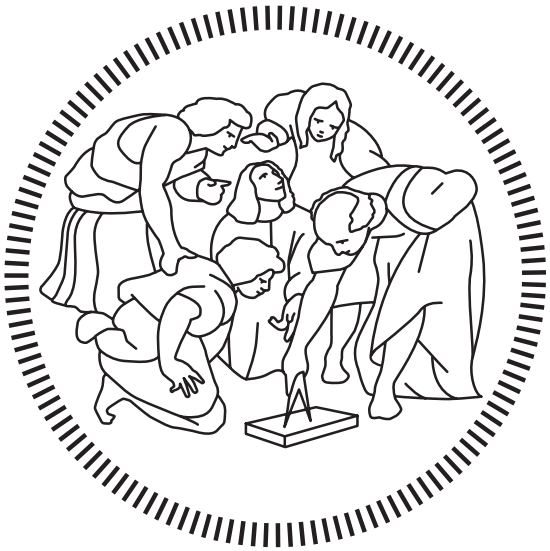
Politecnico di Milano

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science – Space Engineering



Multiple Hypothesis Tracking Method for Autonomous Orbit Determination of a Nanosatellite Swarm

Supervisor
Francesco TOPPUTO

Co-Supervisor
Massimiliano VASILE

Candidate
Andrea LAUS – 920660

Academic Year 2019 – 2020

Acknowledgements

Firstly, I would like to express my gratitude to my family for their support during these years of studies. They have always given me the chance to give my best.

A huge thanks to the friends I met during this journey, in particular to those who have always been there for me.

Thanks to the people who lived with me inside the campus. We have had so much fun together.

Thanks to my classmates who turned even the most annoying lectures into some laughs. I could not ask for more.

Abstract

In the past 20 years space companies and researchers from all over the world have dedicated an increasing effort in studying nanosatellites. They are small satellites weighting less than 10 kg and they are equipped with off-the-shelf components, characterised by a significantly lower cost with respect to components built for a specific mission. Commercial innovation pushed the development and testing of microelectronics for space applications, helping to tear the cost barrier down and to open the doors of space missions to a wider number of companies. Furthermore, their capability of being launched as piggyback opens up affordable launch opportunities.
The drawback of nanosatellites is the limited on-board space that can be allocated to perform science. A possible solution to this drawback is formation flying satellites: multiple satellites with either the same or different individual role, working together to accomplish mission objectives.
Such satellite formation is the object of this thesis. It is composed by a mother satellite and a swarm of nanosatellites flying in the proximity of a binary asteroid system. This thesis proposes the Multiple Hypothesis Tracking as a method to autonomously determine satellites' position and velocity in the case of closely-spaced nanosatellites providing range measurements only.
Even if measurements are not sufficient to analytically compute nanosatellites' state, this method is capable of estimating their full state even when the nanosatellites are so close to each other that their state uncertainty makes their trajectories overlap.

Sommario

Nel corso degli ultimi 20 anni, agenzie spaziali e ricercatori di tutto il mondo hanno dedicato una sempre crescente attenzione allo studio dei nanosatelliti. I nanosatelliti sono satelliti di dimensioni ridotte, con peso inferiore ai 10 kg ed equipaggiati con componenti prodotte in serie, perciò caratterizzate da un costo inferiore rispetto a quelle sviluppate per una specifica missione. Lo sviluppo della tecnologia ha contribuito all creazione di componenti microelettroniche adatte anche ad applicazioni in ambito spaziale. Questo ha contribuito ad abbattere il problema del costo complessivo di una missione e ad aprire le porte dello spazio ad un maggior numero di compagnie. In più, grazie alle loro ridotte dimensioni, i nanosatelliti hanno la capacità di essere lanciati assieme ad altri satelliti, riducendo drasticamente il costo della fase di lancio.
Il limite dei nanosatelliti è rappresentato dal ridotto spazio disponibile a bordo. Una possibile soluzione con cui ovviare a ciò è reppresentata dalle formazioni di satelliti, ovvero più satelliti che lavorano in maniera coordinata per portare a termine la missione. Questi possono avere tutti lo stesso ruolo, o un ruolo diverso all'interno della formazione stessa.
L'oggetto di studio di questa tesi è legato ad una formazione di satelliti composta da un satellite madre e da uno swarm di nanosatelliti che orbitano attorno ad un sistema binario di asteroidi. Questa tesi analizza la possibilià di utilizzare un metodo chiamato Multiple Hypothesis Tracking per determinare in modo autonomo ed in tempo reale posizione e velocità dei nanosatelliti, nel caso in cui questi siano molto vicini fra loro e possano fornire misure di sola distanza.
Nonostante il tipo di misura generata non sia sufficiente a calcolare analiticamente lo stato dei nanosatelliti, con questo metodo è possibile stimarlo anche nel caso in cui i nanosatelliti dovessero essere così vicini da avere le incertezze sulla loro posizione che si sovrappongono.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis proposes a method to perform autonomous and real-time tracking of a swarm of nanosatellites in the proximity of a binary asteroid system. Given the low cost of each platform and the small available on-board space, nanosatellites are considered to be only equipped with sensors providing the absolute value of their distance with respect to other objects. No information about the direction of the vector connecting the nanosatellite to such objects is available.
After that the nanosatellites have performed the measurements, they send them to a mother satellite (the mothership). This spacecraft is equipped with an on-board algorithm whose role is to associate the received measurement sets to the nanosatellites and to compute their states (position and velocity).
This study deals with satellite trajectories in which, for certain time windows, some nanosatellites are so close to each other that their state uncertainty overlap. In these regions it is not possible to uniquely define a measurement-to-satellite association.

The proposed solution to this problem exploits an algorithm applied in the multi-object tracking field: the Multiple Hypothesis Tracking (MHT). This method is capable of resolving ambiguities in the state estimation in case two or more satellites are very closely spaced. In particular, MHT is a decision logic which propagates multiple satellites' state hypotheses when measurements can not be uniquely associated to targets. The ambiguity resolution is delayed of few time steps, until enough information on satellites' state hypotheses are available.

The performances of the MHT will be presented in terms of precision in the estimation of the state and computational time. Due to the high computational cost of a MHT algorithm, it will be investigated if a real-time satellite tracking is viable.
Different scenarios will be tested, with increasing number of swarm satellites and different kind of trajectories.

Before presenting the developed algorithm, the characteristics of the nanosatellites and the importance of studying asteroid binary systems are introduced.

1.1 Nanosatellites

Space missions reflects the intrinsic necessity of humankind to learn more about life and what surrounds us. This is the reason why, since we had the knowledge and the technology to explore space, companies have invested billions of euros per year in research, designing and building launchers and spacecrafts. The amount of work throughout all the mission phases and the time it takes to design, build and test every component, reflects the elevated mission cost. For this reason, not so long ago, small companies did not have enough capital to enter this world.

In 1999 the CubeSat Project was born with the aim of providing university students with an accessible platform that could allow them to give a closer look at the space world. CubeSats are standardised platforms based on CubeSat "unit", referred to as 1U, a 10 cm cube with a mass of 1.33 kg [1]. By combining more units together, different shapes and dimensions can be created according to mission necessities. On-board components are often Commercial Off-The-Shelf (COTS), much more affordable than highly customised components developed for a specific satellite.
In addition, launching these platforms into space is quite affordable due to their capability of being launched as a piggyback, making use of any spare capacity as a single large satellite is launched.
Therefore, the reduced cost of manufacturing, testing and launching, along with the reduced duration of the overall mission, from the beginning of the design to the end, represents a good opportunity to deliver satellite in space with a limited budget.

Nanosatellites are a particular category of small satellites weighting under 10 kg and made of CubeSat units. Being the cost of a satellite proportional to its weight, nanosatellites represent a very good opportunity for an affordable space mission. In fact, depending on their specifications a nanosatellite can be built and placed in orbit for 500,000 €. In comparison, the cost of a conventional satellite can be as high as 500,000,000 € [2].
In the last years the technology matured and nanosatellites are now used to test, demonstrate and validate new systems in space at low cost and within a much shorter timescales than traditional space timetables. Hence, industries and space agencies are embracing the advantages of this technology. In fact, by looking at Fig. 1.1 [3], it is clear that more and more companies have been entering the nanosatellites world in the recent years.

In spite of their growth, nanosatellites present a major drawback, which is the limited space and weight on-board the platform. It means that care shall be taken when designing nanosatellite architecture because a limited on-board space is available. For this reason and due to their lower cost, their instruments are usually not as accurate as those on-board bigger satellites.

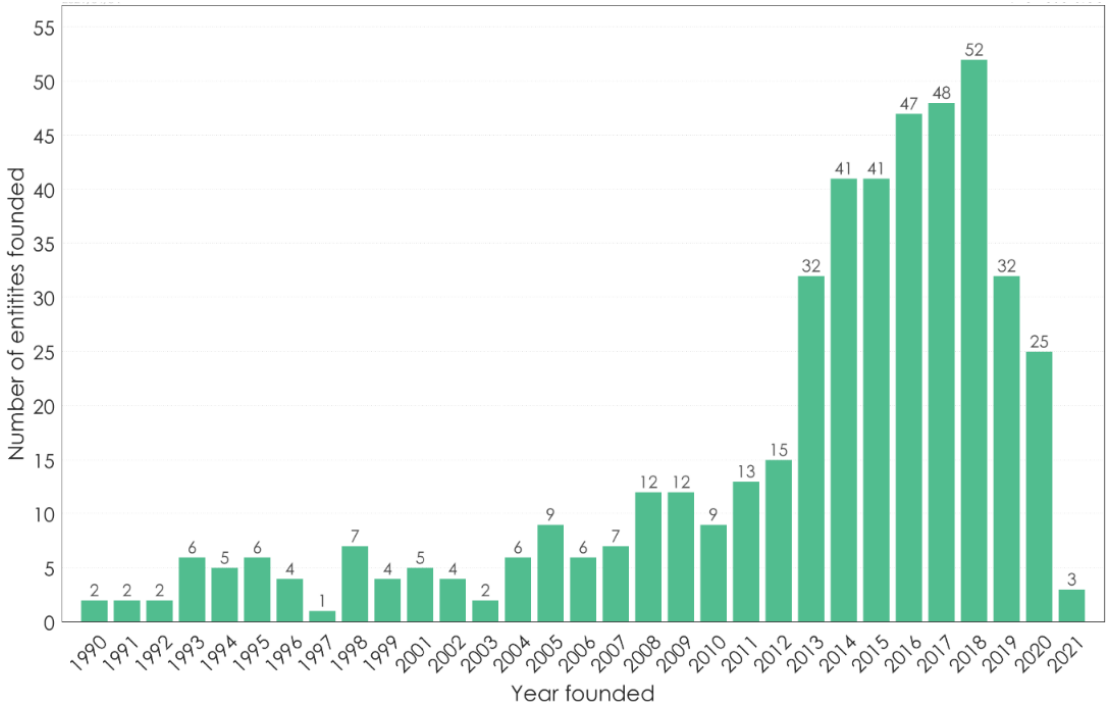


Figure 1.1. Founding years of companies active in nanosatellites since 1990 [3]

1.1.1 Satellite swarm

The solution to the problem of the reduced available on-board space is represented by simultaneously employing more platforms, working in a cluster, to accomplish mission goal. What can be done is to distribute the different mission tasks (and instruments if necessary) to different satellites. In other words, more nanosatellites, each one having a different role, can work as a team to achieve mission goals.
Regardless the small on-board space, it is still recognised that nanosatellites have a great potential for applications with scientific and technological value. Some of these applications include Earth observation, telecommunications, astronomy, and technology demonstrations.
The possibility to fly a large number of small spacecraft turns into a great advantage when dealing with coverage. SpaceX's Starlink mission, for example, aims at launching nearly 12.000 small satellites in Low Earth Orbit (LEO) to implement a new space-borne Internet communication system [4].
A swarm of nanosatellites can also be exploited to simultaneously acquire measurements or images from different locations, providing scientists with a more complete set of data. Moreover, being smaller than usual spacecrafts, they have the capability to orbit closer to objects of interest.

Satellite swarms can be classified into different categories depending on mission architecture. Some missions are designed in a way that the relative structure formed by the cooperating satellites is preserved: this case is classified as formation flying. Its peculiar aspect is that the control law is based on the relative state between two or more infrastructures. It is not important to track the absolute position of a satellite, but, instead, to control the relative separation inside the swarm [5].

Satellites can have different hierarchy inside a formation. For example, a spacecraft could act as leader of other follower spacecrafts: the leader is controlled to a reference orbit, whereas follower spacecrafts control their state relatively to the leader.
Another category is referred to as constellation, in which no relative information are needed and satellite states are controlled in an absolute manner.

In addition, researchers attention has been recently focused also on the possibility to combine a swarm of satellite to a mothership, thus forming a sort of "eco-system". This solution would benefit of great flexibility, combining the advantages of having a swarm of nanosatellites with those coming from a more complex spacecraft with higher reliability and higher capacities.
This concept could, for example, be adopted to collect data from advantageous points around a small body as for the swarm it would be easier to get closer to the body and then send the measurements to the mothership. Furthermore, measurement taken at the same time and from different locations could provide a more complete view of the body, information that a monolithic solution could not furnish. This information could then be transmitted to the mothership that would act as a flying station and as a link between the Earth and the swarm.

However, flying a very high number of small satellites in a cluster constitutes a risk. Their poor manoeuvrability, due to their reduced dimensions, to their relatively low cost instrumentation and to the high satellite density, could considerably increase the risk of collisions. This would increase the number of space debris, jeopardising the safety of the other orbiting satellites and of the planet they are orbiting around. About this aspect, the Committee on Space Research (COSPAR) has formulated a planetary protection policy stating that terrestrial biological contamination must not be introduced to other planets and moons that have potential for past or present life [6].
For this reason it is of paramount importance to precisely track their orbits and to avoid these unlucky events from happening.

1.2 Asteroid systems

Millions of asteroid are present the Solar System. They are small bodies that, when the Solar System was born, had not enough mass to reach a spherical shape and to form a planet. They have irregular shape and they are mainly found within the main asteroid belt between the orbit of Mars and Jupiter.
The scientific interest about asteroids has increased over the past decades when scientists understood that they have played key roles in Solar System's history [7]. They are considered to be the building block of the planets because "*asteroids, comets and other small bodies hold material from the solar system's birth. If we want to know where we come from, we must study these objects*", quoting Lori Glaze, acting director for the Planetary Science Division at NASA Headquarters in Washington [7].
For this reason they are considered to be tracers of the Solar System evolution. Furthermore, they represent a possible hazard for Earth protection as some asteroids happens to pass very close to the Earth. Additionally, NASA suggested manned asteroid mission could be the key for manned mission to Mars [8].

The multiple roles played by the asteroids represent the reasons why many missions to study the various scientific aspects of interests behind asteroids were launched. NASA's NEAR [9] was the first human-made object to orbit and land an asteroid in 2001. Then, other missions like Hayabusa, Rosetta [10], Stardust [11] and Dawn [12] deepen the knowledge on asteroids.
Through the years, it was found that a sizeable fraction of small bodies is a binary or multiple system. Whilst the formation of asteroid satellites is a consequence of large asteroid disruptions, the origin of the majority of smaller asteroids is believed to be different.
In addition, binary systems provide a means of determining critical physical properties (masses, densities, and rotations) with greater ease and higher precision than it is possible for single objects. Moreover, they are flying laboratories that permit to study many physical processes acting on the asteroids [13].
For this reasons, NASA and ESA have planned a joint mission around 65803 Didymos, a binary asteroid system. The scope of the AIDA mission is to study (through ESA's Hera probe) the change in the duration of the secondary body's orbit around the main one after a kinetic impact is performed (through NASA's DART mission).

1.3 Problem description

Due to the growing interest in the study of binary asteroid systems, this thesis aims at carrying on the studies on strategies to autonomously navigate satellites, in particular nanosatellites, in the close proximity of binary asteroid systems.
Though most of the studies have been focused on the navigation of a single spacecraft [14], recent interesting results shifted the attention on the possibility to navigate a swarm [15] or a formation of satellites [16, 17] instead. To do so, the most commonly adopted approach is to exploit optical measurements. In fact, as presented in the paper written by Massimiliano Vasile et al. [18], under certain measurement accuracy assumptions, optical measurements only are sufficient to navigate one or more spacecraft in the proximity of an asteroid binary system.
Differently to this approach, this work tries to exploit a different and reduced available measurement set: it aims at studying if range measurements alone are sufficient to autonomously determine the real-time orbit of a swarm of closely-spaced nanosatellites orbiting a binary asteroid system. The choice of nanosatellites, due to the advantages already mentioned in Ch. 1.1, with their reduced size, makes it of crucial importance to develop a reliable method that requires an amount of information that is as low as possible. In this way, few on-board instruments would be enough to provide the necessary information.

The scenario under investigation is represented by an "eco-system" composed by a swarm of low cost nanosatellites and a mother satellite (the mothership), orbiting the binary asteroid 65803 Didymos. In particular, the nanosatellites are orbiting in the proximity of the asteroids and they are, at least during part of their orbit, closely-spaced with respect to each other. Assuming to exactly know the position of the mothership, this thesis tries to answer the following question: is it possible for the mothership to autonomously determine the real-time orbit of the swarm of closely-spaced nanosatellites if they can only provide measurement of absolute distance

with respect to other bodies?
The following chapters try to answer this question through the use of a multi-object tracking method, the MHT.

Chapter 2

Multiple Target Tracking

The purpose of this chapter is to present the Multiple Target Tracking (MTT) problem and to give a brief historical review providing the most important approaches proposed through the years to tackle this problem. Strong and weak points of the various methods will be described along with the reasons why MHT is the method adopted for this work.

2.1 Tracking system

In the early 1970s autonomous target tracking methods, founded on Kalman filtering, started to be developed. The refinement of the Kalman Filter (KF) technique and the pressure coming from the aerospace field, pushed toward the creation of surveillance methods able to track objects inside a defined Field of View (FOV).
It has been over 40 years since the first survey on MTT methods, entitled "*Tracking Methods in a Multitarget Environment*", was published by Bar-Shalom [19] in 1978.

The objective of a general target tracking method is to interpret the data coming from one or more sensors and to look for targets in order to estimate their number, position and velocity. In particular, data are partitioned into sets of observations (tracks) that are believed to originate from the same source. The process of associating sensor's data to an object is intrinsically complex: the logic behind it must be robust to the background noise sources such as clutter, to internal error sources as thermal noise, to possible missed detection and to non linear target trajectory.
The strength of many tracking methods relies on their capability to update the same target track with more than one measurement at each time step whenever the observation-to-track assignment is not unique. Consequently, multiple hypotheses describing the possible evolution of the target are generated, and they are represented by different tentative tracks. Therefore, the number of tracks results to be higher than the number of real targets.
After a certain number of time steps, the assignment ambiguities are automatically resolved by using the measurement sets received in the meantime. At this point, the tracks associated with the highest hypothesis probability becomes confirmed, whereas the others are deleted.

