PRISMA is a Research and Development (RD) program founded by the Swedish National Space Board. The mission is a rendezvous and formation flying technology test bed with two satellites: Mango, which is the main satellite, and Tango, see Fig. 7.6.

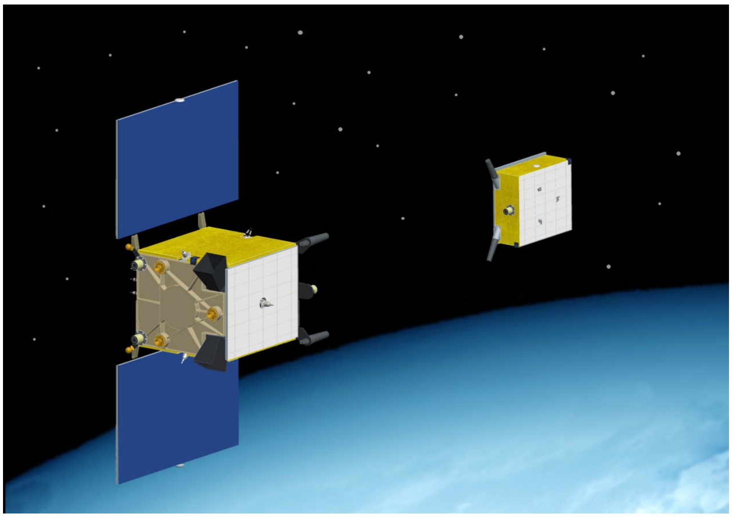


Figure 7.6: The PRISMA mission

The yellow dots in Fig. 7.7 represent the centroids of the stars in the field of view. Some of the stars are present only in few frames because they are near to the sensibility threshold of the camera sensor; therefore not every star is visible in each frame. The camera could also be used as a star tracker to determine the rotation matrix between an inertial frame and the camera frame.

In the images it is possible to detect some points moving differently with respect to the background. For each frame it is possible to subtract a stellar map from the on board data-base, if an object is in the frame but not in the stellar database a possible target has been detected. The green triangles have been depicted just to have a reference for the target pass, the star background is slowly moving because the satellite is not pointing in inertial mode. The frames are 8 seconds spaced. This pass lasts about 40 seconds, this is a typical pass for a LEO observation from LEO; the relative angular velocities are very high not allowing longer period without the tracking of the object.

The results of this first operation are the RA and the Dec of the satellite pass. Classical methods are not useful for such short passes, so a different

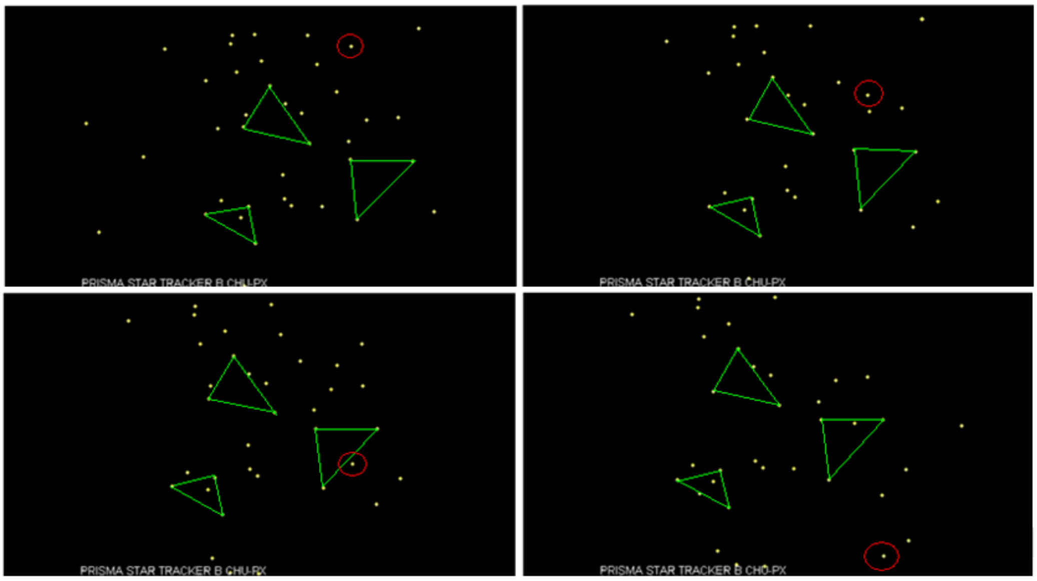

PRISMA STAR TRACKER B CHU-PX
PRISMA STAR TRACKER B CHU-PX
PRISMA STAR TRACKER B CHU-PX
PRISMA STAR TRACKER B CHU-PX

Figure 7.7: A moving object over the star background

algorithm to estimate the orbital parameters is needed. A genetic algorithm will be used to obtain an IOD (Initial Orbit Determination) by using only observation angles. The observed object will be treated as a star, yielding the vector between observer and target in inertial coordinates.

Chapter 8

Conclusions and future work

Massimo segno della fine, è il principio

- Carlo Dossi, *Note azzurre, 1912*

In this final chapter the main conclusions are issued and the possible future work is described. The performances of the algorithm are highlighted and proved through the presented test case. A very accurate location of the observer has been performed with a realistic and complex model. Future works are related to further tests and to the implementation of this algorithm on an automatic telescope with a searching strategy.

8.1 Conclusions

The problem of IOD is becoming more and more important since the number of objects in orbit is still increasing. The high number of satellites makes very likely that more collisions will occur in the next future.

The increasing resolution of the telescopes, also the amateur ones, allows the discovery of a great amount of small objects. For many objects the observation lasts only few seconds, so it is very important to obtain reliable results also with TSA observations.

In this thesis a new method for IOD has been presented. This method, based on a genetic algorithm, allows a first computation of the orbital parameters of the observed objects.

The genetic algorithm moves the candidate solutions in a two-dimensional space; it shows very good performances and absolute reliability.

A correct tuning of the parameters can decrease the computational time and yield better perfomances.

A test of this method has been presented, showing that very good results can be obtained also in one of the most difficult cases: the colocated satellites in GEO.

A very precise location of the observer in an inertial frame is needed. In this thesis a very detailed model has been taken into account to include also the little effects associated to the Earth dynamics: as the polar motion or the ocean tides.

Finally a possible mission is proposed with two different configurations: one regarding a satellite with an optical telescope as main payload, and the other one regarding a payload that is possible to mount on the ISS.

8.2 Future work

The future work can be addressed to make more tests for several targets. Unfortunately, very few images have been available to test the algorithm; the very demanding requirements for the astrometric calibration has made this operation more complicated, but still feasible.

A more detailed analysis of the error of the Two Line Elements can be useful to highlight the performances of the IOD. To obtain an independent full orbit determination, an operation of linkage between two TSA is needed to compare the results with the TLE.

Another work could be based on the evaluation of the error of the estimated orbit. A covariance analysis coul be useful to correlate properly the data with the NORAD database.

An automatic extraction of the angular measurements from the streaks could be very important in the case of an automatic telescope. This instrument could observe the sky all night long with a searching strategy, allowing also the real time tracking and the automatic prevision of the next pass.

Appendix A

The pseudo-random numbers

Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin.

- John von Neumann, *Various techniques used in connection with random digits, 1951*

This chapter deals with the problem of generating random number needed for the genetic algorithm. True random generators can be very complex, so pseudo-random generators are used. Different algotihms and methods are presented. The main focus is on the Matlab generators. In many applications using a pseudo-random generator does not affect the results.

A.1 Random number generation

Random numbers have a large set of important applications. Random numbers give the foundation of stochastic methods and related heuristic optimization techniques. Randomized algorithms for a big amount of problems are revolutionizing several fields and establishing randomization as one of the fundamental ideas of computer science. Unfortunately, it is not easy to generate random numbers. Indeed, it is fundamentally impossible to produce truly random numbers on any deterministic device, see [83]. Another option is given by the pseudorandom numbers, a stream of numbers that appear as if they were generated randomly; i.e. their probability density functions are very similar. It is very important to choose the correct method to generate random-numbers. In one famous case, a Web browsers encryption scheme was broken with the discovery that the seeds of its random-number generator employed too few random bits [31]. Simulation accuracy is regularly compromised or invalidated by poor random number generation, see [71]. How long could be a Monte Carlo simulation depends on many factors. Usually it is not efficient to run a single Monte Carlo with a single seed because of the period of the pseudo-random generation algorithm. It could be more useful to run short Monte Carlo (100 runs) varying the seed.

A.1.1 Actual Random generation

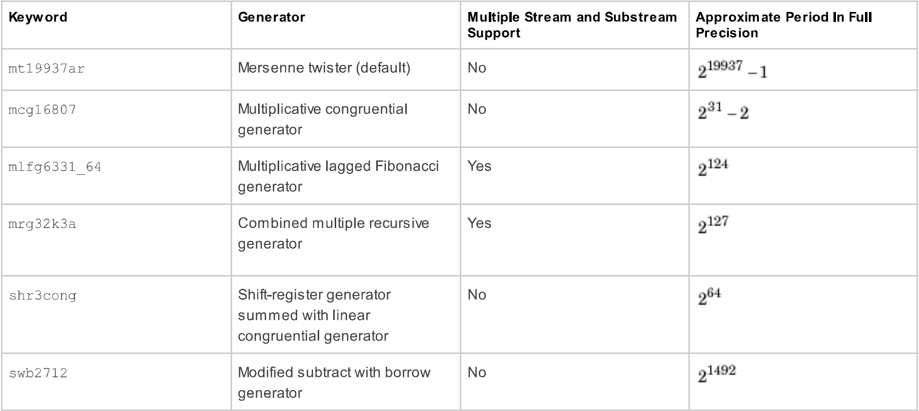
An actual Random Number Generator (RNG) can be applied only with an actual random system. Some mechanical systems have been used in the past to simulate random behaviour, like dice or coin flipping. Generating a long set of random data is very expensive for mechanical devices; a sufficient amount of random number usually is needed to be generated to assure a valid statistics. A physical random number generator can be based on random atomic or subatomic properties in quatum mechanics. Sources of entropy can include thermal noise, clock drift, radio noise et cetera. In any case each of these methods could lead to some errors because of asymmetries or systematic biases in generation or even in the extraction phase. Tables of real random numbers have been published for the users that could use them, see [45]. Humans are obviously not a good source of random numbers.

A.2 Pseudo random data generation

A controlled generation of random numbers is made by algorithms that can reproduce random numbers. The sequence of numbers is not *really* random because a seed is needed for the initialization of the algorithm, and all the produced numbers are sequentially obtained from that seed in an algorithmic way.