Target tracking methods were first implemented in the field of surveillance systems

for military purposes. Through the use of sensors, such as radar, infrared, or sonar, alongside a computer subsystems, tracking methods had to interpret the signals coming from the surrounding environment looking for potential targets. Nowadays, the applications of MTT methods are various. They span from image processing [20], to robotic [21], passing through biomedical researches [22].
There exist many different MTT methods that can also be modified according to their field of application. A first distinction between the MTT methods is based on the number of measurement sets used to generate hypotheses on the update of a single track at each time step. At the expense of the computational time, some methods allow to generate various hypotheses based on different measurement-to-track associations; other methods only allow a unique assignment instead.
The latter methods, called Global Nearest Neighbour Filter (GNNF), are the simplest one, whereas the former are called All-Neighbors Data Association (ANDA) and they require an increased CPU memory to store the multiple hypotheses generated at each time step.

2.2 Global Nearest Neighbour Filter

The Nearest Neighbour Filter (NNF) is the simplest and computationally cheapest method for data association in the contest of target tracking.
It does not include probabilistic models for noise, clutter, detection or false alarm. Its formulation relies on the principle that only the "best" measurement (the one with the highest measurement-to-track association probability) is used to update a track at each time instant [23]. Hence, the number of tracks will always be equal to the number of targets.
In the majority of the applications, the best measurement at time k, $\mathbf{z}_k$, is represented by the one with smallest Euclidean distance d (Eq. 2.1) with respect to the previous track's observation $\mathbf{z}_{k-1}$.

$$d = [(\mathbf{z}_k - \mathbf{z}_{k-1})^T \cdot (\mathbf{z}_k - \mathbf{z}_{k-1})]^{1/2} \tag{2.1}$$

This method results to be vulnerable to dense targets scenario: if multiple measurement-to-track associations are possible, the Euclidean distance alone does not always pick the right association. Therefore, the NNF method could be improved by adopting the expression in Eq. 2.2. A prediction of the measurement ($\hat{\mathbf{z}}_k$) is estimated through a Kalman Filter and the nearest neighbour to that prediction in selected. $\mathbf{S}_k$ represents the measurement residual covariance matrix and this extension is referred to as Nearest Neighbour Standard Filter (NNSF).

$$d = [(\mathbf{z}_k - \hat{\mathbf{z}}_k)^T \, \mathbf{S}_k^{-1} \, (\mathbf{z}_k - \hat{\mathbf{z}}_k)]^{1/2} \tag{2.2}$$

When dealing with multiple targets, this method seeks to identify the most likely mutual exclusive assignment of observations to existing tracks by only looking at the information at current scan. This implementation is often referred to as Global Nearest Neighbour (GNN).
Despite this improvement, the GNNF still remains only useful for non demanding applications and, nowadays, it is considered to be obsolete.

2.3 The All-Neighbors Data Association approach

This approach is designed to improve the performance of the GNNF. The term *all-neighbors*, refers to the fact that the track update is performed using all the neighbouring observations within a certain region around the track.
As described by S. Blackman and R. Popoli in [24], if the GNNF merely seeks the single most likely hypothesis, the ANDA generates multiple hypotheses after each scan of data through the use of multiple observations. As a result, the update estimate of a track may contain contributions from more than one observation.

The approach described in this section is employed by the conventional MTT systems adopting a ANDA approach. They share the following common guidelines, also illustrated in Fig. 2.1.
The input data for these systems is represented by the a set of measurements (or observation) collected by the sensors. Each target is associated to a region of space (gate) inside which we expect its measurement to be with a certain probability. When a measurement set is received, the gating test determines which possible observations-to-tracks pairings have enough probability to be real in order to be considered plausible. If two or more objects are closely-spaced, their measurements can fall inside more than one gate, thus making the measurement-to-track assignment not univocal. In this case, more track hypotheses will be propagated.
Exploiting the information coming from the latest observation, tracks can be deleted or updated. The best track hypotheses are then computed. They are found in a way that they are associated with the highest probability to represent the reality and each of their measurements shall be assigned to one and only one track. Then, targets' state are predicted ahead to the arrival time of the next set of measurements.
The gates of these predicted targets are computed and the cycle of Fig. 2.1 starts again.

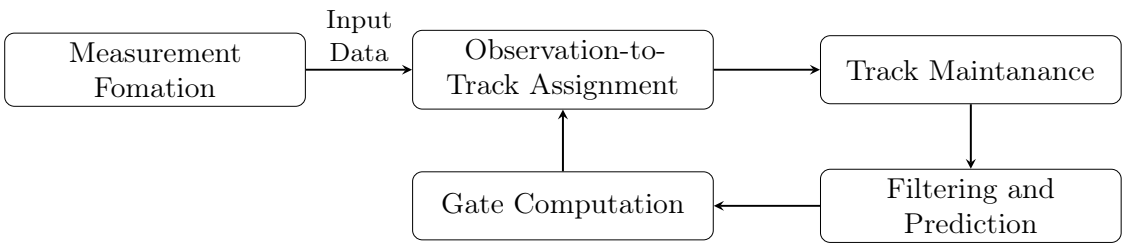


Figure 2.1. Basic elements of a conventional MTT system

This approach gives the possibility to deal with situations of closely-spaced targets: when two or more measurements fall within the same gate, the track associated to such gate will generate many updated state, each one updating the track using a different measurement. As a consequence, the number of propagated tracks will be higher than the number of actual objects, resulting in a significant increase of the computational cost.
The next sections of this chapter will present the basic working principles of some methods based on the ANDA approach.

2.4 Probabilistic Data Association Filter

An all-neighbour approach was first presented by Bar-Shalom and Tse [25]. It was called Probabilistic Data Association (PDA) and it led to a modified tracking filter called Probabilistic Data Association Filter (PDAF). It was conceived as an algorithm for tracking a single target in clutter and, in fact, it was shown that it is not able to correctly work in the case of multiple targets.

The idea behind this filter is to develop an estimator which incorporates all the observations that might have originated from the same object. In order to do so, the PDAF assumes that targets history is a Gaussian with a mean value and covariance. Moreover, the probability density function describing the new target density β_{NT} (the probability of a new object entering the observation space) and the false alarm density β_{FT} (the probability of missed detection) are both described by a Poisson distribution. Targets motion estimation is described by Kalman filter equations introduced in Ch. 3.2.

The algorithm start with the detection of N observations falling within the gate of track i at time k. At this point, the probability p_{ij} that measurement j is valid and the probability p_{i0} that all the N measurements are not valid are computed as follows:

$$p_{ij} = \begin{cases} \frac{b}{b+\sum_{l=1}^{N} \alpha_{il}} & j = 0 \text{ (no valid observation)} \\ \frac{\alpha_{ij}}{b+\sum_{l=1}^{N} \alpha_{il}} & 1 \leq j \leq N \end{cases} \tag{2.3}$$

where

$$\begin{aligned} b &= (1 - P_D P_G)\beta(2\pi)^{M/2}\sqrt{|S_i|} \\ P_D &= \text{probability of detection} \\ P_G &= \text{probability that a target falls within the track gate} \\ \beta &= \beta_{NT} + \beta_{FT} \\ M &= \text{measurement dimension} \\ |S_i| &= \text{determinant of the residual covariance matrix} \\ \alpha_{ij} &= P_D e^{-d_{ij}^2/2} \\ d_{ij}^2 &= \text{normalised distance} \end{aligned}$$

According to [24], all these hypotheses can be merged into a weighted sum that provides the value of the residual in Eq. 2.4. The result is then inserted into Eq. 2.5 to update target state $\hat{\mathbf{x}}_i$:

$$\tilde{\mathbf{y}}_i(k) = \sum_{j=1}^{N} p_{ij}\tilde{\mathbf{y}}_{ij}(k) \tag{2.4}$$

$$\hat{\mathbf{x}}_i(k \mid k) = \hat{\mathbf{x}}_i(k \mid k-1) + K(k)\tilde{\mathbf{y}}_i(k) \tag{2.5}$$

considering that

$$\begin{aligned}\tilde{\mathbf{y}}_{ij}(k) &= \mathbf{y}_j(k) + H\hat{\mathbf{x}}_i(k \mid k-1) \text{ , measurement innovation}\\ \mathbf{y}_j(k) &= \text{observation } j \text{ received at time } k\\ p_{ij} &= \text{probability of measurement } j \text{ to be the correct one for track } i\\ K(k) &= \text{Kalman Gain at time } k\end{aligned}$$

After that, the covariance of the state at time k can be updated. In order to do so, two components are exploited: the first one is the covariance that would result if only one observation happened to be a valid association (Eq. 2.6), whereas the second term keeps into account the possibility of uncertain association (Eq. 2.7).

$$P^0(k \mid k) = p_{i0}P(k \mid k-1) + (1-p_{i0})P^*(k \mid k) \tag{2.6}$$

$$dP(k) = K(k)\left[\sum_{j=1}^{N} p_{ij}\tilde{\mathbf{y}}_{ij}\tilde{\mathbf{y}}_{ij}^{'} - \tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^{'}\right]K^{'}(k) \tag{2.7}$$

Defining $P^*(k \mid k)$ as the standard covariance matrix for a Kalman filter of the type described in Tab. 3.2, the updated covariance is:

$$P(k \mid k) = P^0(k \mid k) + dP(k) \tag{2.8}$$

In the presence of a single target in clutter, the application of a PDA generally shows a significant decrease in the number of lost tracks when compared to a standard GNN method at the expense of a slight increase of computational cost.

However, the PDA method does not perform well in the presence of multiple targets: when two targets have similar trajectories, this method tends to merge their paths. As a consequence, it was implemented an extension of the PDA to deal with a multi-target environment.

2.5 Joint Probabilistic Data Association Filter

The Joint Probabilistic Data Association (JPDA) method was developed as a derivation of the PDA with the aim of enhancing the performances in the presence of multiple targets [26].
The algorithm of the JPDA is identical to the PDA one: the state estimation is still computed using Eq. 2.5 and the covariance matrix still comes from Eq. 2.8. The only difference lies in the probabilities calculation.

Since this method makes use of all the observations falling within a track gate to compute the measurement-to-track association probabilities, the formula in Eq. 2.3 must now be expanded in order to embed multiple track probabilities.
In other words, if one measurement falls within more than one gate, it is used to generate multiple track hypotheses, as multiple measurement-to-track associations are plausible. A typical example of conflict situation that this method can handle is illustrated in Fig. 2.2.
In this example, the gates of track $T1$ and $T2$ are represented around their predicted

states, respectively $\hat{x}_1$, and $\hat{x}_2$, and three observations ($O1$, $O2$, $O3$) fall within such gates. Figure 2.2 is an example of a situation in which the gating areas of two tracks overlap and the observation-to-track assignment is not unique. In fact, observation $O1$ only belongs to the gate of $T1$, whereas observation $O2$ and $O3$ fall within both the gates.

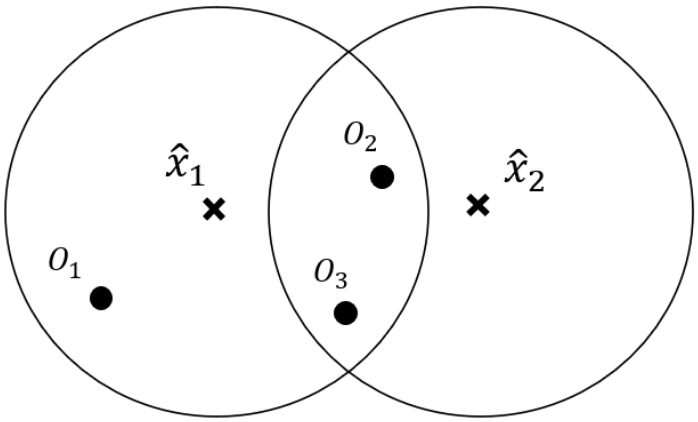


Figure 2.2. Gating areas of track 1 ($T1$) and track 2 ($T2$)

In this case, as all the observations belong to the first gate, the JPDA method will compute the residual for track $T1$ using $O1$, $O2$ and $O3$. A lower weight will be given to $O2$ and $O3$, accounting for their presence inside the second gate too. Differently, $T2$ will only be computed through $O2$ and $O3$.
The Gaussian likelihood function, associated with the assignment of observation j to track i, is different with respect to PDA approach and it is defined as:

$$p_{ij} = \frac{e^{-d_{ij}^2/2}}{(2\pi)^{M/2}\sqrt{|S_i|}}$$

Following this approach, different assignment hypotheses can be formulated, each one with its own likelihood value.
As a consequence, at the expense of a significant increase in the computational demand, this method permits to deal with multi-target scenarios. Nevertheless, the JPDA experiences the same coalescence problem found for the PDA, merging similar trajectories.

2.6 Multiple Hypothesis Tracking

The most commonly adopted tracking method is the MHT method. It is based on a deferred decision logic that, whenever a new set of measurements is received, generates and updates already existing hypotheses with new measurements.
Instead of combining these hypotheses as in the JPDA approach, whenever the measurement-to-track assignment is non unique, the MHT algorithm stores and propagates the conflicting hypotheses until subsequent scans of observation will

resolve the uncertainty at current time.
The idea behind this method makes it possible to deal with difficult multiple target tracking situation: probability of detection lower than the unity, unknown targets number, false reports or closely-spaced targets.

In 1974 Singer *et al.* [27] started to developed this modern algorithm to deal with target tracking problems. However, the first to implement a complete MHT framework was D. B. Reid in 1979 [28]. The basic assumption behind this approach is that measurements must not necessarily be assigned to tracks at each scan; anyway, in case of measurement-to-track association, each measurement can be assigned to, at most, one confirmed track.
The algorithm developed by Reid is called Hypothesis Oriented Multiple Hypothesis Tracking (HOMHT) because it maintains hypotheses from scan to scan. A different approach was proposed in 1990 by Kurien [29], when he developed the first Track Oriented Multiple Hypothesis Tracking (TOMHT) method. Instead of maintaining hypotheses through scans, the TOMHT initialises the tracks and it saves different sequences of measurements associated to each of the track whenever conflicts arise in the measurement-to-track assignment phase.

A general MHT formulation is able to deal with measurements assignment conflicts. As a result, it can track many different targets with similar trajectories without merging their paths as it happened in both the PDA and the JPDA algorithms.
However, this characteristic comes at the price of an exponential increase in the computational demand. For this reason, for many years MHT has been considered as not capable of providing real-time tracking information. Despite this, the evolution of technology lead, in the early 2000s, to new studies that revealed that this method was then able to meet real-time demands [30].

2.6.1 Hypothesis Oriented MHT

The original MHT algorithm presented by Reid is a hypothesis-oriented (or, alternatively, measurement-oriented) algorithm. The MHT assumes that the target state and covariance evolve with time according to the equations of a standard Kalman filter. As written by Reid in [28], "*the conditional probability distribution of the state variables of each target is a multivariate normal distribution given by the Kalman filter*".
The name of this method refers to the fact that the main structure, embedding all the information that are carried throughout the scans, is built collecting compatible hypotheses. A track hypothesis is defined as a sequence of measurements associated to a track. All of these measurements fell inside the track's gate (or validation region). A set of hypotheses is said to be compatible if they do not share any common observation at the same time instant. A set of compatible track hypotheses generates a global hypothesis.
Whenever a set of measurements is received, it is computed a list describing which observations fall within which targets validation region. The dimension and shape of the gate depends on the problem dimension and on the probability that a measurement is found in such region (see Ch. 4).
Then, each of the already existing hypotheses is expanded using all the possible

measurement-to-track assignments, in order to form a set of new compatible track hypotheses. Each hypothesis is associated to a probability value of representing the correct sequence of measurements associated to targets' motion. The value of such probabilities is described by Eq. 16 of [28].
After that, the global hypotheses shall be generated by choosing compatible track hypotheses. The likelihood value of a global hypothesis depends on the likelihood of its track hypotheses. The global hypothesis with the highest likelihood is usually provided to the user as the best target's position estimate.

The main drawback of a general MHT approach is represented by the computational time and memory needed to store all the possible track hypotheses. In order to limit the exponential increase of the number of hypotheses, Reid proposed different techniques [28]. The simplest technique is to prune all the unlikely hypotheses characterised by likelihood value lower than a threshold value. Another way to keep under control the number of hypotheses is represented by the N-scan pruning technique: defined a temporal window of N steps, only the hypotheses that originate from the root nodes at $t = N - 1$ are saved. The root node is the node associated to the measurement, at time $t = N - 1$, of a track belonging to the best global hypothesis (see Ch. 4.6).

Therefore, through this approach, it is possible to keep track of all the possible measurement-to-track assignment. Despite the presence of pruning techniques, HOMHT still experiences an exponentially growing number of hypotheses from scan to scan. Even though in [30] it was proved that an HOMHT is capable of guaranteeing real-time tracking, this algorithm is considered too complex to be widely adopted in real-time target tracking problems.

2.6.2 Track Oriented MHT

In 1990, Kurien introduced a new MHT method that differed from Reid's algorithm. Instead of maintaining hypothesis from scan to scan, the TOMHT maintains a set of tracks for each target. However, the method was not so efficient until a significant contribution in the development of such algorithm was given by Bar-Shalom *et al.* [31] in 2007. They proposed a way of drastically reducing the computational time by implementing a recursive formula to compute the score of an hypothesis.
In a TOMHT approach, the tracks are assumed to be initialised and, whenever more than one measurement falls within a track validation region, the track splits, originating two or more branches of the so-called track tree. Each target path is associated with a target tree that collects all the possible measurement-to-track associations.

Fig. 2.3 shows an example of track trees originating from the situation depicted in Fig. 2.2. *Tree 1* and *Tree 2* are respectively associated to target 1 and target 2 and they are initialised with $O1$ and $O2$. At time $t = k$, three observations are received, two of those ($O2$ and $O3$) are inside both the gates. They will thus expand the track tree of both the tracks.
A track tree branch is an observation sequence that defines a track hypothesis. A global hypothesis is a set of non conflicting track hypotheses that expresses the measurement-to-track assignment sequence for all the targets. As tracks are compatible only if

they do not share any measurement at the same time, all the branches of a tree are incompatible with each other as they share (at least) the root node. The ensemble of all the branches of a tree is called family. Consequently, only one track hypothesis per each family can be part of the global hypothesis.