- Like random noise, the local sequence has a very low correlation with any other sequence in the set, or with the same sequence at a significantly different time offset
- Unlike random noise, it must be easy to generate exactly the same sequence

Matlab can handle various methods or generating Pseudo Random Noise Generator (PRNG), see Fig. A.1



Keyword	Generator	Multiple Stream and Substream Support	Approximate Period In Full Precision
mt19937ar	Mersenne twister (default)	No	$2^{19937}-1$
mcg16807	Multiplicative congruential generator	No	$2^{31}-2$
mlfg6331_64	Multiplicative lagged Fibonacci generator	Yes	2^{124}
mrg32k3a	Combined multiple recursive generator	Yes	2^{127}
shr3cong	Shift-register generator summed with linear congruential generator	No	2^{64}
swb2712	Modified subtract with borrow generator	No	2^{1492}

Figure A.1: The Matlab PNRG

Pseudo random numbers are very important and useful thanks to the speed in generation and also to the reproducibility; the latter peculiarity could seem very unuseful intuitively, but demonstrates its power simulating large MC tests. If, for some reasons, one of the run of the MC does not seem to show good results, with the reproducibility it is possible to obtain exactly the same numbers to investigate why the algorithm has failed; this approach is of course impossible with true random numbers (unless all the data are saved). So, for MC test purposes, PRNG are very essential. In the next paragraphs the most known algorithms are explicated.

A.2.1 The linear congruential generator

The Linear Congruential Generator (LCG) is one of the most famous algorithm that yields a sequence of randomized numbers calculated with a linear equation. This method is one of the oldest and best-known pseudorandom number generator algorithms. The generator has a recurrence relation:

$$X_{n+1} \equiv (aX_n + c) \pmod{m} \tag{A.1}$$

Where X_n is the current value, m is the modulo, a the multiplier, c the increment; X_0 is the seed. When $c = 0$ the LCG becomes the Multiplicative Congruential Generator (MCG).

Let see how the modulus is applied: for a positive integer m, two integers a and b are said to be congruent modulo m, written:

$$a \equiv b \pmod{m} \tag{A.2}$$

if the difference $(a - b)$ is an integer multiple of m, or m divides $(a - b)$. The number m is called the modulus of the congruence.

A.2.2 The Lagging Fibonacci Generator

Another PRNG is the Lagging Fibonacci Generator (LFG), with this recurrence relation:

$$X_{n+1} = X_n \star X_{n-1} \tag{A.3}$$

In this case, the new term is the combination of the two past terms, the symbol $\star$ denotes a general binary operation. In case the operation is the multiplication, the Multiplicative LFG (MFLG) is obtained. If the XOR operation is used, the LFG is known as Generalized Feedback Shift Register (GFSR), see Fig. A.2.

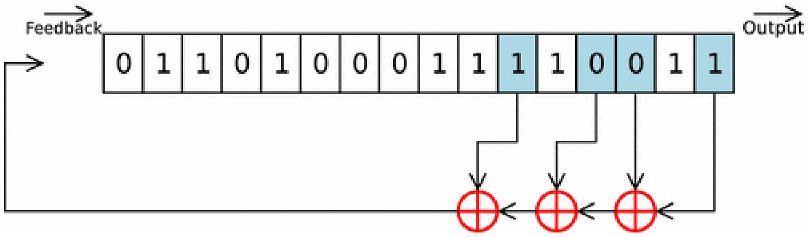


Figure A.2: The shift register

The initialization of LFG is a very complex problem. The output of LFG is very sensitive to initial conditions, and statistical defects may appear initially, but also periodically, in the output sequence.

A.2.3 The Mersenne twister algorithm

The Mersenne Twister Algorithm (MTA) is a pseudo random number generator developed in 1997 by Makoto Matsumoto [55]. The Mersenne Twister has been optimized for use with MC simulations in a number of fields, including simulating complex biochemical pathways, photon migration, genome coalescence, cellular biology, and computational finance.

- It has a very long period of 219937-1. While a long period is not a guarantee of quality in a random number generator, short periods (such as the 232 common in many software packages) can be problematic
- It is k-distributed to 32-bit accuracy for every $1 \leq k \leq 623$
- It passes numerous tests for statistical randomness, including the Diehard tests. It passes most, but not all, of the even more stringent TestU01 Crush randomness tests

Currently it is the default PRNG for Matlab, altough it is not adapt for cryptography because, if a sequence of a sufficient number of iterations is observed, it is possible to predict the next numbers. This method is based on a generalized form of a twisted GFSR. Each value is created using two 32-bit integers from the generator; the possible values are all multiples of strictly within the interval (0,1). A typical output in the plane $x - y$ is shown in Fig. A.3. The *randn* algorithm used by default for Matlab random streams is the ziggurat algorithm [50], but with the MTA generator underneath.

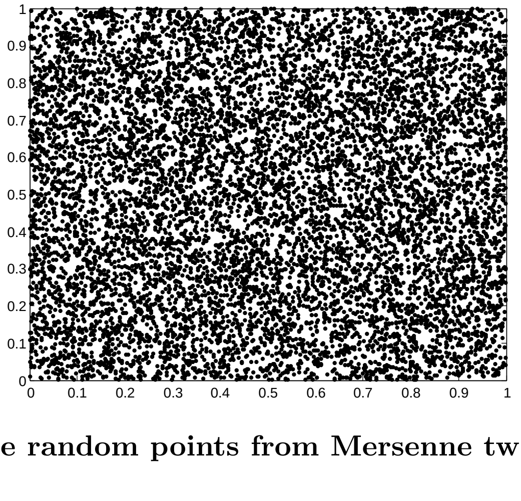


Figure A.3: The random points from Mersenne twister algorithm

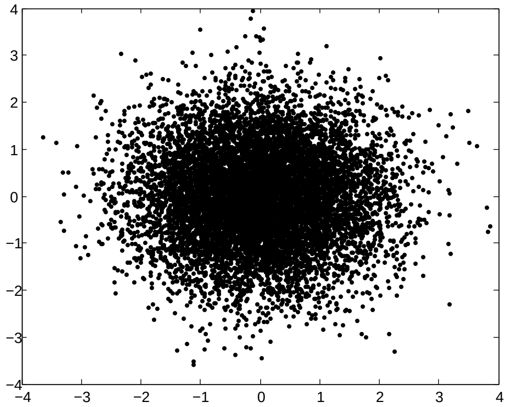


Figure A.4: The random normalized points from Mersenne twister algorithm

A.2.4 The cryptographic methods

The cryptographic methods need to be very accurate, but very unuseful for simulation and MC purposes because of their very slowness. The two most known methods are: the Blum Blum Shub (BBS) and the Fortuna methods.

TheBBS method uses this formulation [9]:

$$X_{n+1} = X_n^2 \pmod{m} \tag{A.4}$$

where $m = pq$ is the product of two large primes p and q. The output is commonly either the bit parity of X_{n+1} or one or more of the least significant bits of X_{n+1}. The seed X_0 should be an integer that is co-prime to m (i.e. p and q are not factors of X_0) and different from 1 and 0.

One interesting property of the BBS method is that it is possible to obtain any X_n knowing the seed and the steps from the seed, see Eq. (A.5):

$$X_n = \left(X_0^{2^i mod \lambda(m)}\right) \pmod{m} \tag{A.5}$$

where λ is the Carmichael function. The BBS is one of the algorithms used for cryptography; the Fortuna algorithm is used as well [23]. Its main characteristics are:

- The generator itself, which once seeded will produce an indefinite quantity of pseudo-random data
- The entropy accumulator, which collects genuinely random data from various sources and uses it to reseed the generator when enough new randomness has arrived

- The seed file, which stores enough state to enable the computer to start generating random numbers as soon as it has booted

A.2.5 Testing the pseudorandom

Some tests are used to check if the numbers generated have the features of real random numbers. An early famous work for these studies is [41].

Generally, there are various kind of test:

- The frequency test: make sure that there were roughly the same number of 0s, 1s, 2s, 3s, etc.
- The serial test, did the same thing but for sequences of two digits at a time (00, 01, 02, etc.)
- The poker test, tested for certain sequences of five numbers at a time (aaaaa, aaaab, aaabb, etc.)
- The gap test, looked at the distances between zeros (00 would be a distance of 0, 02250 would be a distance of 3, etc.)

Very specific tests are carried out especially for secure cryptographic methods, see [66].

A.3 Fields of application

The PRNG are used especially for three main applications:

- Monte Carlo simulations

  A MC simulation needs two assumptions to not need infinite pseudo random number and infinite simulations: *optimism* and *utilitarianism*, see [34]. *Optimism* means that producing hopefully few test will give a complete description from the statistical point of view. *Utilitarianism* means that the used pseudo random numbers are supposed to yield a reliable answer.

- Cryptography

  Randomization is a key feature in cryptography. The main concern regards the security, one of the main approach is in the Kerckhoffs's principle [43]: *A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.* There is no better key than a random key, because it is not guessable.

- Games

  Randomization is used in gambling. Many electrion casino games use PRNG, including the slot machines. One more interesting use is in the *procedural generation*, when memory is limited it is more useful to calculate something rather than storing it, or referring to generate something with an algorihtm rather than manually. This is used for example in Perlin Noise [63], where the material textures are computed using random noise to simulate real surfaces or the generation of landscapes when rendering, see Fig. A.5.

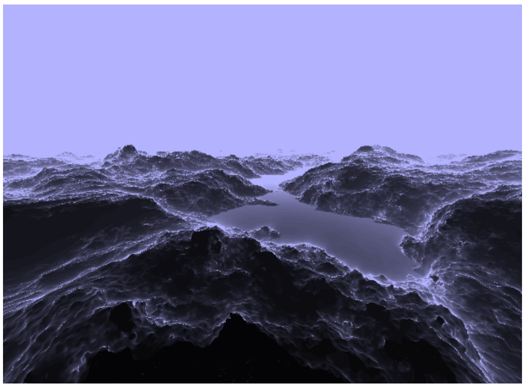


Figure A.5: A landscape created with Perlin noise

Appendix B

Main used solvers

The quest for certainty, blocks the search for meaning. Uncertainty is the very condition to impel man to unfold his powers.

- Erich Fromm, *Man for Himself, 1947*

In this appendix the main solvers used in this work are briefly explained and reported. The genetic algorithm finds the boundary conditions for the Lambert's problem, once it has been solved we can propagate the initial condition of the found orbit to the final time of the observation period. We need to find two accurate and reliable solvers to these two problems. Two iterative algorithms, that have been extensively tested, are described in this appendix..

B.1 The Lambert's Problem Solver

The description of this algorithm can be found at: `http://www.esa.int/gsp/ACT/inf/op/globopt.htm`.