A characteristic of the TOMHT is to include, at every time step and inside every tree, the possibility that a false alarm (FA) is reported. This term is referred to the situation that could arise from the case of a measurement falling outside the gating region, from a non properly working sensor or from a target leaving the observation space.
Furthermore, in general, it is accounted for the possibility that new targets could enter the scene. For this reason, at each time step, some dummy track trees (*Tree 3*, *Tree 4* and *Tree 5* in Fig. 2.3) are initialised with the false alarm. These trees simulate that objects that were not present in the observation space at time $t = k - 1$, could be detected at time $t = k$.

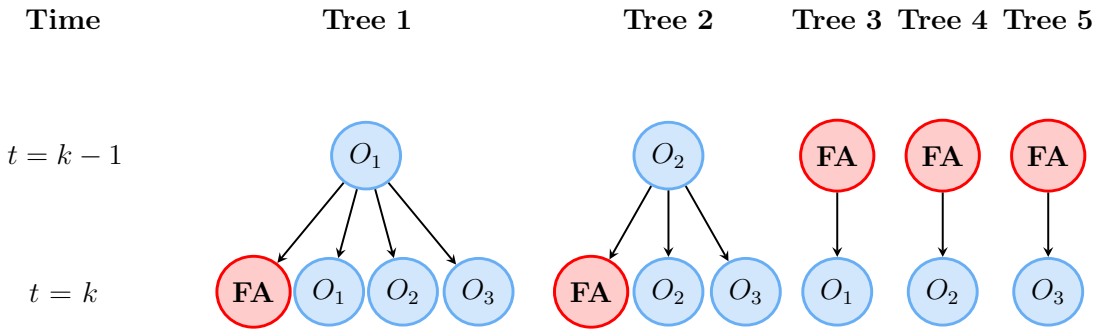


Figure 2.3. Track trees referred to the situation in Fig. 2.2

The problem of finding the best global hypothesis can be formulated as a Maximum Weighted Independent Set (MWIS) problem [32] and can be solved with an Integer Linear Programming (ILP) algorithm [33].

Also for this tracking method, a pruning technique is required in order to keep under control the growth of the CPU memory needed. The most common one is the N-scan pruning. The length of a temporal window is chosen and only the observations received within those N instants are retained and used for target's state estimation (see Ch. 4).

The main drawback of the TOMHT approach is represented by the difficulty in the track initialisation for many applications. In addition, including the possibility of targets entering the scene at each time instant, (*Tree 3*, *Tree 4* and *Tree 5* in Fig. 2.3), considerably increases the computational time and the memory needed to save all the possible track hypotheses.
Nevertheless, if the working scenario already provides the number of targets, this approach is much more convenient than a HOMHT one.

Chapter 3

Model description

This chapter introduces the model used to describe the motion of the satellites around the binary asteroid system. It is then described the process of satellites position and velocity estimation through the use of an EKF.

3.1 Dynamic model

The scope of this work is not to design and test precise orbital trajectories of the nanosatellites around a precise asteroid system. Instead, the aim is to analyse the feasibility of a new autonomous orbit determination approach to be applied to a low budget mission that employs a nanosatellite swarm cooperating with a mother-satellite; this method has been tested on a real environment scenario, such as the binary asteroid system of Didymos. For this reason, the dynamic model describing the motion of a body around the binary system neglects minor components that would not carry any advantage to this work.
In this section it will be presented the dynamic model adopted to simulate the motion of a satellite around the binary asteroid systems, pointing out the assumptions that have been made.

65803 Didymos is a binary asteroid system which is part of the Near Earth Objects (NEOs). It is composed of a primary body, Didymos, that has a diameter of around 780 m, and a secondary body, Dimorphos, of around 160 m of diameter that rotates around the primary at a distance of 1.18 km in around 11.92 h [34].
Table 3.1 summarises some of the most important characteristics of the binary system [18]. It is here highlighted the accentuated ellipsoidal shape of the secondary body, which generates a non uniform gravity field that could be model through spherical harmonics.
For the purpose of this work, as this contribution only introduces a slight variation in the gravitational attraction of the body, it is neglected and both the asteroids are considered to be perfectly spherical. In the adopted model, also the contributions given by the Solar Radiation Pressure (SRP) and the gravity field of the Sun are neglected .
This asteroid system has been chosen as case study for this work as its characteristics are well known and, moreover, because of the attention that the HERA mission [34] is putting on it.

The equations of motion of a satellite in proximity of this asteroid system (Eq. 3.1) are those of a Circular Restricted Three Body Problem (CR3BP) expressed with respect to a rotating frame (Fig. 3.1).
Referring to the main asteroid Didymos with the index A and to the secondary asteroid Dimorphos with index B, the acceleration that a body undergoes in their presence is:

$$\begin{cases} \ddot{x} &= -\dfrac{Gm_A(x-x_A)}{r_A^3} - \dfrac{Gm_B(x-x_B)}{r_B^3} + 2\omega\dot{y} + x\omega^2 \\ \ddot{y} &= -\dfrac{Gm_A(y-y_A)}{r_A^3} - \dfrac{Gm_B(y-y_B)}{r_B^3} - 2\omega\dot{x} + y\omega^2 \\ \ddot{z} &= -\dfrac{Gm_A(z-z_A)}{r_A^3} - \dfrac{Gm_B(z-z_B)}{r_B^3} \end{cases} \tag{3.1}$$

where

$$\begin{aligned} [x, y, z] &= \text{coordinates of satellite's position } [m] \\ G &= \text{gravitational constant } [m^3/kg \cdot s^2] \\ \omega &= \text{angular velocity of the binary system } [rad/s] \\ m_i &= \text{mass of asteroid } i \ [kg] \\ r_i &= \text{distance from asteroid } i \text{ to the satellite } [m] \end{aligned}$$

The simulations will be performed according to the reference frame illustrated in Fig. 3.1. It is a rotating frame with origin in the centre of mass of the binary system and it is such that the asteroids always lie on the x-axis. The system is rotating around the z-axis with angular velocity $\boldsymbol{\omega} = [0, 0, \omega]$. In this frame, the asteroids have a fixed position and the relative motion of the satellites with respect to the binary asteroid system can therefore be analysed.

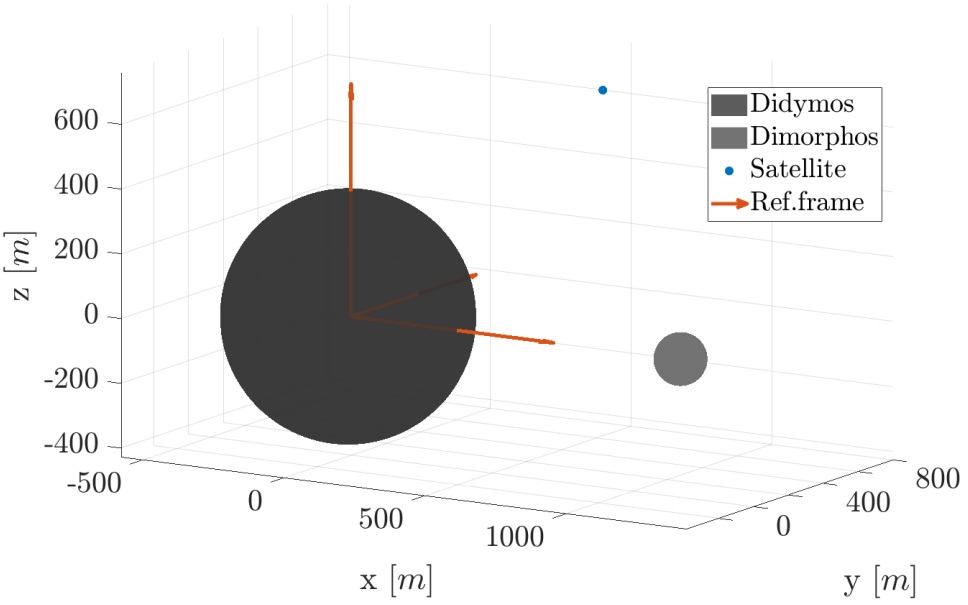


Figure 3.1. Rotating reference frame adopted inside the model

Orbital characteristics		
Semi-major axis	a	1.6446 AU
Eccentricity	e	0.3838
Inclination	i	3.4077°
RAAN	Ω	72.2219°
Argument of periapsis	ω	319.2516°
Orbital period	T	2.11 $years$
Physical characteristics		
Asteroid distance	d_{AB}	1.18 km
System rot. period	T_{AB}	11.92 h
Gravitational param. A	μ_A	$3.4098 \cdot 10^{-8}$ km^3/s^2
Gravitational param. B	μ_B	$3.1781 \cdot 10^{-10}$ km^3/s^2
Rotational period A	T_A	2.26 h
Rotational period B	T_B	11.92 h

Table 3.1. Characteristics of 65803 Didymos

3.2 State estimation

In a target tracking problem it is fundamental to continuously monitor some important parameters of the targets such as position and velocity. For this purpose, it is necessary to exploit the capacity of a filter that, with the help of the available measurements, is able to estimate these important parameters.
All the MHT methods need to work in couple with a filter in order to estimate targets' information. The most of the times MHT algorithms are associated to a KF, thanks to its power and robustness. Hence, the same filter is adopted for this work.
Working with the reference frame in Fig. 3.1, the target state $\boldsymbol{X}$ is expressed in Cartesian coordinates and it is composed by its position and velocity:

$$\boldsymbol{X} = [x, y, z, \dot{x}, \dot{y}, \dot{z}] \tag{3.2}$$

Whenever dealing with a nonlinear target dynamic or a nonlinear observation process, an EKF is necessary to correctly estimate targets' motion. This variation of the filter is able to linearize both the motion and the observation processes about the current mean and variance through Taylor series approximation.
In the scenario analysed in this thesis, both the dynamic (Eq. 3.1) and the observation process are nonlinear (Eq. 3.7). The EKF models targets' evolution and observation process through the nonlinear functions $f_{k-1}(\cdot)$ and $h_k(\cdot)$ as:

$$\boldsymbol{X}_k = \boldsymbol{f}_{k-1}(\boldsymbol{X}_{k-1}) + w_k \tag{3.3}$$

$$\boldsymbol{z}_k = \boldsymbol{h}_k(\boldsymbol{X}_k) + \nu_k \tag{3.4}$$

where $\boldsymbol{X}_k$ and $\boldsymbol{z}_k$ are the state and the observation at time k. w_k and ν_k are zero mean additive Gaussian noises that are independent of each other. They respectively represent the process noise and the measurement noise.

The estimation process implemented in this thesis consists of the two phases illustrated in Tab. 3.2.
In the prediction phase the filter computes a predicted state estimate $\hat{\boldsymbol{X}}_k^{\,-}$ along with its predicted error covariance matrix $\boldsymbol{P}_k{}^{-}$. The dynamic is propagated through the state transition matrix $\boldsymbol{\Phi}(t_k,\ t_{k-1})$. In order to compute it, it is necessary to linearize the dynamic. The linearization is embedded in the Jacobian of the dynamic calculated at current state $\hat{\boldsymbol{X}}(k)$, that is matrix $\mathbf{A}_k$:

$$\boldsymbol{A}_k = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{X}}\right|_{\hat{\boldsymbol{X}}(k)} \tag{3.5}$$

In order to estimate the predicted error covariance matrix, it is necessary to know not only the state transition matrix, but also the covariance matrix of the process noise $\boldsymbol{Q}_k$.
In the update phase, the measurement residual $\mathbf{Y}_k$ is used to update the predicted state estimate ($\hat{\boldsymbol{X}}_k^{\,+}$) and the predicted error covariance ($\boldsymbol{P}_k^{+}$). In this process, the the linearization of the observation process (Eq. 3.6) and the measurement noise covariance matrix $\boldsymbol{R}_k$ are needed too.

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}_k}{\partial \boldsymbol{X}}\right|_{\hat{\boldsymbol{X}}_k^{-}} \tag{3.6}$$

The complete EKF algorithm can be summarised as follows:

Prediction

Predicted state estimate	$\hat{\boldsymbol{X}}_k^{\,-} = \int_{t_{k-1}}^{t_k} \boldsymbol{f}(\boldsymbol{X}(\tau),\tau)d\tau$
State transition matrix	$\boldsymbol{\Phi}(t_k,t_{k-1}) = \int_{t_{k-1}}^{t_k} \boldsymbol{A}(\tau)\boldsymbol{\Phi}(t_k,\tau)d\tau$
Predicted error covariance	$\boldsymbol{P}_k{}^{-} = \boldsymbol{\Phi}(t_k,t_{k-1})\boldsymbol{P}_{k-1}^{+}\boldsymbol{\Phi}^T(t_k,t_{k-1}) + \boldsymbol{Q}_k$

Update

Measurement residual	$\boldsymbol{Y}_k = \boldsymbol{z}_k - \boldsymbol{h}_k(\hat{\boldsymbol{X}}_k^{\,-})$
Kalman gain	$\boldsymbol{K}_k = \boldsymbol{P}_k{}^{-}\boldsymbol{H}_k{}^{T}(\boldsymbol{H}_k\boldsymbol{P}_k{}^{-}\boldsymbol{H}_k{}^{T} + \boldsymbol{R}_k)^{-1}$
Update state estimate	$\hat{\boldsymbol{X}}_k^{\,+} = \hat{\boldsymbol{X}}_k^{\,-} + \boldsymbol{K}_k\,\boldsymbol{Y}_k$
Update error covariance	$\boldsymbol{P}_k^{+} = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k]\boldsymbol{P}_k^{-}$

Table 3.2. Algorithm of an EKF in its prediction and update phases

For what concerns the scenario discussed in this thesis, a filter of this type is implemented on-board the mothership. The nanosatellites are not equipped with any kind of filter. They only take the necessary measurements and send them to the mothership, which elaborates the information in order to identify satellites' position and velocity in the reference frame in Fig. 3.1.

3.3 Measurement model

As already mentioned, the observation process adopted in the developed filter is nonlinear. In fact, nanosatellites send to the mothership information about the norm of relative distances, without any information about the direction.
The expression of the distance between two bodies is nonlinear with respect to the adopted state in Eq. 3.2.
The measurement sets sent by the nanosatellite i to the mothership a time k is in the form of vector $\mathbf{z}_k$ in Eq. 3.7, where the indices A, B and M, respectively stand for the main asteroid, the secondary asteroid and for the mothership.

$$\mathbf{z}_k = \begin{bmatrix} \sqrt{(x_i - x_M)^2 + (y_i - y_M)^2 + (z_i - z_M)^2} \\ \sqrt{(x_i - x_A)^2 + (y_i - y_A)^2 + (z_i - z_A)^2} \\ \sqrt{(x_i - x_B)^2 + (y_i - y_B)^2 + (z_i - z_B)^2} \end{bmatrix} \tag{3.7}$$

The measurements are taken from the nanosatellites to the mothership and to the two asteroids. In this way, as the position of the mothership and of the asteroids are known, there will be, in general, two possible solutions defining the position of a nanosatellite. Therefore, assuming to have an initial guess for the filtering process that is close enough to the real position, the EKF will automatically try to estimate the solution associated to the real position of the satellites.

Provided the measurements in Eq. 3.7, the solving equations to find the position of a spacecraft are derived in a similar manner to those of Chapter 4.1 found in [35]. Here, M. Vasile *et al.* analysed a scenario composed of 3 spacecrafts communicating to a station their inter-satellite measurements made of relative distance, local azimuth and elevation. Exploiting the Time Of Arrival (TOA) equations, they proposed a way of computing the position of the spacecrafts one by one.
In this thesis the mothership acts as a flying station receiving the distances information from the nanosatellites. In analogy to the system of Eqs. 13 in [35], the two asteroids can be treated as two satellites with known position and "communicating" to the mothership the value of their relative distance.
For the sake of simplicity, the TOA equations here presented are written with respect to a reference frame originating in the mothership. The indices A and B refers to the main and secondary asteroids, whereas index 1 refers to the nanosatellite whose position has to be computed.
As the position of the asteroids with respect to the mothership is known ($\mathbf{r}_A$ and $\mathbf{r}_B$), the TOA equations associated to the measurement set taken on-board nanosatellite 1 are:

$$d_1{}^2 = x_1{}^2 + y_1{}^2 + z_1{}^2 \tag{3.8}$$

$$d_{A1}{}^2 = (x_A - x_1)^2 + (y_A - y_1)^2 + (z_A - z_1)^2 \tag{3.9}$$

$$d_{B1}{}^2 = (x_B - x_1)^2 + (y_B - y_1)^2 + (z_B - z_1)^2 \tag{3.10}$$

where $d_1{}^2, d_{A1}{}^2$ and $d_{B1}{}^2$ are respectively the distances from the nanosatellite to the mothership, to asteroid A and to asteroid B. In order to compute the position of the

nanosatellite, Eq. 3.8 is substituted in Eq. 3.9 and in Eq. 3.10:

$$d_{A1}{}^2 - d_A{}^2 - d_1{}^2 = -2(x_A x_1 + y_A y_1 + z_A z_1) \tag{3.11}$$

$$d_{B1}{}^2 - d_B{}^2 - d_1{}^2 = -2(x_B x_1 + y_B y_1 + z_B z_1) \tag{3.12}$$

with $d_A{}^2 = x_A{}^2 + y_A{}^2 + z_A{}^2$ and $d_B{}^2 = x_B{}^2 + y_B{}^2 + z_B{}^2$. At this point, if x_B and x_A are different from zero, Eq. 3.11 and Eq. 3.12 can respectively be multiplied by x_B and by $-x_A$. Then, solving for y_1, one gets:

$$y_1 = \frac{\Delta_1 - b_1 z_1}{a_1} \tag{3.13}$$

where

$$\Delta_1 = (d_{A1}{}^2 - d_A{}^2 - d_1{}^2)x_B - (d_{B1}{}^2 - d_B{}^2 - d_1{}^2)x_A$$
$$a_1 = 2(y_B x_A - y_A x_B)$$
$$b_1 = 2(z_B x_A - z_A x_B)$$

Then, substituting Eq. 3.13 into Eq. 3.9, the expression of x_1 can be found:

$$x_1 = \frac{\Delta_2 - b_2 z_1}{a_2} \tag{3.14}$$

with

$$\Delta_1 = -d_{A1}{}^2 + d_A{}^2 + d_1{}^2 - 2\frac{y_A \Delta_1}{a_1}$$
$$a_2 = 2x_A$$
$$b_1 = 2(z_A - \frac{y_A b_1}{a_1})$$

Now, the expression of y_1 and x_1 can be inserted in Eq. 3.8 and a second order algebraic equation in the z_1 variable is obtained in Eq. 3.15:

$$A z_1{}^2 + B z_1 + C = 0 \tag{3.15}$$

defining the parameters A, B and C as follows:

$$A = \frac{b_1}{a_1}{}^2 + \frac{b_2}{a_2}{}^2 + 1$$
$$B = -2(\frac{\Delta_2 b_2}{a_2{}^2} + \frac{\Delta_1 b_1}{a_1{}^2})$$
$$C = \frac{\Delta_1}{a_1}{}^2 + \frac{\Delta_2}{a_2}{}^2 - d_1{}^2$$

It is now important to study how the solution to the system composed by Eq. 3.8, Eq. 3.9 and Eq. 3.10 varies depending on the relative positions assumed by the bodies during their motion. This analysis could permit to avoid dangerous relative satellite positions that would not allow to correctly estimate target's motion.