This routine implements a new algorithm that solves the Lambert's problem. The algorithm has two major characteristics that makes it favorable to other existing ones.

B.1.1 Step 1

This algorithm describes the generic orbit solution of the boundary condition problem through the variable:

$$X = \log(1 + \cos(\alpha/2)) \tag{B.1}$$

where α is the transfer angle.

By doing so the graph of the time of flight become defined in the entire real axis and resembles a straight line. Convergence is granted within few iterations for all the possible geometries (except, of course, when the transfer angle is zero). When multiple revolutions are considered the variable is:

$$X = \tan(\cos(\alpha/2) * \pi/2) \tag{B.2}$$

B.1.2 Step 2

Once the orbit has been determined in the plane, this routine evaluates the velocity vectors at the two points in a way that is not singular for the transfer angle approaching to π (Lagrange coefficient based methods are numerically not well suited for this purpose).

As a result Lambert's problem is solved (with multiple revolutions being accounted for) with the same computational effort for all possible geometries. The case of near 180 transfers is also solved efficiently.

We note here that even when the transfer angle is exactly equal to π the algorithm does solve the problem in the plane (it finds X), but it is not able to evaluate the plane in which the orbit lies. A solution to this would be to provide the direction of the plane containing the transfer orbit from outside. This has not been implemented in this routine since such a direction would depend on which application the transfer is going to be used in.

By default, the short-way solution is computed. The long way solution may be requested by giving a negative value to the corresponding TOF.

For problems with $m \neq 0$ there are generally two solutions. By default, the right branch solution will be returned. The left branch may be requested by giving a negative value to the corresponding number of complete revolutions m.

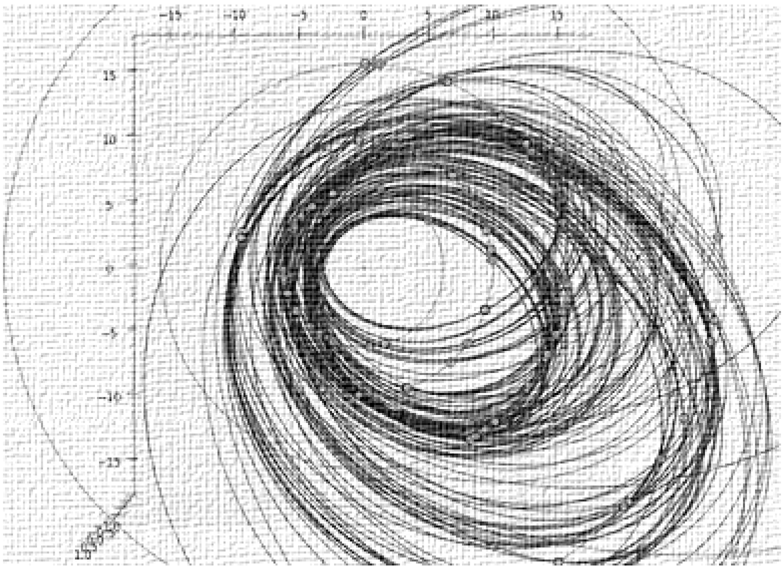


Figure B.1: Laurelin, a trajectory from the GTOC competition

This routine is very reliable for very short TOF as in our case; altough this routine was intended to deal with multiple revolutions.

B.2 The Kepler's Problem Solver

The Kepler problem has been solved with the universal variable formulation, see [5].

This approach is based on a fundamental theorem:

if A, B and C are coplanar vectors, and A and B are not colinear, it is possible to express C as a linear combination of A and B

Since for a keplerian orbit the position vector and the velocity vector define the orbital plane, it is possible to express position and velocity at any time using a linear combination of the initial conditions.

$$\vec{r} = f\vec{r}_0 + g\vec{v}_0 \tag{B.3}$$

This expression is also known as f and g expansion.

The velocity can be easily derived, notice that f and g are not constants.

$$\vec{v} = \dot{f}\vec{r}_0 + \dot{g}\vec{v}_0 \tag{B.4}$$

f, g, $\dot{f}$ and $\dot{g}$ are not independent, so we need to find just three parameters, the fourth can be derived from this identity:

$$1 = f\dot{g} - \dot{f}g \tag{B.5}$$

This method works for all conic orbits, showing a very good convergence.

f and g expressions are dependent from the x variable that is defined as:

$$\dot{x} = \frac{\sqrt{\mu}}{r} \tag{B.6}$$

The solution equation for x is transcendental, so usually a good approximation can be obtained using the Newton iteration algorithm.

Bibliography

[1] Abdelkhalik, O., Mortari, D. & Junkins, J.L., *Space Surveillance with Star Trackers. Part II: Orbit Estimation*, Paper AAS 06-232 of the 16th AAS/AIAA Space Flight Mechanics Meeting, January 22-26, 2006, Tampa, Florida.

[2] Ansalone, L. & Curti, F., *A genetic algorithm for Initial Orbit Determination from a too short arc optical observation*, Advances in Space Research, Volume 52, Issue 3, p. 477-489, DOI: 10.1016/j.asr.2013.04.004.

[3] Ansalone, L. & Curti, F., *Space Debris Orbit Determination from an ISS Onboard Camera*, AAS 12-714, p. 203-211, Volume 114 of the Science and Technology Series, Results and Opportunities - The Decade of Utilization, ISSN 0278-4017.

[4] Ansalone, L., Cappelletti, C., Curti, F. et al., *Study on debris detection, identification and orbit reconstruction using ground and space based telescopes*, IAC-11.A6.6.6, Vol.3, p. 2317-2322, Proceedings of the 62nd International Astronautical Congress 2011, 3-7 October 2011, Cape Town, South Africa.

[5] Bate, R.R., Mueller, D.D. & White, J.E., *Fundamentals of Astrodynamics*, Dover, New York, 1971, pp. 20, 25-33, 212-222, 396-412.

[6] Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised ed. Reston, VA: AIAA, 1999.

[7] Biggs, N.L., *Discrete mathematics*, Oxford University Press, 2002.

[8] Bizouard, C. & Gambis, D., *The combined solution C04 for Earth Orientation Parameters consistent with International Terrestrial Reference Frame 2005*, IAG Symp, vol 134, pp 265-270, doi : $10.1007/978-3-642-00860-3_41$.

[9] Blum, L., Blum, M., Shub, M., *A Simple Unpredictable Pseudo-Random Number Generator*, SIAM Journal on Computing 15 (2): 364383.

[10] Bock, H., Jaggi, A., Svehla, D., et al., *Precise orbit determination for the GOCE satellite using GPS*, Adv. Space Res., 39, 1638-1647, 2007.

[11] Bondy, J.A. & Murty, U.S.R., *Grapf theory with applications*, Springer, Graduate Texts in Mathematics, Vol. 244, first published in 1976.

[12] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., *Introduction to Algorithms*, MIT press, first edition 1990, ISBN: 978-0-262-03384-8.

[13] Dam, J.K., Schafer, W., et al., *Simulation of Servo Loops in Atomic Clock Ensemble in Space*, EFTF 2010 - European Frequency and Time Forum, 13-16 April 2010.

[14] DeMars, K.J., Jah, M.K. & Schumacher P.W.Jr., *Initial Orbit Determination using Short-Arc Angle and Angle Rate data*, IEEE Transactions on Aerospace and Electronic Systems , Vol. 48, 2628-2637, No. 3 July 2012.

[15] Dettmann, J. & Gianfiglio, G., *EuTEF The European Technology Exposure Facility*, On Station, No.14 September 2003.

[16] Di Roberto, R., Ansalone, L., Cappelletti, C., *UNISAT-5 a microsatellite for space debris monitoring*, IAC-12.A6.1.20, Vol.3, p. 2301-2307, Proceedings of the 63nd International Astronautical Congress 2013, 1-5 October 2011, Naples, Italy.

[17] Dorigo, M. & Stutzle, T., *Ant Colony Optimization*, A Bradford Book, The MIT Press 2004, ISBN 0-262-04219-3.

[18] Eckstein, M.C., Rajasingh, C.K., Blumer, P., *Colocation Strategy and Collision Avoidance for the Geostationary Satellites at 19 Degrees West*, International Symposium on Space Flight Dynamics, Toulouse 1989.

[19] Eiben, A.E. & Smith, J.E., *Introduction to Evolutionary Computing*, 2003 Springer, ISBN 978-3-540-40184-1.

[20] Escobal, P.R., *Methods of Orbit Determination*, John Wiley and Sons, New York, 1965.

[21] Ettouati, I., Mortari, D. & Pollock, T., *Space Surveillance with Star Trackers. Part I: Simulations*, Paper AAS 06-231 of the 16th AAS/AIAA Space Flight Mechanics Meeting, January 22-26, 2006, Tampa, Florida.

[22] Farnocchia, D., Tommei, G., Milani, et al., *Innovative methods of correlation and orbit determination for space debris*, Celestial Mechanics and Dynamical Astronomy, Volume 107, Number 1-2, pp. 169-185 (2010).

[23] Ferguson, N. & Schneier, B., *Practical Cryptography* Wiley and Sons, 2003. ISBN 0-471-22357-3.

[24] Flohrer, T., Peltonen, J., Kramer, A., Eronen, T., Kuusela, J., Riihonen, E., Schildknecht, T., Stveken, E., Valtonen, E., Wokke, F. & Flury, W., *Space-Based Optical Observations of Space Debris*, Proceedings of the Fourth European Conference on Space Debris, p. 165 Darmstadt, Germany, 18-20 April 2005.

[25] Flohrer, T., Schildknecht, T., Musci, R., et al, *Performance estimation for GEO space surveillance*, Advances in Space Research 35 (2005) 1226-1235

[26] Flohrer, T., Krag, H., Klinkrad, H. et al., *Feasibility of performing space surveillance tasks with a proposed space-based optical architecture*, Advances in Space Research 47 (2011) 1029-1042.

[27] Fruh, C., & Schildknecht, T., *Accuracy of Two-Line-Element Data for Geostationary and High-Eccentricity Orbits*, Journal of Guidance, Control, and Dynamics, Vol. 35, No. 5 (2012), pp. 1483-1491, doi: 10.2514/1.55843.

[28] Fujimoto, K. & Scheeres, D.J., *Short-arc correlation and initial orbit determination for space-based observations*, Proceedings of the 2011 Advanced Maui Optical and Space Surveillance Technologies Conference, p. 55, September 2011.

[29] Gill, E., D'amico, S. & Montenbruck, O., *Autonomous Formation Flying for the PRISMA Mission*, Journal of Spacecraft and Rockets, Vol.44, No.3, May-June 2007.