In [35], M. Vasile *et al.* proved that, depending on the case, the system of equations in Eq. 13 of such paper can have a or not have a unique solution. Extending those results to this case scenario due to the analogy of the TOA equations, where $\boldsymbol{r}_A$ and $\boldsymbol{r}_B$ are the position vectors of the asteroids with respect to the mothership, it can be stated that:

- the system has two distinct solutions provided that $||\boldsymbol{r}_A \wedge \boldsymbol{r}_B|| > 0$ and $B^2 - 4AC > 0$;
- the system has one solution if $||\boldsymbol{r}_A \wedge \boldsymbol{r}_B|| > 0$ and $B^2 - 4AC = 0$;
- if $||\boldsymbol{r}_A \wedge \boldsymbol{r}_B|| = 0$ the system of equation has no unique solution;
- if all the bodies are aligned, the system has one degenerate solution.

These cases will be tested during the simulations in order to study the behaviour of the filter when the relative positions between the bodies change. In particular, alignments between the bodies and coplanar motion will be simulated. The performances of the filter in the different situations will be compared.

Chapter 4

Algorithm implementation

The scope of this thesis is to investigate if range measurements only, taken on-board the nanosatellites, are sufficient to compute nanosatellites' orbits through a filter implemented on-board the mothership.
This problem is addressed through the use of a MHT algorithm. In particular, a TOMHT has been selected because it is generally less computationally demanding with respect to a HOMHT. The former implementation could be more computationally expansive if track initialisation is needed. However, in the scenario treated in this thesis where the number of targets is assumed to be already known, there is no need for implementing track initialisation, making the TOMHT simpler and more suited than a HOMHT.

This chapter will serve as a presentation of the developed TOMHT algorithm. It is applied to the scenario in which a swarm composed by N nanosatellites is orbiting in proximity of the 65803 Didymos binary asteroid system and it is communicating with a mothership that orbits the same asteroid system.

Fig. 4.1 illustrates the basic steps of the algorithm through the use of a flowchart. The algorithm starts building N track trees, one for each nanosatellite of the swarm. The initial guesses of nanosatellites' state are assumed to be close enough to the real initial states that the predicted measurements they generate fall inside the validation region of their related satellites.
The first step of the algorithm is represented by the *Prediction* of the state and covariance of the N satellites. In case a measurement set has been received inside the gate between t and $t + \Delta t$, all the possible measurement-to-track associations are evaluated in the *Gating* phase. Only those measurement characterised by a certain likelihood of association to a track are saved and used to *Update* such state.
After that, through the *Track management* phase, the states are organised into tracks, each one with its own observation sequence, state and covariance estimation, score function and other useful information like the presence of isolated clusters. At this point, if the number of measurements composing the observation sequences of the tracks is higher than a threshold level, the tracks undergo the *Pruning* process. This procedure is able to reduce the number of track hypotheses and to prevent their exponential growth.
The last step of the loop consists in the *Global Hypothesis Formation*, which will be provided to the user.

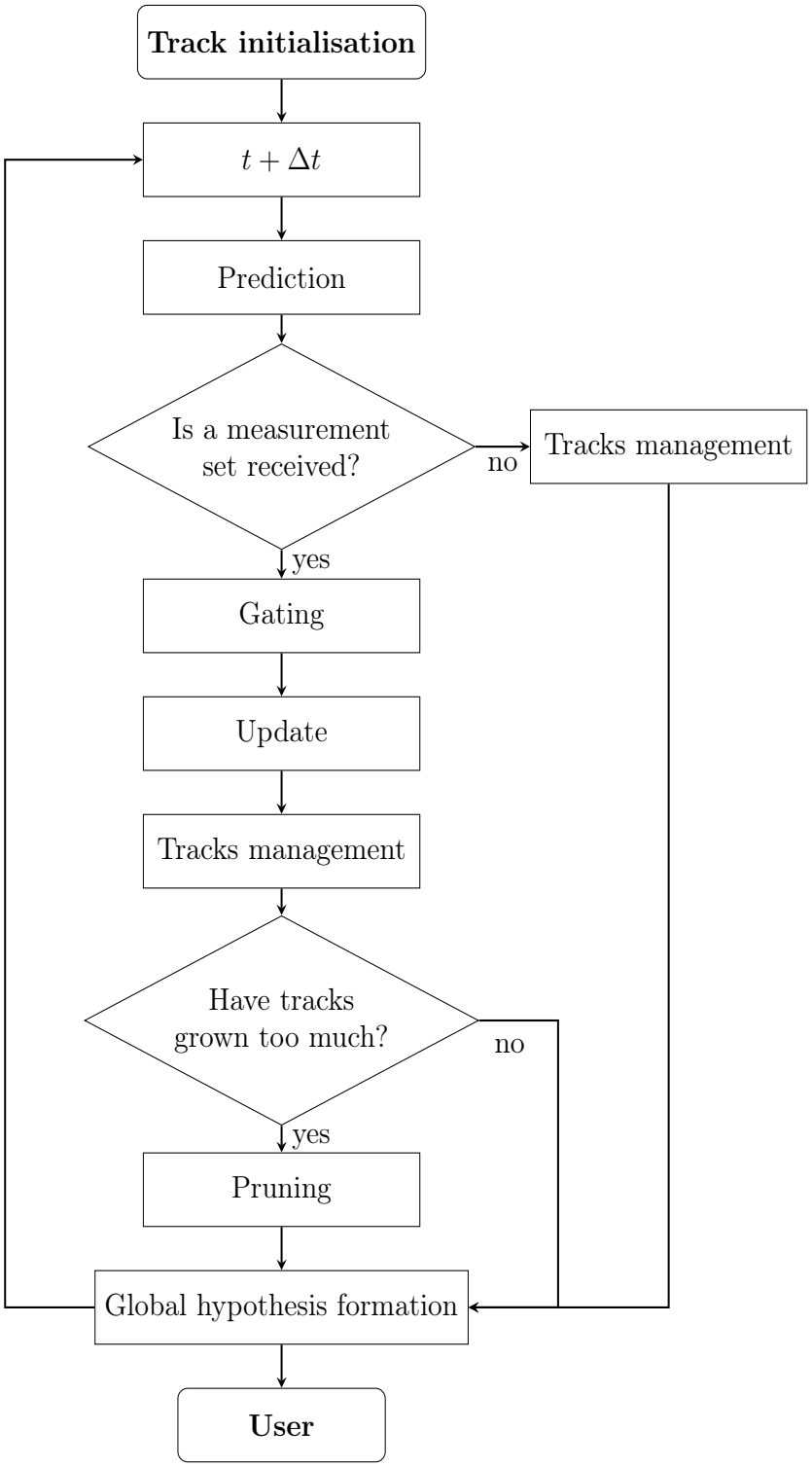


Figure 4.1. Algorithm flowchart

A more detailed description of the main steps of the TOMHT algorithm is given in the following sections of this chapter.

4.1 Gating

The first step of the algorithm consists in propagating the initial state toward the next integration instant. It is done computing a prediction of the state $\hat{\boldsymbol{X}}_i^-$ and of its covariance $\boldsymbol{P}_i^-$ through the *Prediction* phase of the EKF in Ch. 3.2.
Whenever new measurement sets are received, it is necessary to understand from which track they could have originated. Hence, such measurements are associated to the existing tracks following a certain criteria. In fact, associating all the received measurements to all the existing tracks, even when their association likelihood is very low, would represent a useless waste CPU memory.

The gating process represents a selection criteria that is used to limit the number of generated track hypotheses. This process assumes that measurements are distributed along a Gaussian centred in the measurement prediction $\hat{\boldsymbol{z}}_p(k)$ and with covariance $\boldsymbol{B}_k$. Such covariance matrix defines a region of acceptance, called validation gate, and its size depends on the value of a confidence level selected. If a measurement is compliant with this confidence constraint, it will then fall inside the gate.
The confidence level is related to the gate threshold value γ, which is typically defined as the inverse of the ${\chi_{k,\alpha}}^2$ cumulative distribution function. The ${\chi_{k,\alpha}}^2$ function assumes different values based on the values of the confidence level α and on the number of degrees of freedom k (the dimension of the problem). Through this parameter, the validation gate can be defined as a region of acceptance such that the $100(1-\alpha)\%$ of true measurements are rejected as they fall outside target gate [36].
Fig. 4.2 illustrates how the gate threshold value γ varies with respect to the confidence level α when the number of degrees of freedom increases.

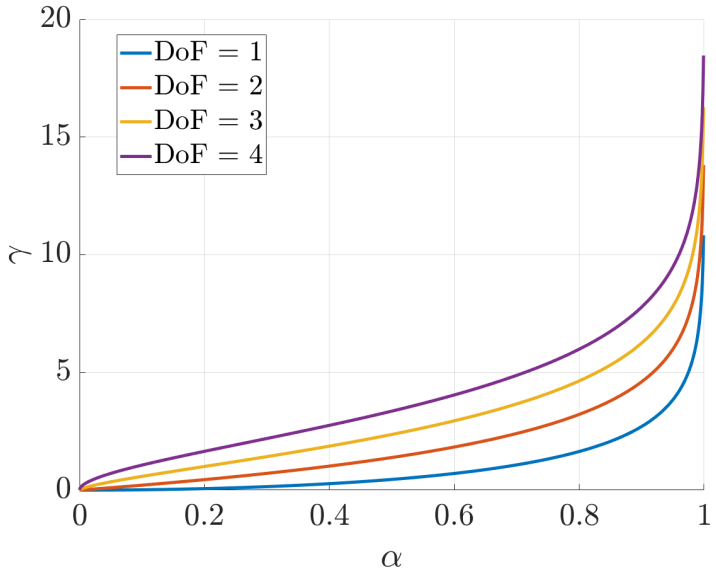


Figure 4.2. Inverse of the χ^2 cumulative distribution function

In a 3 dimensional problem the validation region has the shape of an ellipsoid centred in the measurement prediction. The higher the threshold value γ, the larger the volume of the gate.

Now that the gate has been built, how can one check if a measurement falls within it or not? In order to do so, the Mahalanobis distance (Eq. 4.1) is employed. It is defined as the distance, in a multivariate space, from a point to a distribution [37]. Denoted with $\mathcal{Z}_k = \{\boldsymbol{z}_{1_k}, \boldsymbol{z}_{2_k}, ..., \boldsymbol{z}_{k_k}, \ ..., \boldsymbol{z}_{M_k}\}$ the set of M measurements received by the mothership at time k and with $\boldsymbol{h}_k(\hat{\boldsymbol{X}}_k^{\ -})$ the measurement function generating the predicted measurement $\hat{\boldsymbol{z}}_{p_k}$ at the same time instant, the Mahalanobis distance d is defined as:

$$d^2 = (\boldsymbol{z}_{i_k} - \boldsymbol{h}_k(\hat{\boldsymbol{X}}_k^{\ -}))^T \boldsymbol{B}_k^{\ -1} (\boldsymbol{z}_{i_k} - \boldsymbol{h}_k(\hat{\boldsymbol{X}}_k^{\ -})) \tag{4.1}$$

$$\boldsymbol{B}_k = \boldsymbol{H}_k \boldsymbol{P}_k^{\ -} \boldsymbol{H}_k^{\ T} + \boldsymbol{R}_k \tag{4.2}$$

where the matrices in Eq. 4.2 have been already defined in Ch. 3.2. After that, comparing the Mahalanobis distance to the gate threshold, if $d^2 < \gamma$, the received measurement set will fall inside the gate.
The results of the association process are saved inside a gating matrix. It is filled with 1 in position (i, j) if the $j-$th measurement is inside the $i-$th gate and with 0 if such measurement falls outside the gate (Eq. 4.3). At the first step of the algorithm the number of gates coincides with the number of targets, but, in general, the number of gates is equal to the number of propagated tracks (or branches).
At this point, each measurement inside the region is used to update the state of the target associated to such gate through the *Update* sequence of the EKF (Tab. 3.2). In other words, each of these measurements will create a new branch in the track tree with its own likelihood to represent the right measurement-to-track association. Through the gating process, only likely associations will be stored and will expand the track tree. For the sake of clarity, Fig. 4.3 serves as an example to explain the gating process and to better illustrate how the algorithm is developed in the case here illustrated. Fig. 4.3 refers to the case in which three target satellites are moving in the observation space and it represents the conditions that can be found during the first loop of the algorithm.

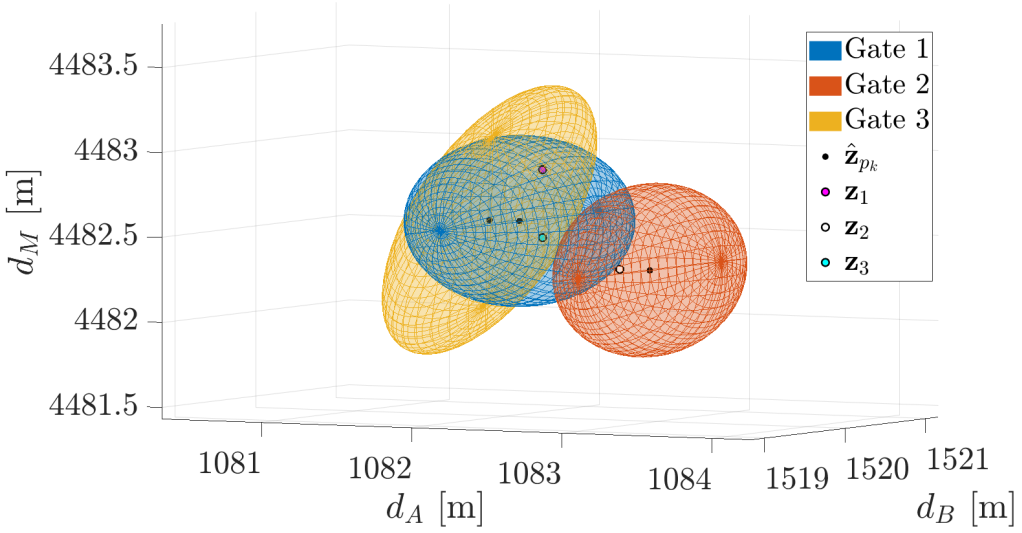


Figure 4.3. Example of track gates in the observation space

The gates are centred in the state predictions $\hat{\boldsymbol{z}}_{p_k}$ of the three nanosatellites in Fig. 4.3 and the gate volume is such that the confidence level α is $\alpha = 0.99$. A higher confidence level would result in bigger gates. The measurements $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ and $\boldsymbol{z}_3$ are vectors collecting the distances from the three nanosatellites to the main (A) and secondary asteroid (B) and to the mothership (M). One thing that should be pointed out is that the gates do not belong to the space in which targets are moving, they belong to the observation space. Due to the shape of the measurement vector $\mathbf{z}_k$, the gates can be represented through a 3D plot in which the axes are reporting the distance values composing the measurement vector.
The gating matrix for the situation depicted in Fig. 4.3 is:

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{4.3}$$

as the measurements $\boldsymbol{z}_1$ and $\boldsymbol{z}_3$ belong to both gate 1 and gate 3, whereas only measurement $\boldsymbol{z}_2$ falls inside gate 2.

4.2 Track management

When the gating process is completed, the next step of the TOMHT algorithm is to expand the track trees propagating satellites' states. This is done by updating a state with the different measurement sets as suggested by the gating process. Different updated states will be propagated, each one with a different probability to represent the best measurement-to-track relation.
This process also accounts for the possibility of a non received measurement or a coarse measurement falling outside targets' gates. Hence, at each observation time, all the tracks shall have a branch associated with a dummy measurement. The state associated to this false alarm only undergoes the *Prediction* phase in Tab. 3.2, without any available measurement to perform the *Update* phase with. On the contrary, the measurements corresponding to the $1s$ inside the gating matrix of Eq. 4.3 are exploited to update the tracks they were found to be close to.

Referring to the gating situation in Fig. 4.3, the track trees are expanded as shown in Fig. 4.4. False alarms are here represented through 0s, whereas 1, 2, and 3 are respectively associated to the $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ and $\boldsymbol{z}_3$ measurements.
In this case, as three nanosatellite are assumed to be part of the swarm, 3 track trees have been initialised. This means that 3 track trees are created a priori and each of them will be a priori associated to a different target. For the sake of clarity, the trees' ID in Fig. 4.4 corresponds to targets' ID they are associated with.
No other targets are expected to enter the observation space because the number of nanosatellites is known since the beginning of the mission. This is why the algorithm does not account for the possibility of new targets entering the scene. Therefore, there are no tracks initialised with the false alarm (as it was for Fig. 2.3).

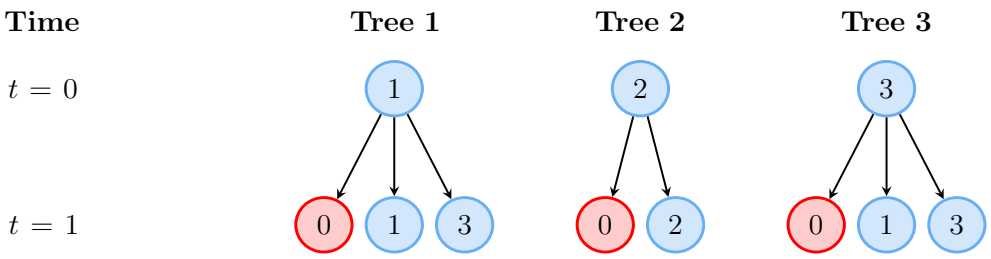


Figure 4.4. Track trees referred to Fig. 4.3

The tracks history described by the trees in Fig. 4.4 is stored inside the algorithm into the matrix form in Eq. 4.4:

$$\mathcal{Z} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 2 & 0 \\ 2 & 2 \\ 3 & 0 \\ 3 & 1 \\ 3 & 3 \end{bmatrix} \tag{4.4}$$

The number of columns in the $\mathcal{Z}$ matrix is equal to the number of time instants whose information are saved, whereas the number of rows is equal to the number of total branches (track hypotheses).
All the eight branches of matrix $\mathcal{Z}$ are associated to a different state and error covariance estimations, originating from the different associations between tracks and measurements used to update targets' states.