[30] Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

[31] Goldberg, I., & Wagner, D., *Randomness and the Netscape browser*, Dr. Dobbs Journal, pages 6670, 1996.

[32] Gooding, R.H., *A procedure for the solution of Lambert's orbital boundary-value problem*, Celestial Mechanics and Dynamical Astronomy, 48:145-165,1990.

[33] Gooding, R.H., *A new procedure for the solution for the classical problem of minimal orbit determination from three lines of sight*, Celestial Mechanics and Dynamical Astronomy, vol. 66, no. 1, pp. 387423, 1997.

[34] Hammersley, J.M. & Handscomb, D.C., *Monte Carlo Methods* Monograph on Statistics and Applied Probability 3, Publisher: Chapman & Hall, 1992.

[35] Ho, Y.C. & Pepyne, D.L., *Simple Explanation of the No-Free-Lunch Theorem and Its Implications*, Journal of Optimization Theory and Applications 115, 549-570.

[36] Hog, E., Fabricius, C., Makarov, V.V., et al., *The Tycho-2 Catalogue of the 2.5 Million Brightest Stars*, 2000, Astronomy & Astrophysics , 355, L27.

[37] Hoots, R.R. & Roehrich, R.L., *Spacetrack Report No. 3: Models for propagation of NORAD element sets*, Aerospace Defense Center, Peterson Air Force Base, 1980.

[38] Izzo, D., Becerra, V.M., Myatt, et al., *Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories*, Journal of Global Optimization (2007) 38:283-296.

[39] Jehn, R., *Debris Detection and Observation Systems*, International Interdisciplinary Congress on Space Debris, McGill University, Montreal, 7-9 May 2009.

[40] Kaya, D., & Snow, D., *Short arc initial orbit determination using angles-only space-based observations*, Proceedings AAS/AIAA Astrodynamics Specialists Conference, pp. 29, Durango, CO, Aug 19-22, 1991.

[41] Kendall, M.G. $ Babington Smith B., *Randomness and Random Sampling Numbers*, Journal of the Royal Statistical Society 101:1 (1938), 147-166.

[42] Kennedy, J., & Eberhart, R., *Particle Swarm Optimization*, Proc. IEEE International Conf. on Neural Networks (Perth, Australia), 1995.

[43] Kerckhoffs, A., *La cryptographie militaire*, Journal des sciences militaires, vol. IX, pp. 583, January 1883, pp. 161191, February 1883.

[44] Kessler, D.J., Cour-Palais B.G., *Collision Frequency of Artificial Satellites: The Creation of a Debris Belt*, Journal of Geophysical Research, Vol. 83, No. A6, June 1, 1978.

[45] Kolmogorov, A.N., *On table of random numbers*, The Indian Journal of Statistics, Series A (1961-2002), Vol. 25, No. 4 (Dec., 1963).

[46] Lancaster, E.R. & Blanchard, R.C., *A unified form of Lambert's theorem*, NASA technical note TN D-5368,1969.

[47] Lane, M.T., *A numerical approach to the angles-only initial orbit determination problem*, Proceedings of the AAS/AIAA Astrodynamics Conference, Durango, CO, Aug. 19-22, 1991. Pt. 1 (A92-43251 18-13). San Diego, CA, Univelt, Inc., 1992, p. 55-71.

[48] Lenz, S.M., Bock, H.G. & Schlder, J.P., *Multiple Shooting Method for Initial Satellite Orbit Determination*, Journal of Guidance, Control, and Dynamics, Vol. 33, No. 5, 1334-1346, September-October 2010.

[49] Li, D. & Yi, D., *Initial Orbit Determination of Space Object Using Space-Based Optical Observations*, Presented at the 2012 International Conference on Computer Science and Electronics Engineering, Vol. 1, 128-132.

[50] Marsaglia G., & Tsang, W.W., *The ziggurat method for generating random variables*, Journal of Statistical Software, 5:17. 2000.

[51] Marti, R. & Reinelt, G., *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization* Springer 2011, ISBN 978-3-642-16729-4.

[52] Martinez Fadrique, F., Agueda Mat, A., Jorgera Grau, J., Fernandez Sanchez, J. & Aivar Garcia, L., *Comparison of Angles Only Initial Orbit Determination Algorithms for Space Debris Cataloguingi*, Presented at the 22nd International Symposium on Space Flight Dynamics, Sao Jose dos Campos, Brazil, n.1, pp 39-51 28 Feb-4 Mar 2011.

[53] Maruskin, J.M., Scheeres, D.J., Alfriend, K.T., *Correlation of Optical Observations of Objects in Earth Orbiti*, Journal of Guidance, Control, and Dynamics, Vol. 32, pp 194-209, No. 1, January-February 2009.

[54] Maskell, P., Oram, P., *Sapphire: Canadas answer to space-based surveillance of orbital objectsi*, Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, p.E5, September 16-19 2008, Wailea Beach, Maui, United States, 2008.

[55] Matsumoto, M., Nishimura, T., *Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator*, Transactions on Modeling and Computer Simulation, Vol. 8, No. 1, January 1998, Pages 3-30.

[56] Milani, A., Tommei, G., Farnocchia, D., et al., *Correlation and orbit determination of space objects based on sparse optical data*, Mon. Not. R. Astron. Soc. 417, 2094-2103 (2011).