4.3 Track score

Between the track hypotheses generated in the previous step, the algorithm has to provide the user with the set of non conflicting track hypotheses that better than the others represents the state of the targets. This set is defined as the best global hypothesis.

This goal is achieved through the set up of a constrained optimisation problem. It is based on the minimisation of a cost function which is related to each measurement-to-track assignment. This score represents the likelihood that such assignment is correct. By expressing the score in terms of log-likelihood ratio (LLR), the assignment cost for a set of tuples is always the sum of the costs for the individual tuples. This is valid because the expressions of the scores are independent across tracks [31].
It means that, in order to compute the score function of a track hypothesis, it is sufficient to sum the score of each measurement-to-track associations composing such track hypothesis.
For numerical convenience, as the usual criterion to find the best assignment is by minimising a cost function, the negative log-likelihood ratio (NLLR) is often employed to represent the score function ($\ell_{t,j}(k)$). In this way, the cumulative NLLR for a track

t through time k can be recursively computed by adding the new association score at time l as illustrated in Eq. 4.5:

$$\ell_t{}^k = -ln\mathcal{L}_t{}^k = -\sum_{l=0}^{k} ln\mathcal{L}_{t,j}(l) = \sum_{l=0}^{k} \ell_{t,j}(l) \tag{4.5}$$

where j represents the measurement that can either be a false alarm ($j = 0$) or a true measurement ($j > 0$) at time l. Adopting for the score the expression used in [31], the NLLR in case of true measurement ($j > 0$) at time k becomes:

$$\ell_{tj}(k) = \frac{[\boldsymbol{z}_j(k) - \hat{\boldsymbol{z}}_p(k)]^T \boldsymbol{B}_{tj}(k)^{-1} [\boldsymbol{z}_j(k) - \hat{\boldsymbol{z}}_p(k)]}{2} + ln\frac{\lambda_{ex}|2\pi \boldsymbol{B}_{tj}(k)|^{1/2}}{P_{D_t}(k)} \tag{4.6}$$

where $P_{D_t}(k)$ is the detection probability of track t at time k, whereas λ_{ex} represents the spatial density of extraneous measurements (expected number per unit volume in frame/scan).
Each track is associated to an a priori score, which only depends on the spatial density of the new targets λ_ν and on the spatial density of false measurements λ_ϕ:

$$\ell_{tj}(0) = ln(\frac{\lambda_\nu}{\lambda_\phi}) \tag{4.7}$$

In the opposite situation, when a false alarm is reported ($j = 0$) at time k, the expression of the NLLR is:

$$\ell_{tj}(k) = -ln[1 - P_{D_t}(k))] \tag{4.8}$$

The expression in Eq. 4.8 reduces the likelihood of a track that has reported a false alarm as last observation.
Through the use of Eq. 4.6 and Eq. 4.8, a score function is associated to each track branch (or to each of the rows in the matrix in Eq. 4.4). It expresses the probability for each tracks to represent the best targets' state estimation.

4.4 Clustering

The main drawback of the MHT algorithms is represented by the computational burden when the number of branches of the track trees grows too much. Many techniques have been developed through the years to keep under control the allocated CPU memory and, consequently, the computational time.

One of these techniques is clustering. It consists in gathering together tracks that share the same measurements, at the condition that those measurements are not present in any other track. This is useful to lighten the optimisation problem because it can be subdivided into smaller optimisation problems.
With reference to the case in Fig. 4.4, Tree 1 and Tree 3 can be grouped together into a cluster as measurements 1 and 3 only belong to those trees. The search for the optimal solution for this cluster is independent from the search of the solution for Tree 2. Moreover, as only measurement 2 is present in this track tree, its optimal solution will be equal to the one associated to the lowest score in the tree.

4.5 Global Hypothesis formation

After the clustering process of the tracks, the best global hypothesis shall be searched. Global hypotheses are defined as compatible combinations of track hypotheses. In others words, a global hypothesis is composed by a sequence of measurements-to-track associations for each target, with the constraint that, at each observation time, one measurement can be associated to one target only.

This problem can be formulated either in the form of a Multidimensional Assignment Problem (MAP) or in the form of a Maximum Weighted Independent Set Problem (MWISP). In fact, D. J. Papageorgiou and M. R. Salpukas demonstrated in [32] that a solution to the MWISP is optimal if and only if it is optimal to the corresponding MAP too.
In order to define a general MAP, let's first introduce some useful parameters. Assume that index k denotes the most recent frame and that M_k objects have been detected in that frame. The observation selected at time k to be part of a given track hypothesis over k frames is i_k, where $i_k \in \{0, 1, ..., M_k\}$. Such track hypothesis is associated to the observation sequence $i_1, i_2, ..., i_k$.
As presented in [38], a general k-dimensional assignment problem can be formulated as in Eq. 4.9 in order to determine the most likely combination of tracks at frame k:

$$\min_{z} \sum_{i_1=0}^{M_1} \sum_{i_2=0}^{M_2} ... \sum_{i_k=0}^{M_k} \ell_{i_1 i_2 ... i_k} z_{i_1 i_2 ... i_k} \tag{4.9}$$

$$\text{subject to} \sum_{i_1=0}^{M_1} ... \sum_{i_{u-1}=0}^{M_{u-1}} \sum_{i_{u+1}=0}^{M_{u+1}} ... \sum_{i_k=0}^{M_k} z_{i_1 i_2 ... i_u ... i_k} = 1 \tag{4.10}$$

$$\text{for} \quad i_u = 1, 2, ..., M_u \quad \text{and} \quad u = 1, 2, ..., k$$

$z_{i_1 i_2 ... i_k}$ is a binary variable that indicates whether or not a track hypothesis is selected in the global hypothesis, $\ell_{i_1 i_2 ... i_k}$ is the score associated to a track hypothesis and the variable u indicates the time scan considered in the optimisation process of the M_u tracks.
In case of a negative score function (as in this work with the NLLR), the objective function needs to be minimised. In case of a positive score, the objective function should be maximised instead. The constraint equation in Eq. 4.10 is added to Eq. 4.9 to enforce the compatibility between the track hypotheses constituting the global hypothesis.

The problem in Eq.4.9-4.10 can be equivalently expressed as a MWISP. This problem is formulated by defining an undirected graph $G = (V, E)$ with vertex set $V = \{1, ..., n\}$ and edge set $E \subseteq V \times V$. If the graph contains the edge (i, j), then vertices i and j are said to be adjacent. An independent set is defined as a subset of the vertices whose elements are pairwise nonadjacent, as stated in [32].
The MWISP can be applied to a MHT problem to generate the best global hypothesis. In this contest, each vertex represents a track hypothesis and it is assigned to a weight that corresponds to its track score. For what concerns the edges, they act as constraints between two tracks that share the same observation at the same frame, which are, therefore, incompatible.

This is why finding the best global hypothesis means finding the combination of nonadjacent vertices whose sum of the scores is minimised. This concept is graphically illustrated by Fig. 4.5.

In order to do so, it turns out to be necessary to calculate an incompatibility list between the track hypotheses. As previously mentioned, all the tracks belonging to the same family (track tree) are incompatible with each other as they at least share the observation at their root node. In addition, whenever a measurement is present in two or more track hypotheses at the same time, these hypotheses are incompatible and they can not appear together in the same global hypothesis.
Taking as example the case of Fig. 4.4, Tab. 4.1 illustrates all the parameters needed to compute the global hypothesis adopting a MWISP approach. Each track is named through an ID and it is associated to its measurement-to-track sequence, to an incompatibility list (ICL) with the other tracks and to its score. In addition, each track hypothesis belongs to a family, which is defined by the original root node at the start of the algorithm.

Track ID	**Family ID**	**Measurement sequence**	**ICL**	**Score**
1	1	{1 0}	{2 3}	-4.5
2	1	{1 1}	{1 3 7}	-9.4
3	1	{1 2}	{1 2 8}	-7.1
4	2	{2 0}	{5}	-4.5
5	2	{2 2}	{4}	-9.5
6	3	{3 0}	{7 8}	-4.5
7	3	{3 1}	{2 6 8}	-8.7
8	3	{3 3}	{3 6 7}	-9.2

Table 4.1. Structure of useful data to find the global hypothesis for the problem in Fig. 4.4

With the data in Tab. 4.1, the best global hypothesis $\mathcal{H}_1$ is composed by track 2, 5 and 8, $\mathcal{H}_1 = \{2,\ 5,\ 8\}$. This result can be appreciated through Fig. 4.5, where the numbers inside the circles are the track IDs (vertices of the MWISP), each one associated to the score above them. In particular, the blue circles are the tracks belonging to the best global hypothesis. These are not the only compatible tracks that can be grouped into a global hypothesis, but they are the tracks whose associated score ℓ_1 is minimised. For instance, $\mathcal{H}_2 = \{3,\ 5,\ 7\}$ represents another global hypothesis, with an associated score of $\ell_2 = -25.3$, which is lower than the score of $\mathcal{H}_1$, that is $\ell_1 = -28.1$.
From Fig. 4.4 it is easier to notice track incompatibilities and to identify possible clusters. In fact, it is straightforward to distinguish the two different clusters, as track 4 and 5 are only associated to an edge between them.

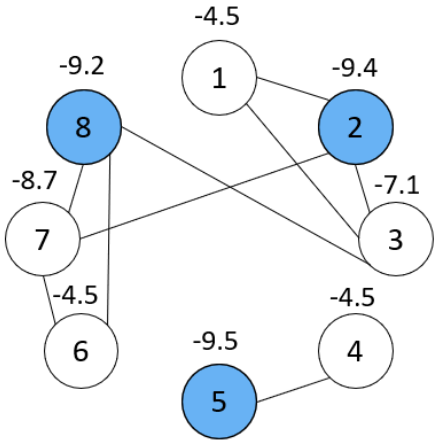


Figure 4.5. Undirected graph associated to the MWISP for the situation in Fig. 4.4

In this thesis the MWISP is solved exploiting its equivalent mathematical programming formulation as an ILP problem, which is defined as:

$$\begin{aligned} &\min_{z} \sum_{i=1}^{M} \ell_i z_i \\ &\text{such that } z_i + z_j \leq 1, \ \forall (i,j) \in E \\ &z_i \in \{0,1\}, \ \text{for } i = 1, ..., M \end{aligned} \tag{4.11}$$

As the rest of the code, the problem in Eq. 4.11 is implemented and solved through MATLAB® [39]. For this particular optimization problem, the function "*intlinprog*" has been exploited [33]. It is an optimization method that allows to find the minimum value of a function in the form of the problem in Eq. 4.12:

$$\min_{x} f^T x \text{ subject to } \begin{cases} x \text{ are integers} \\ A \cdot x \leq b \\ Aeq \cdot x \leq beq \\ lb \leq x \leq ub \end{cases} \tag{4.12}$$

In this thesis contest, the score function plays the role of objective function f. Matrix A embeds all the track incompatibilities , where $A \in \mathcal{R}^{K \times N}$ with K being the number of incompatibility constraints and N the number of tracks at current time. The incompatibility relations of such matrix derive from matrix E, where $E \in \mathcal{R}^{K \times 2}$, which is collecting all the edges (constraints) between the vertices (tracks). Regarding vector b, it is composed by $1s$ and is such that $b \in \mathcal{R}^{K \times 1}$.
Matrix Aeq and vector beq are used to enforce the number of track hypotheses belonging to the global hypothesis, as, in this thesis scenario, the number of targets (nanosatellites) is already known. In particular, Aeq is a vector composed of N $1s$, $Aeq \in \mathcal{R}^{1 \times N}$, whereas beq is a scalar equal to the number of targets.
Vectors lb and up are respectively the lower and the upper bounds of the values assumed by the x vector and they both belong to $\mathcal{R}^{N \times 1}$. For this problem the elements of the x vector can only be equal to 1 or to 0.

For the sake of completeness, the expression of such vectors is reported in the scenario

of Fig. 4.4. Rearranging the incompatibility list of Tab. 4.1 into the matrix E, it indicates the position of the nonzero elements into the A matrix, dealing to:

$$E = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 2 & 3 \\ 2 & 7 \\ 3 & 8 \\ 4 & 5 \\ 6 & 7 \\ 6 & 8 \\ 7 & 8 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

4.6 N-scan pruning

As already mentioned, the strength of an MHT algorithm relies in its capacity to delay the measurement-to-track association whenever it is not unique. In this case, it is capable of storing and propagating different hypotheses of association and their relative states. Due to the exponential growth of the number of hypotheses with time, it is not possible to store all the association history. Only a limited number of hypotheses can be stored.

After clustering, another technique often adopted to reduce the computational heaviness of the TOMHT is called N-scan pruning. This technique reduces the number of track hypotheses to be saved by only storing all the observations up to N time steps backward.
In order to do so, the size N of a time window shall be defined. Its length establishes up to how many time steps backwards the algorithm will store information. Usual values of window size do not exceed $N = 5$ because of the exponential growth of the number of branches once the window is enlarged. The example in Fig. 4.6 is used to graphically explain this technique. Here $N = 3$ is selected as window size.
The first thing to do when applying this technique, is to identify the branches associated to the best global hypothesis at time k. In Fig. 4.6 it is represented by the sequence $\{1 - 1 - 2 - 2\}$ in the dark blue circles.
Then, it is necessary to trace back to the node at time $k-N+1$ of the selected branch. This node will become the new root node. For what regards this example, node 1 at time $t = k - 2$ will be the new root node. Hence, all the sub-trees originating from node 0 and 3 at time $t = k - 2$ will be pruned, whereas, those originating from node 1 will survive.

In other words, what the N-scan pruning technique actually does is to wait N time steps to consolidate the data associations for the observations up to frame $k-N+1$ [38]. This technique is based on the MHT principle that ambiguities at time $k - N$ will be resolved by looking at the information later received.

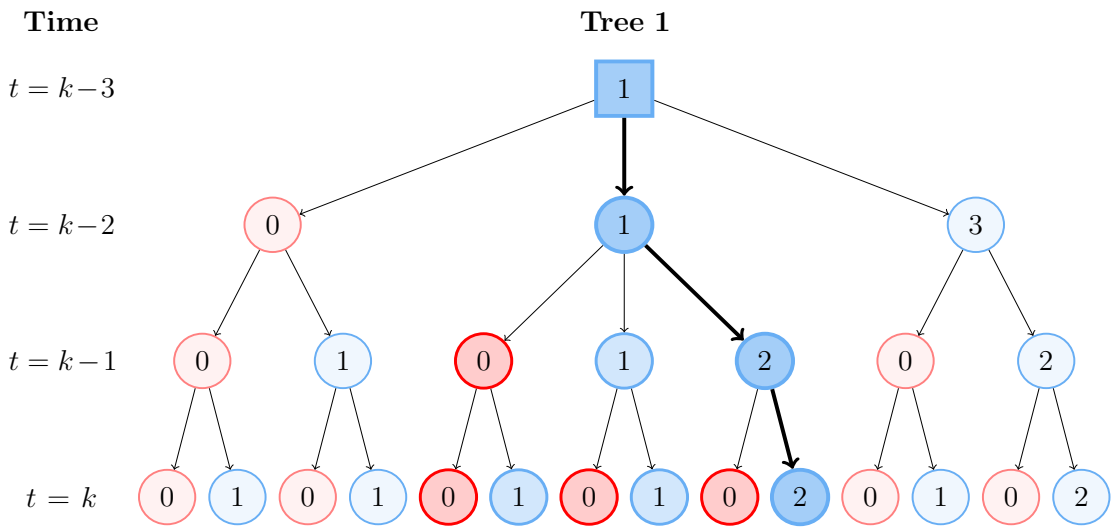


Figure 4.6. Example of the $N-$scan pruning technique

Chapter 5

Testing scenario

The algorithm presented in Ch. 4 will be tested in the case of a satellite formation orbiting around a binary asteroid system. The MHT algorithm will be stressed simulating different initial conditions in order to analyse its behaviour under different mission scenarios. The results of such simulations will then be presented and critically analysed in order to understand if this method can provide real-time satellite tracking for the situation depicted in Ch. 3.

In all the proposed scenarios, the satellite ecosystem will consist of a mothership and a swarm of N nanosatellites orbiting the binary asteroid system of 65803 Didymos. Specifically, the mothership will receive the measurements carried out by the nanosatellites and, through the use of the aforementioned TOMHT implemented on-board, it will try to estimate position and velocity of all the nanosatellites in the swarm.
The state of the mothership is assumed to be perfectly known and its orbit is further to the asteroids with respect to those of the swarm. At the beginning, it is also supposed that the position of the asteroids is exactly known by the mothership. An error on the asteroid position will then be introduced, to analyse the effect on the satellites' state estimation process.

Lastly, the information sent by the nanosatellites to the mothership will be discussed. In a general MHT application, a radar is employed to look for targets in the observation space when these bodies are not sending any signal to the station. Hence, there is usually no information regarding the a priori association between measurement and satellite.
For the case treated in this thesis, this information could either be present or not. So, in the case such data was already known, why implementing such algorithm?
In case of closely-spaced nanosatellites, the a priori measurement-to-track association might not provide the best state estimation due to the intrinsic noise of the measurement process. For example, let's assume to have two satellites; when they are very close to each other and their state uncertainty overlap due to the measurement error, the measurement set generated by satellite 1 could happen to estimate the state of satellite 2 better than measurement 1.
Thus, for this thesis scenario, the presence of a satellite identification number associated to its measurement would not be beneficial for the scope of this work. For this reason the developed algorithm does not account for the presence of any satellite ID along with the measurement set.

5.1 Parameters definition

In this section the value of the main parameters used in the following simulations will be presented along with the reason behind their choice. Most of the simulations will adopt these values. However, it will be explicitly stated whenever values different from the ones below are used.

First, the tuning parameters of the EKF are introduced.
Starting from the process noise covariance matrix $\mathbf{Q}(k)$ at time k, it has the shape of Eq. 5.1:

$$\mathbf{Q}(k) = \begin{bmatrix} Q_q \cdot dt^3/3 & Q_q \cdot dt^2/2 \\ Q_q \cdot dt^2/2 & Q_q \cdot dt \end{bmatrix} \quad \mathbf{Q_q} = \left(q \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^2 \tag{5.1}$$

where the q parameter is $q = 5 \cdot 10^{-5}$. For what concerns the measurement noise covariance matrix $\mathbf{R}(k)$:

$$\mathbf{R}(k) = \begin{bmatrix} {R_m}^2 & 0 & 0 \\ 0 & {R_m}^2 & 0 \\ 0 & 0 & {R_m}^2 \end{bmatrix} \tag{5.2}$$

with $R_m = 0.1$, whereas the error covariance matrix is:

$$\mathbf{P}(k) = \begin{bmatrix} {P_p}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & {P_p}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & {P_p}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & {P_v}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & {P_v}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & {P_v}^2 \end{bmatrix} \tag{5.3}$$

where $P_p = 0.01$ and $P_v = 0.001$.
Another class of parameters plays an important role in the tuning of the TOMHT algorithm. They are all the parameters appearing in the track score expressions in Eq. 4.6-4.8, i.e. the detection probability $P_D(k)$ and the spatial density of extraneous measurement λ_{ex}. The former parameter indicates the probability that the instruments correctly perform the measurement and that it falls inside the expected gate. This is set to $P_D = 0.999$.
In order to assign λ_{ex}, it is necessary to define some other parameters first. In Chapter 4.D of [31], it is defined that $\lambda_{ex} = \lambda_\phi + \lambda_\nu$. The spatial density of false measurement, λ_ϕ, is defined as in Eq. 5.4, where P_{FA} is the false alarm probability and V_{cell} is the resolution cell of the instrument. The spatial density of new target, λ_ν, instead, is defined in Eq. 5.5, with n_ν being the number of new targets expected in a frame/scan and V the measurement space.

$$\lambda_\phi = P_{FA}/V_{cell} \tag{5.4}$$

$$\lambda_\nu = n_\nu/V \tag{5.5}$$

The probability of false alarm depends on instrument reliability. In literature, the values vary from 10^{-3} to 10^{-10}, so here $P_{FA} = 10^{-6}$ is selected and the resolution of

the instrument is set to $V_{cell} = 1\ m^3$. The measurement space is defined as the space in which the instruments are able to receive the measurement from. Due to the physical characteristics of the Didymos binary system, this is selected to be $V = 10^9\ m^3$.
Given the fact that the number of targets is already known and that no other targets are expected to enter the scene, the spatial density of new targets is set to a very low value, $n_\nu = 10^{-12}$.
All the simulations will be run with a laptop equipped with an Intel(R) Core(TM) i7-7700HQ processor.

5.2 General considerations

Before presenting the different scenarios alongside the associated results, it is useful to analyse the capabilities of the filter. Let us consider a system where only one nanosatellite is present along with the mothership, and then study the behaviour of the filter. This case study is useful to test filter accuracy and to look for possible dangerous relative positions between the satellites and the asteroids.
Going back to the TOA equations (Eq. 3.8-3.9-3.10) in Ch. 3.3, their analytical solutions depend upon the relative position between the mothership and the asteroids. In particular, when the vectors connecting the mothership to the asteroids are aligned, the state could be poorly estimated. Therefore, some anomalies could arise from situations like alignments or coplanar motion between the bodies.

5.2.1 Mothership-asteroids alignment

The first situation under investigation is represented in Fig. 5.1. It illustrates the motion of the nanosatellite and the mothership having no particular relative initial conditions: bodies are not aligned and satellites do not orbit along the same plane.

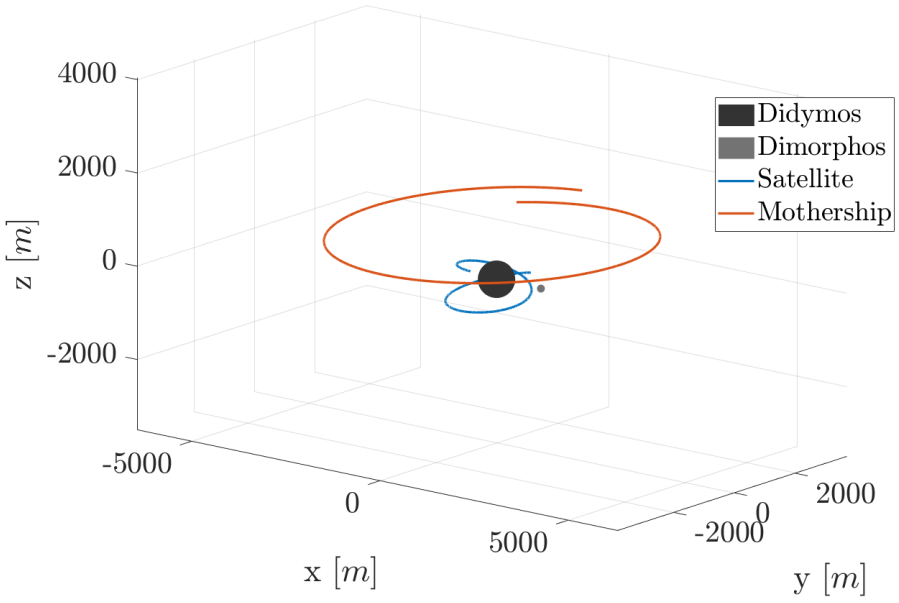


Figure 5.1. Satellites' trajectories with random initial conditions

The blue trajectory is the nominal trajectory of the nanosatellite. It will be compared to the estimated one in order to evaluate the performances of the filter.
As for all the other simulations, the trajectories of the satellites are not controlled. As the scope of this thesis is just to test whether or not the TOMHT method is able to reconstruct satellites' state, it is not beneficial to implement a control low and to define specific closed trajectories.

The motion of the nanosatellite is simulated for around 7 h using the simulation parameters defined in Ch. 5.1. The results, in terms of position estimation error, are shown in Fig. 5.2.

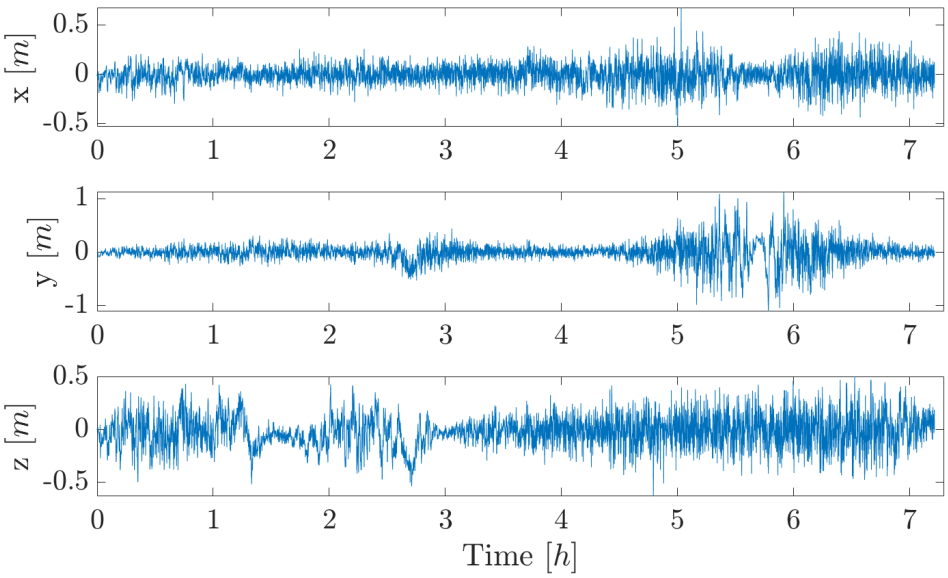


Figure 5.2. Nanosatellite position estimation error for the scenario in Fig. 5.1

Fig. 5.2 shows that the algorithm is capable of estimating the motion of the satellite with an error lower than 0.5 m on each axis for most of the time.
However, in the plot of the y-component of nanosatellite's position, it is interesting to notice the increase of the noise in the region between 4.5 h and 7 h. This is a consequence of the relative position assumed by the mothership and the asteroids during that time interval.
This is indeed confirmed by Fig. 5.3. In here it is depicted the evolution of the norm of the cross product between the vectors going from the mothership to main and secondary asteroid, respectively $\mathbf{r}_A$ and $\mathbf{r}_B$. In the time window between 4.5 h and 7 h, this value approaches zero. In other words, the mothership is almost perfectly aligned with the two asteroids.
Hence, as expected developing the TOA equations in Ch. 3.3, the alignment between mothership and asteroids deteriorates nanosatellite's state estimation. Nevertheless, the estimation error on the y-component in this critical region is still below 1 m. Moreover, the error decreases as the value of $||\mathbf{r}_A \times \mathbf{r}_B||$ increases.
However, in case the mothership and the two asteroids are perfectly aligned, with $||\mathbf{r}_A \times \mathbf{r}_B|| = 0$, the error is expected to be higher. Such case will be analysed in Ch. 5.2.2.

A plot of the complete state estimation error comprising the error on the velocity components (Fig. A.1) can be found in Appendix A.

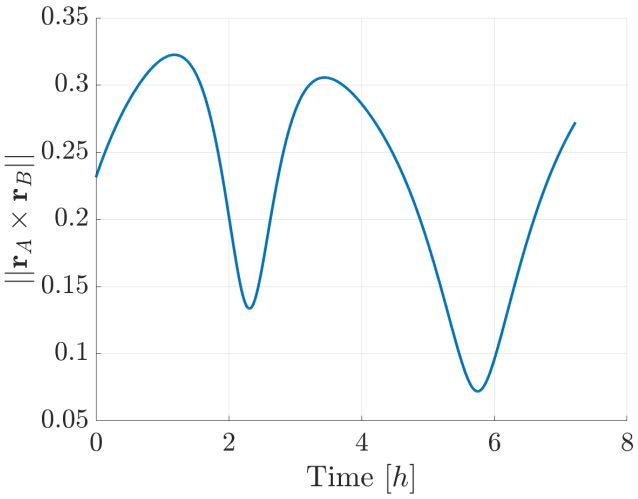


Figure 5.3. Norm of the cross product between the vectors going from the mothership to the asteroids

5.2.2 Coplanar motion

Another situation that could cause a coarse state estimate is analysed in this section. It is the case in which the mothership and a nanosatellite are both moving inside the plane characterised by $z = 0$ (referring to the reference frame defined in Fig. 3.1). This means that alignments between satellites and asteroids are possible and more frequent with respect to other general configurations. Consequently, in such regions, an accuracy loss could be experienced.

For this reason, the coplanar motion of the mothership and the nanosatellite around the asteroids is simulated, and the accuracy in the estimation of nanosatellite's state is analysed.
The trajectories of the satellites are represented in Fig. 5.4 during the 3 h of simulation. The results of this simulation, in terms of position estimation error of the nanosatellite, are presented in Fig. 5.5.
The first thing that can be noticed when looking at Fig. 5.5, is that the position components are estimated with different precision. The z-component is the one experiencing the highest error with respect to the others, reaching a maximum error of 20 m. This means that, in a coplanar motion scenario, the algorithm is not able to correctly estimate the z-component of satellite's position and, therefore, satellite's motion too. This is due to the fact that the measurement set used to update the state of the satellite does not include any information about the z-coordinate, as all the bodies lay on the plane defined by $z = 0$.

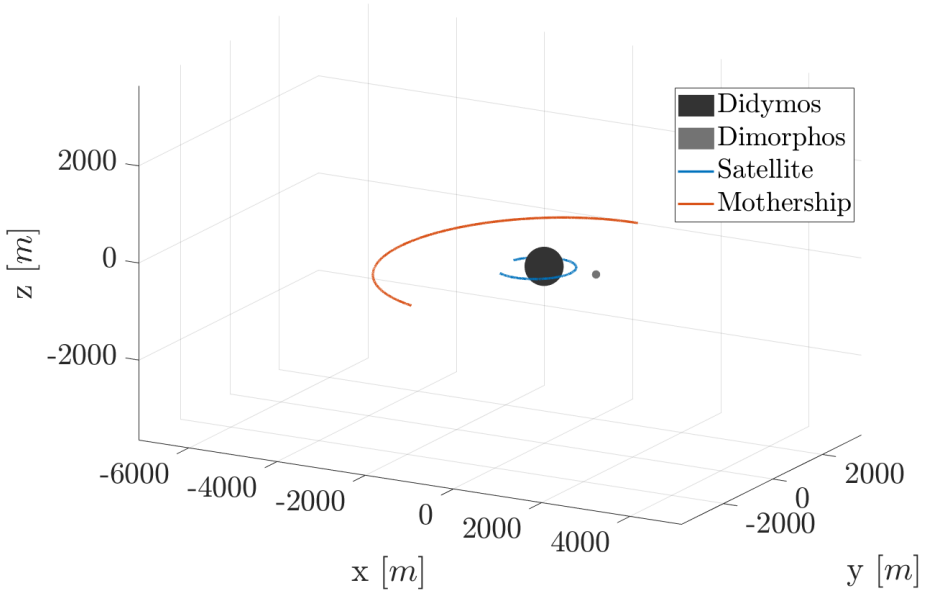


Figure 5.4. Trajectories of mothership and nanosatellite during their coplanar motion

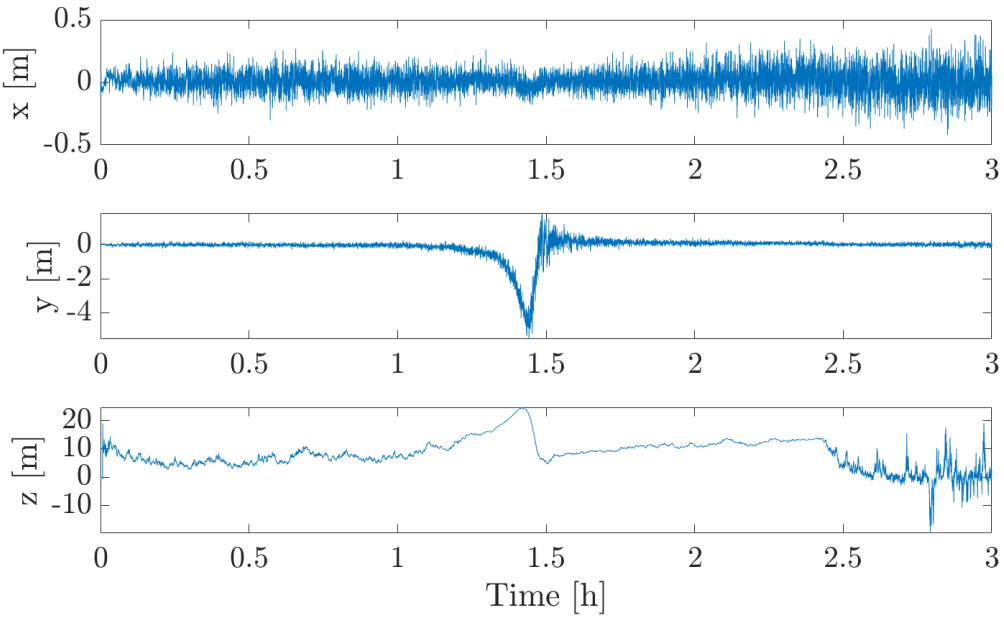


Figure 5.5. Nanosatellite position estimation error for the scenario in Fig. 5.4

For what concerns the x-component, it is the one estimated with higher precision with respect to the others. In fact, its error is around $0.2 - 0.3\ m$. Moving to the y-component, its estimation error increases from $0.2\ m$ in the region before $t = 1.1\ h$, up to $5.5\ m$ when $t = 1.45\ h$. Why does the error increase so much?
The answer to this question relies on the relative position between the bodies. As it can be appreciated from Fig. 5.6, the error growth coincides with the time instant in which the mothership and the asteroids are aligned. So, as it was for the case

associated to Fig. 5.2, when such alignment occurs, the estimation of the y-component of satellite's position looses accuracy (as the bodies are aligned along the x-axis and the y-components of the three bodies' position are zero).
In this particular case, the error increases more than in the case in Fig. 5.2 as the alignment is perfect here, i.e. $||\mathbf{r}_A \times \mathbf{r}_B|| = 0$.

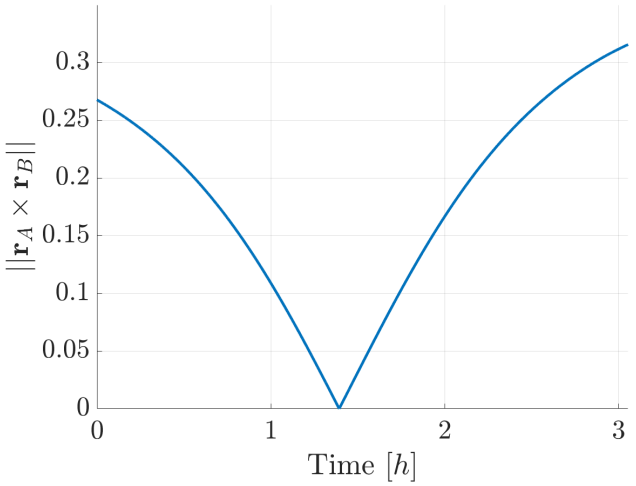


Figure 5.6. Norm of the cross product between the vectors going from the mothership to the asteroids

The alignment also introduces a significant error in the z-component and a very small variation in the x-component.
In conclusion, the problems arising from a coplanar motion are mainly related to the lack of information regarding the z-components of the performed measurements. This makes the estimation of such satellite's component very inaccurate. In fact, the z-component of satellite's position is estimated with an error of around 10 m. In addition, the eventuality of a mothership-asteroids alignment introduces a further uncertainty in the estimation of y-component and, in turn, of the other components. For these reasons, a coplanar motion scenario should be avoided.

5.3 Satellites encounter

Let's now move to the evaluation of the performances of the TOMHT method in different mission scenarios.
The first scenario under investigation is the case in which two or more satellites have a close encounter within a certain time interval during their trajectories around the asteroids. Their position during this time window is such that their validation regions overlap. Consequently, the measurement-to-track association is not unique and different updated states are propagated using different measurement sequences.
Three nanosatellites are employed for the examined scenario. Their orbits, here propagated for 2 hours, are shown in Fig. 5.7.