[57] Milani, A., Gronchi, G.F., De' Michieli, M., et al, *Orbit determination with very short arcs. I admissible regions*, Celestial Mechanics and Dynamical Astronomy, September 2004, Volume 90, Issue 1-2, pp 57-85.

[58] Milani, A., Gronchi, G.F:, Knezevic, Z., et al, *Orbit determination with very short arcs: II. Identifications*, Icarus, Volume 179, Issue 2, 15 December 2005, Pages 350-374.

[59] Milani, A., Chesley,S.R, *Virtual Impactors: Search and Destroy*, Icarus, 145, 12-24, (2000).

[60] Mitchell, M., *An Introduction to Genetic Algorithms*, A Bradford Book The MIT Press, fifth editiion 1999, ISBN 0-262-13316-4.

[61] Orr, A., *ESAs Atmosphere-Space Interactions Monitor (ASIM) for the ISS*, 1st TEA - IS Summer School, June 17th 2012, Malaga, Spain.

[62] Oswald, M., Wiedemann, C., Wegener, P., et al., *Space-Based Radars for the Observation of Orbital Debris in GEOi*, AIAA Space Conference 2003, Long Beach, CA, pp. 1-10, AIAA 2003-6294, September 23-25 2003.

[63] Perlin, K., *Improving noise*, ACM SIGGRAPH, 681-682.

[64] Raymond, C., *Nature-Inspired Algorithm for Optimisation*, Studies in Computational Intelligence, Vol. 193, 2009, ISBN 978-3-642-00267-0.

[65] Rodgers, J.L., Nicewander, W.A., *Thirteen ways to look at the Correlation Coefficienti*, The American Statistician, Vol. 42, No. 1. (Feb., 1988), pp. 59-66.

[66] Rukhin, A., Soto, J., Nechvatal, J., et al., *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, NIST, Special Publication 800-22, Revision 1a.

[67] Russell, S.J. & Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, NJ, second edition, 2002.

[68] Rutkowski, L., *Computational Intelligence, methods and techniques*, Springer 2008, ISBN: 978-3-540-76287-4.

[69] Sela, A.F., Michel, A., *The SOLAR Mission Tool - an integrated ops tool*, Proceedings SpaceOps 2012, 11-15 June, Stockholm.

[70] Sharma, J., *Space-Based Visible Space Surveillance Performance*, Journal of Guidance, Control, and Dynamics, Vol. 23, No. 1, JanuaryFebruary 2000, pp 153-158.

[71] Skiena, S.S., *The Algorithm Design Manual*, November 14, 1997, ISBN-10:0387948600

[72] Snow, D., & Kaya, D., *Short arc orbit determination using angles-only space-based observationsi*, Proceedings AAS/AIAA Spaceflight Mechanics Meeting, pp. 57-74, Colorado Springs, CO, Feb 24-26, 1992.

[73] Socha, K. & Dorigo, M., *Ant colony optimization for continuous domains*, European Journal of Operational Research 185 (2008) 1155-1173.

[74] Tapley, B.D., Schutz, B.E. & Born, G.H., *Statistical Orbit Determination*, Elsevier 2004, ISBN-10: 0126836302.

[75] *Technical Report on Space Debris: Text of the report adopted by the Scientific and Technical Subcommittee of the United Nations Committee on the Peaceful Uses of Outer Space*, United Nations, New York, 1999.

[76] Tommei, G., Milani, A., & Rossi, A., *Orbit Determination of Space Debris: Admissible Regionsi*, Celestial Mechanics and Dynamical Astronomy, Vol. 97, No. 4, 2007, pp. 289-304.

[77] Turon, C., Gomez, A., Crifo, F., et al., *The HIPPARCOS Input Catalogue. I - Star selection*, Astronomy and Astrophysics (ISSN 0004-6361), vol. 258, no. 1, p. 74-81.

[78] Vallado, D.A., Carter, S.S., *Accurate Orbit Determination from Short-Arc Dense Observational Data*, The Journal of the Astronautical Sciences, Vol. 46, No. 2, AprilJune 1998, pp. 195213.

[79] Vallado, D.A., *Fundamentals of Astrodynamics and Applications, Second Editioni*, Microcosm Press, El Segundo California, 2001.

[80] Vallado, D.A., *Evaluating Gooding Angles-only Orbit Determination of Space Based Space Surveillance Measurements*, Proceedings of the AAS George Born Astrodynamics Symposium, pp 1-21, Boulder, CO, May 2010.

[81] Van den Ijssel, J., Visser, P., Patino Rodrigues, E., *CHAMP Precise Orbit Determination using GPS data*, Adv. Space Res. 31, 8, 1889-1895, 2003.

[82] Vetter, J.R., *Fifty Years of Orbit Determination: Development of Modern Astrodynamics Methods*, Johns Hopkins Apl Technical Digest, 252 Vol. 27, N. 3 2007.

[83] Von Neumann, J., *Various techniques used in connection with random digits*, National Bureau of Standards Applied Mathematics Series, 12: 36-38.

[84] Wolpert, D.H. & Macready, W.G., *No Free Lunch Theorems for Optimization*, IEEE Transactions on Evolutionary Computation 1, 67, 1997.

[85] Yang, X.S., *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, second edition 2010, ISBN-10:1-905986-28-9.

[86] Yi, T., MeiPing, W., & XiaoFeng, F., *A method of correlation analysis for space-based GEO object surveillancei*, Science China Technological Sciences, June 2012 Vol.55 No.6: 1749-1756.