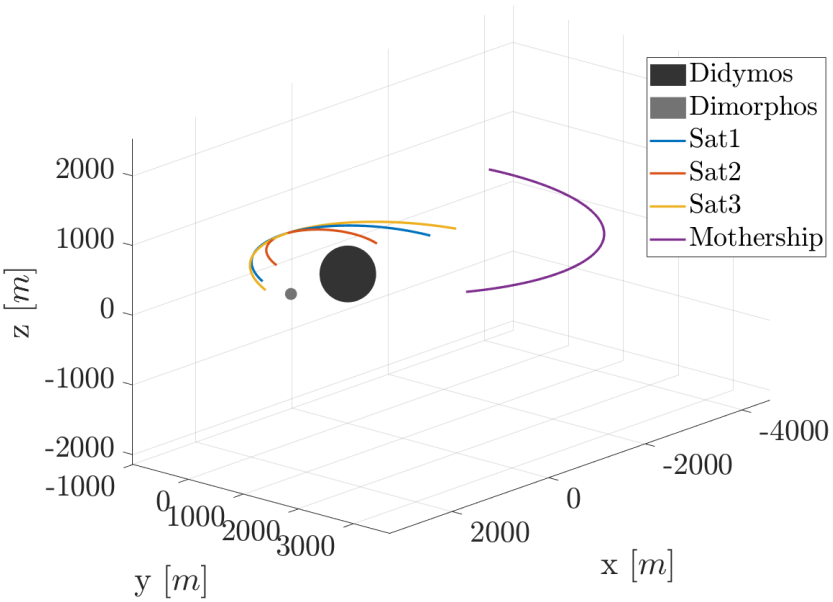


Figure 5.7. Satellite trajectories for the case of close encounter

The three satellites are following their paths around the asteroids until all of them get very close to each other for a certain time window. The length of this interval depends on the size of the gates that, in turn, depends on the selected confidence level α introduced in Ch. 4.1. The higher the value of α, the larger the gates. For these simulations $\alpha = 0.999$ is selected.

What we expect from the TOMHT algorithm in this case is that, in the region of closely-spaced targets, there could be some measurements that are associated to a satellite that is not the one who generated such measurement set. On the contrary, for the rest of the simulation, we expect that the states of the satellites are updated using the measurement set they generated.

The first simulation is performed setting a pruning window of length $N = 2$ and a simulation time of 60 s. The simulation is carried on in a way that, after 30 s, the nanosatellites experience their closest approach and, after that, they drift away.
Considering the case in which a measurement set is sent to the mothership at every second, the results in terms of norm of the error in the position and velocity estimation are presented in Fig. 5.8. The results here illustrated are obtained displaying satellites' state that are associated to the global hypothesis with the highest likelihood at each time step, i.e. the best global hypothesis.
In this situation, the TOMHT algorithm is capable of computing the position of the satellites with an error lower than 0.5 m throughout all the simulation.
By looking at Fig. 5.8 it can be pointed out that the state estimate is not available for the last seconds of the simulation. This is due to the basic idea behind the MHT method: all the ambiguities at time k can be resolved by looking at the information received $N+1$ time steps later. The variable N is the length of the time window used in the pruning phase. As $N = 2$ is adopted for this simulations, the algorithm only confirms the states estimation up to $t = 57$ s.

This also means that the confirmed information about satellites' state will always be available after $N+1$ time steps with respect to the moment in which the measurement set is received. Nevertheless, the global hypotheses with the state information at current scan are still available. However, they are not yet confirmed, so they can change after successive observations are received by the mothership. In other words, they could be less accurate with respect to the global hypothesis computed after $N+1$ scans.

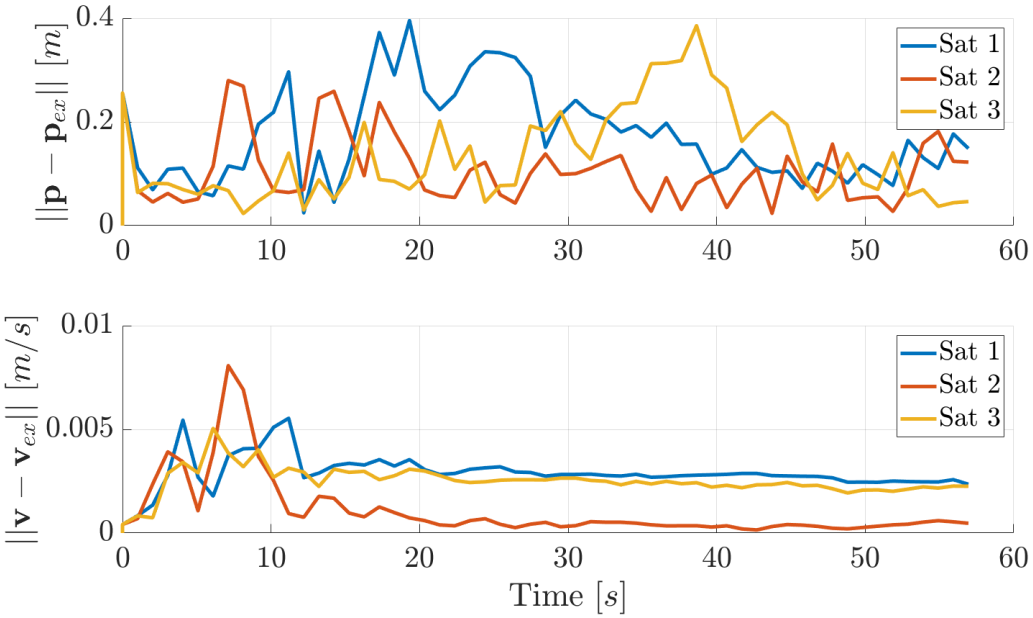


Figure 5.8. Norm of the position and velocity error of the three satellites

Another interesting aspect that can be analysed is represented by the measurement sequences associated to the best global hypothesis, illustrated in Fig. 5.9.

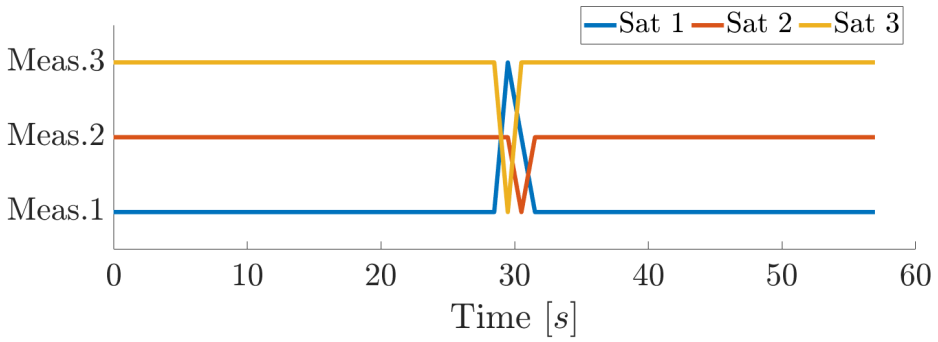


Figure 5.9. Measurement sequence of the best global hypothesis for $N = 2$

On the y-axis there are indicated the measurement sets coming from the satellites. The plot shows the association history between satellites and measurement sets used to update satellites' state belonging to the best global hypothesis.
The trajectories of the satellites are such that satellites' validation region overlap between 27 s and 35 s. As expected, the observations are associated to the spacecraft

they are coming from before and after this time window. However, when satellites are very close to each other, it is possible to notice that their states are not always updated with the measurement they generated. It happens because this association brings to a more precise state update and, consequently, to a higher association likelihood, increasing the track score in Eq. 4.6.
In a situation like this, it is important to notice that, after the close approach, the algorithm does not confuse the trajectory of a nanosatellite with the trajectory of another nanosatellite. Hence, it is able to correctly keep estimating satellites' motion.

One important thing to always keep in mind when dealing with a MHT method is the computational time needed to estimate targets' state. As already mentioned in Ch. 2, it is known that this method is highly computationally expensive. In order to perform real-time target tracking, the time needed by the algorithm to compute the set of global hypothesis shall always be lower than the elapsed time between two consecutive measurement sets are received by the mothership.
With the pruning parameter set to $N = 2$, the algorithm enables a real-time tracking for the case of Fig 5.8 and Fig 5.9, as the computational time is very low.

The results of the simulation can still be improved by enlarging the pruning window. However, a trade-off between computational time and accuracy of the results shall always be performed.
What we expect when increasing the length of the pruning window, is a possible change in the measurement-to-track association sequence and a consequent improvement of the state estimation.
This is exactly what happens when the previous simulation is repeated with a pruning window characterised by $N = 3$. Fig. 5.10 shows the outcome in terms of measurement sequence. What can be noticed is a different measurement sequence for satellite 2 and satellite 3 with respect to the previous case. In particular, only one observation-to-track association is different, and it occurs soon after $t = 30\ s$.

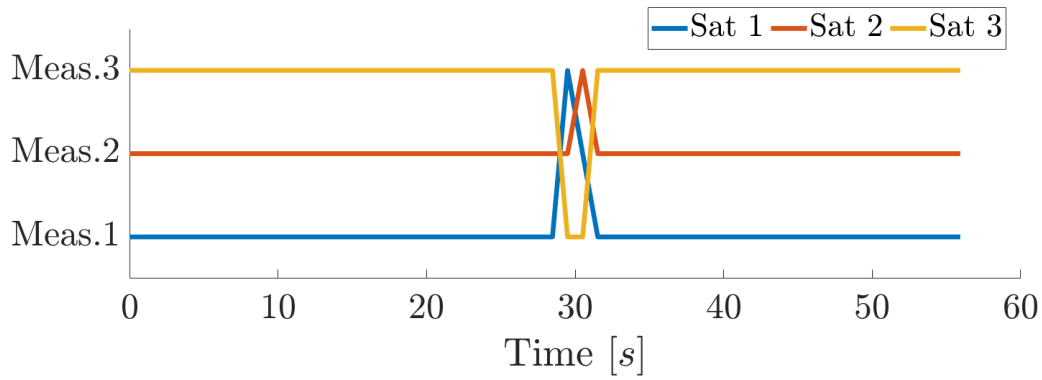


Figure 5.10. Measurement sequence of the best global hypothesis for $N = 3$

The consequences deriving from this difference are illustrated by Fig. 5.11. The first subplot of such figure highlights that an increment in the size of the pruning window corresponds to a reduction of the state estimation error of satellite 2. Obviously, the improvement is experienced because of the different measurement sequence employed this time to update the state of satellite 2, which happens to better estimate its real position. As the sequence associated to satellite 1 is unchanged, the results related to

this satellite do not experience any change and, for this reason, they have not been included in Fig. 5.11.

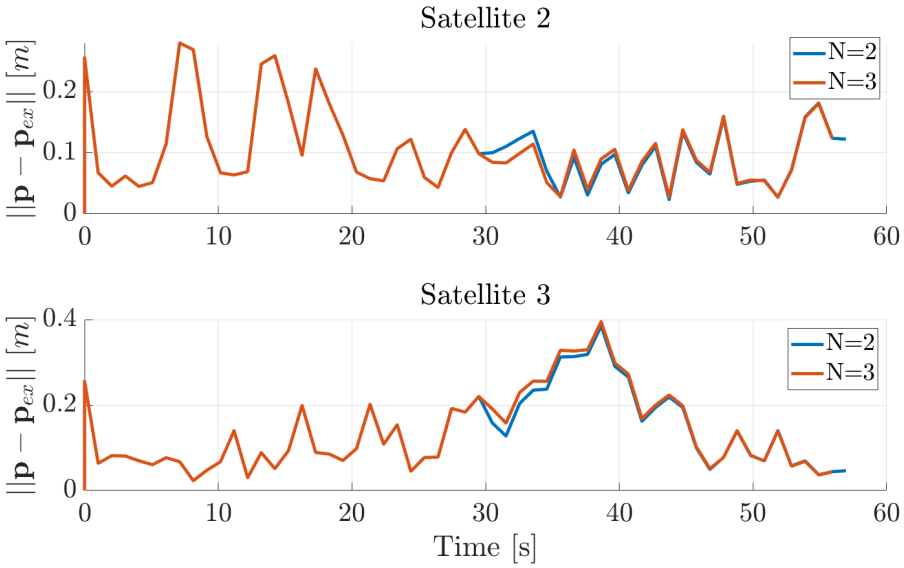


Figure 5.11. Norm of the position error of satellite 2 and satellite 3

Contrarily to what happens for satellite 2, in the second subplot of Fig. 5.11, the state estimation error of satellite 3 computed using $N = 2$ is lower than the one calculated with $N = 3$. Why does it happen?
The answer to this question relies in the expression of the score function in Eq. 4.5. In fact, through the score associated to the global hypothesis, the TOMHT method minimises the score corresponding to a combination of targets' track hypotheses. In other words, this algorithm chooses as best global hypothesis the combination of measurement-to-track association whose sum of the scores is minimum. Due to tracks incompatibilities, the best combination of tracks' score is not always equal to the sum of the best tracks' score. As a consequence, it could happen that the global estimation error of the three satellites decreases by enlarging the pruning window, even if the state estimate of a specific target is less precise with respect to the case with a shorter pruning window.
So, this algorithm minimises the estimation error in a global manner, intended as the sum of the estimation error of all the three satellites.
For this reason, the results associated to satellite 3 in Fig. 5.11 are coherent with the idea behind a TOMHT method, in the sense that the global estimation error using $N = 3$ is improved.

Even in this case, the computational time needed by the filter is quiet low and it still permits to perform real-time satellite tracking. A further increment of the pruning window size would not turn into any precision enhancement for this special case. Actually, it would only exponentially increase the computation time, making the real-time satellite tracking unfeasible.

Additional considerations about the variation of the computational time due to a change of the pruning window size or due to a higher number of satellites can be

found in Ch. 5.4.1.

5.4 Nanosatellites deployment

A different scenario is investigated in this section. The following results are referred to the case in which the nanosatellites of the swarm are initially very close to each other and then, due to a slight difference in their velocity components, they drift away from each other.
This scenario is built to simulate the case of deployment of the swarm from the mothership and the nanosatellites orbital injection. This is a particularly challenging situation for a TOMHT algorithm, as there can be a relevant number of satellites occupying the same portion of space for a relatively long time. For this reason, this particular scenario will be used to perform an analysis to understand how the computational time needed by the filter varies due to a variation of the number of targets and of the pruning window size.

For this scenario it is assumed that, after the nanosatellites are deployed from the mothership, some time passes before they start drifting apart. The simulation begins with the mothership orbiting the asteroid with a larger semi-major axis with respect to those of the orbits of the nanosatellites. The swarm satellites are assumed to have the exact same initial position, but different velocity components at $t = 0\ s$. Of course, it is not a physically compliant initial condition. However, this condition can be more critical for the tracking algorithm as nanosatellites' gates are all overlapped and the number of possible measurement-to-track combinations is maximum.
A first simulation is run considering 4 swarm satellites whose trajectories after the deployment are shown in Fig. 5.12. The results will be analysed under the point of view of the measurement sequence and of the estimation error.

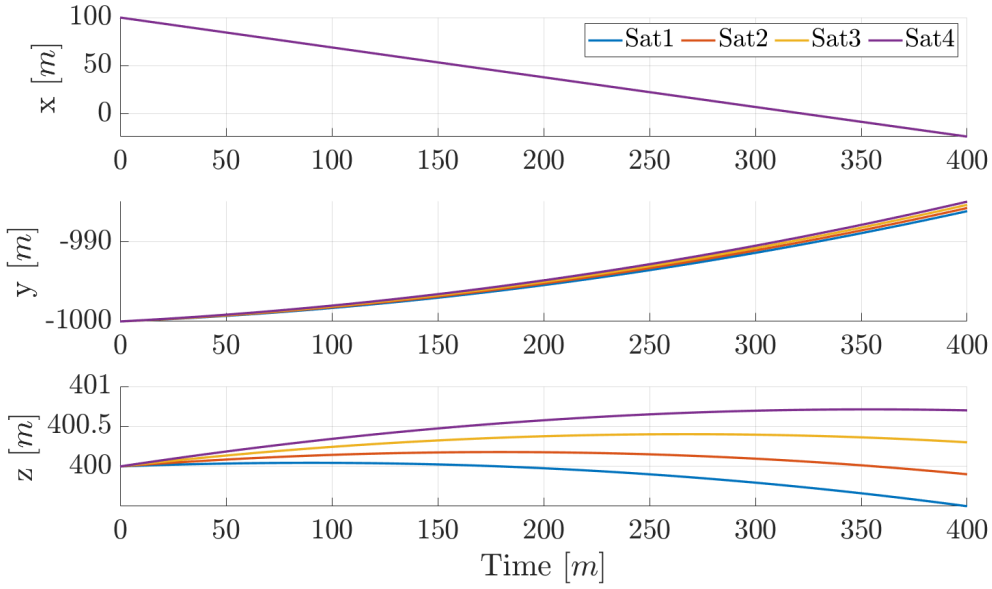
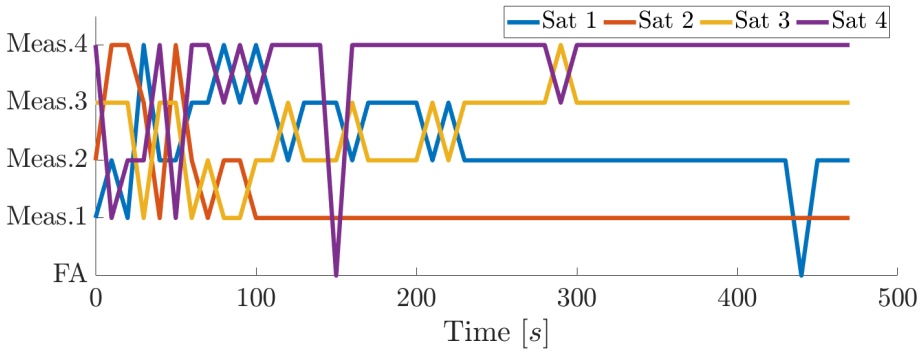


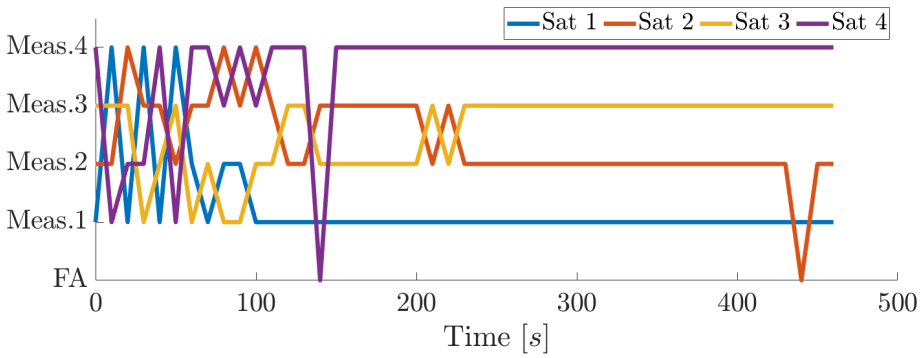
Figure 5.12. Satellites' coordinates after the release

Talking about the measurement sequences used to update satellites' states, what we expect to experience is an initial phase in which the assignments are not "stable": it means that states of the satellites could be updated using measurements coming from other satellites. This could happen in the initial phase, when the gates overlap as the predicted satellites' states are very close to each other.
After this stage, the filter should be capable of distinguishing the different satellite trajectories, i.e. updating a satellite's state using the measurement set generated by the satellite itself.
As the satellites are expected to be very close to each other for a significant time interval, their gates are selected to be smaller than they were in the previous cases, in particular $\alpha = 0.9$. The reason behind this choice is that, by reducing the gate size, the number of measurements falling within more than one gate will decrease, and the same will happen to the number of possible measurement-to-track combinations. Consequently, the computational time needed by the algorithm to estimate satellites motion will decrease.
The drawback of this choice is an increased probability that a measurement set falls outside the validation region of its satellite if there is no possibility to increase its precision. This would turn into a false alarm.

The simulation is run assuming that the mothership receives a measurement set every 10 s for 500 s. The measurement-to-track association history for a pruning window characterised by $N = 2$ and $N = 3$ are respectively represented in Fig. 5.13a and Fig. 5.13b.



(a) $N = 2$



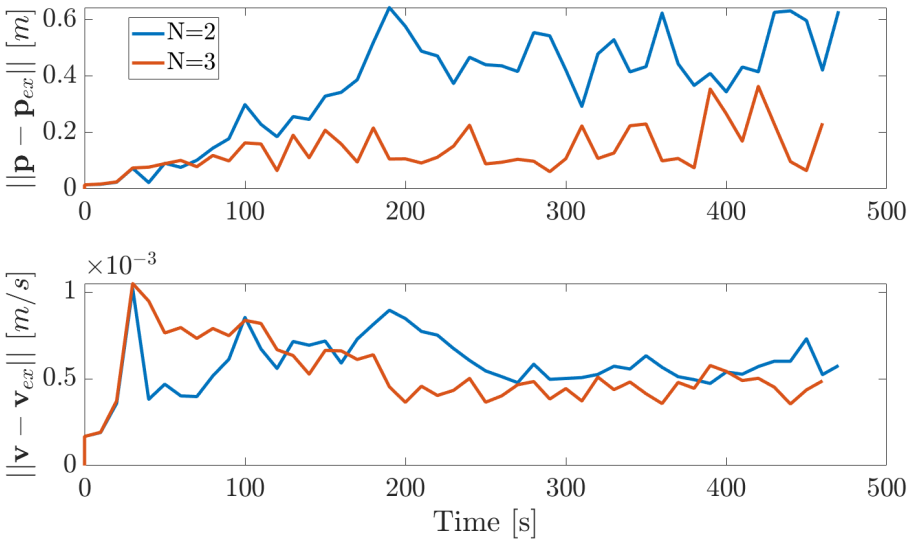
(b) $N = 3$

Figure 5.13. Measurement sequence for the deployment scenario

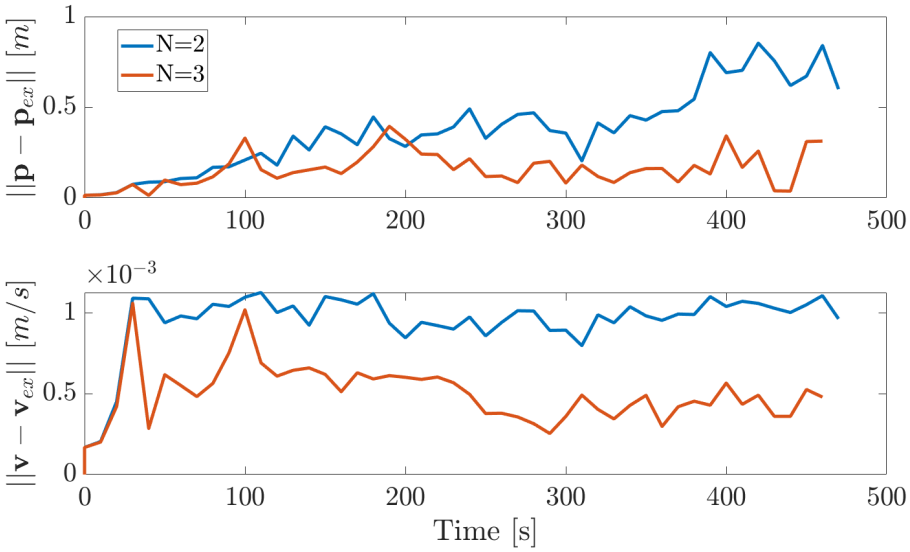
Each satellite is here represented by a colour and it is related to the measurement on the ordinates. On the y-axis, FA stands for false alarm, which occurs whenever an observation does not fall within the validation region. The presence of false alarms in the measurement sequence of the best global hypothesis represented in Fig. 5.13, means that the update phase of the EKF for that track has not been performed at the time instants in which FA is present in Fig. 5.13. It happens if the score function associated to the other measurement-to-track associations is lower (in absolute value) than the one associated to the false alarm, or if none of the observations fall within a target's gate.
As expected, in the beginning of both Fig. 5.13a and Fig. 5.13b, satellites' states are updated with different observations at every time. After this time interval, the associations become more stable.
However, in Fig. 5.13a it is possible to observe that, after the initial phase, the trajectory of satellite 1 is still updated through the measurements coming from satellite 2 and vice versa. It is the same as saying that the algorithm has swapped the trajectory of satellite 1 with the trajectory of satellite 2. This is because satellite 1 was initialised as the target associated to measurement 1, whereas satellite 2 was associated to measurement 2.
This problem can be overcome by enlarging the pruning window. In fact, adopting $N = 3$ for the pruning window as in Fig. 5.13b, brings to the expected observation sequence for all the nanosatellites, without any mix-up.

The difference in terms of state estimation error between the two previous cases can be appreciated through Fig. 5.14. Here only the results related to satellite 1 and satellite 2 are presented because a significant difference arises in their estimation when increasing the pruning interval. This plot illustrates the estimation error in terms of norm of the difference between estimated position and real position, and the same for the velocity.
In both Fig. 5.14a and Fig. 5.14b targets' positions computed with the shorter pruning interval tend to diverge from the exact position. It is due to the fact that the user expects to receive the estimates of satellite 1, whilst he is receiving those of satellite 2 and vice versa.
In a 500 s simulation, the errors in the case of $N = 2$ remain quit low because the two satellites are still close to each other. However, by increasing the simulation time, the satellites would drift away and the error would consequently uncontrollably grow. In this situation, increasing the information history stored by the algorithm helps solving this problem. In fact, Fig. 5.14 illustrates that satellites estimate improve when $N = 3$, both in terms of position and velocity. This solution does not diverge anymore because the algorithm has not confused the trajectories of the two satellites, as already visible by looking at Fig. 5.13b.

Both the previous configurations, with $N = 2$ or $N = 3$, allow a real-time tracking as the time needed by the TOMHT algorithm is lower than 10 s, the time that passes between two consecutive measurement sets generation.



(a) Satellite 1



(b) Satellite 2

Figure 5.14. Norm of the position and velocity estimation error with different pruning windows lengths N

5.4.1 Computational time

The scenario represented by the swarm deployment is exploited to perform additional analysis on the computational time needed by the filter. In this section it will be investigated how the time needed by the TOMHT algorithm to find the best global hypothesis varies due to a variation of the number of swarm satellites and to a variation of the pruning window length.
The aim of this study is to understand whether or not there exists a maximum number of swarm nanosatellites that the filter can manage in order to still provide their real-time trajectory tracking.

In order to do so, different simulations with a varying number of satellites are run and the time needed by the algorithm to run each time step is monitored. Additionally, the number of incompatibilities between the track hypotheses is stored at each time. This variable represents the number of constraints in the solution of the optimization process in Eq. 4.12. In particular, the number of rows of the $\mathbf{A}$ matrix is equal to the number of incompatibilities. As a general rule, when the number of overlapping gates increase, the same happens for the number of incompatibilities.

Such simulations are characterised by a simulation time of 500 s, with 21 observations received by the mothership during this time interval. The number of measurement sets received by the mothership is reduced with respect to the case in Ch. 5.4 in order to decrease the total computational burden of the simulations.
For these simulations the pruning window is set to $N = 2$, whereas the number of swarm nanosatellites varies from 2 to 7. All of the satellites start their motion in the same point and with a slight difference in the velocity components.
The results of the simulations are reported in Fig. 5.15. In each of the plots in Fig. 5.15 a different number of nanosatellites is chosen. The plot compares the number of track incompatibilities with the computational time needed to find the best global hypothesis at time k when the k-th measurement set on the x-axis is received.

The first thing that can be noticed from Fig. 5.15 is that, as expected, when increasing the number of satellites, the time needed to process a measurement set and to generate a global hypothesis increases too.
The most important thing that should be observed is that the number of incompatibilities is proportional to the CPU time. It means that the time to run the algorithm is not strictly related to the number of satellites, but it is proportional to the number of incompatibilities between track hypotheses instead.
For this reason it does not make sense talking about an upper limit in the number of swarm nanosatellites that the algorithm can deal with. This number depends on the trajectories of the satellites, on how many overlapping gates are present at the each time instant and, therefore, on the number of track incompatibilities.
Additionally, the number of satellites that the filter can manage, also depends on the elapsed time between two successive observations. The higher this time interval, the higher the time available by the filter to process the current information.

The last thing that can be highlighted about Fig. 5.15, is represented by the variation of the number of incompatibilities. This value reaches the maximum at the beginning of the simulation before the first pruning process is performed. It happens because, for some initial time instants, the validation regions of all the satellites occupy the same portion of space. In other words, all the satellites' gate are overlapped and the number of possible measurement-to-track associations is the highest possible. Additionally, the first pruning phase is only performed when $N + 1$ scans of information are stored. After that, the number of incompatibilities significantly decreases also because the satellites are drifting away from each other, and the same happens for their gates.

When the pruning interval is enlarged to $N = 3$, for example, the number of incompatibilities are of course higher then the case with $N = 2$ and same number of satellites. As a consequence, the necessary computational time increases too, as can be appreciated through Fig. A.3 in Appendix A.

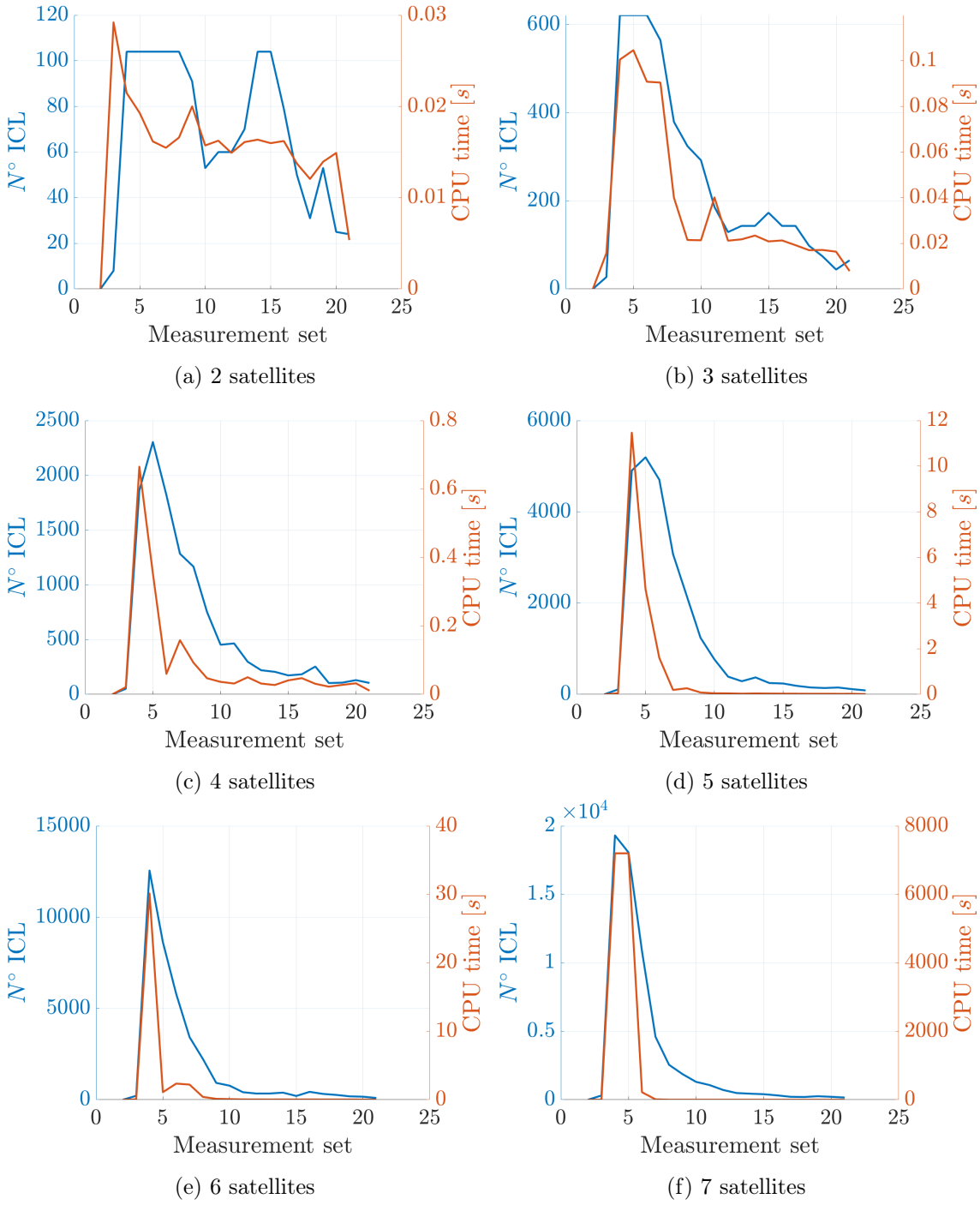


Figure 5.15. CPU time vs number of incompatibilities at each time step if $N = 2$

Given the results in Fig. 5.15, it is possible to compare the total time that the algorithm takes to run a simulation employing a different number of satellites in case the size of the pruning window is equal to 2. The outcomes are reported in Fig. 5.16. The time is here expressed through a logarithmic scale on the y-axis, whereas the x-axis indicates the number of satellites employed in the simulation. The trend of the CPU time when the number of satellites increases is approximately linear, which

means that the running time of the algorithm exponentially increases with the number of satellites.
For the nanosatellite deployment scenario under discussion, the TOMHT algorithm takes more than 3 hours to simulate 8 minutes of motion of 7 satellites. Most of this time is spent in correspondence of the firsts measurement sets, i.e. before the first pruning and when the nanosatellites are all very close to each other.
Of course the situation with 7 nanosatellites does not allow a real-time target tracking. In order to guarantee real-time time tracking capabilities, the filter shall estimate targets' state within a time interval of 23 s, the elapsed time between two successive observations. With these parameters and in this particular scenario, the TOMHT method can deal with no more than 5 satellites. In fact, by looking at Fig. 5.15d, all the satellites estimations are computed within the limit time of 23 s.

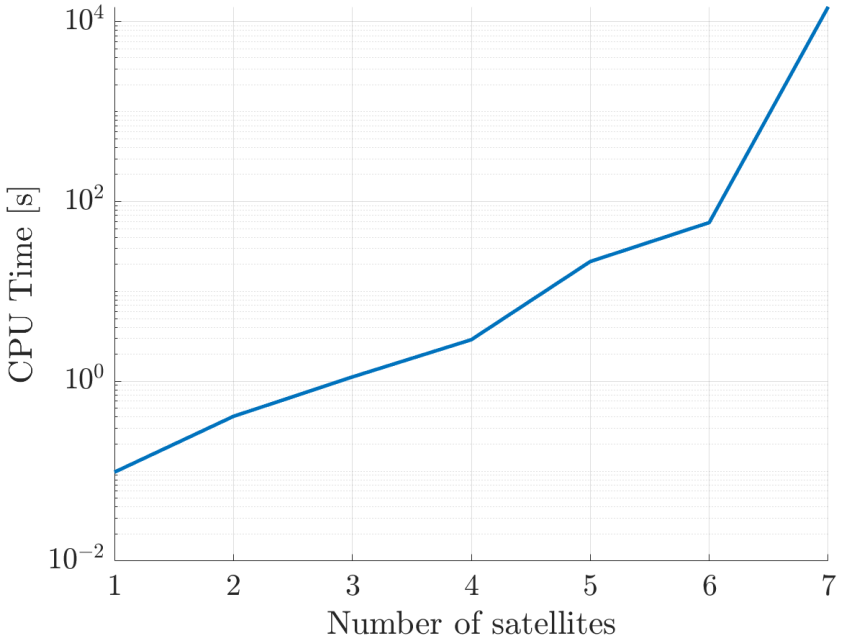


Figure 5.16. Logarithmic scale of the CPU time vs number of satellites

As previously mentioned, the bottleneck to perform real-time target tracking is represented by the number of constraints between track hypotheses, not by the number of satellites itself. Of course, the number of satellites plays a role in the number of track constraints, but it is not the only one. In fact, the number of maximum incompatibilities that the filter can manage also depends on other factors like satellites trajectories, the observation schedule defined in the mission design phase and the gate size (the α parameter).
For this reason it does not exist a general rule that defines the maximum number of satellites that the filter can work with at the same time. In fact, a swarm could be, for example, composed of tens of satellites, but if they only get closer in pairs, it is not a problem for the TOMHT. Hence, even if the number of satellites could seem high, the number of incompatibilities is quite low as satellite gates only overlap in pairs.

5.5 Asteroid position error

The previous simulations were all based on the assumption that the mothership exactly knows the position of the asteroids, but what happens if it is not true anymore? How does an error on the knowledge of asteroids positions affect the state estimation of a satellite? The answer to these questions will be covered in this section.

This analysis is performed taking as example the previous scenario in which some satellites occupy the same initial position and they then drift away due to a difference in their velocity components.
The error is inserted as a random Gaussian with zero mean value and standard deviation σ_{ast}. In this way, the estimated trajectories of the satellites are integrated using different asteroids positions values at each time step.

The sensitivity of the algorithm to an error on the position of the asteroids is presented in Fig. 5.17. There are here illustrated the results of a 500 s simulation in which, every 10 s, a measurement set is received by the filter on-board the mothership. The adopted pruning window size is selected to be equal to 3. The simulation highlights the difference in the estimation error when the standard deviation assumes the value of 1 m, 10 m and 100 m.

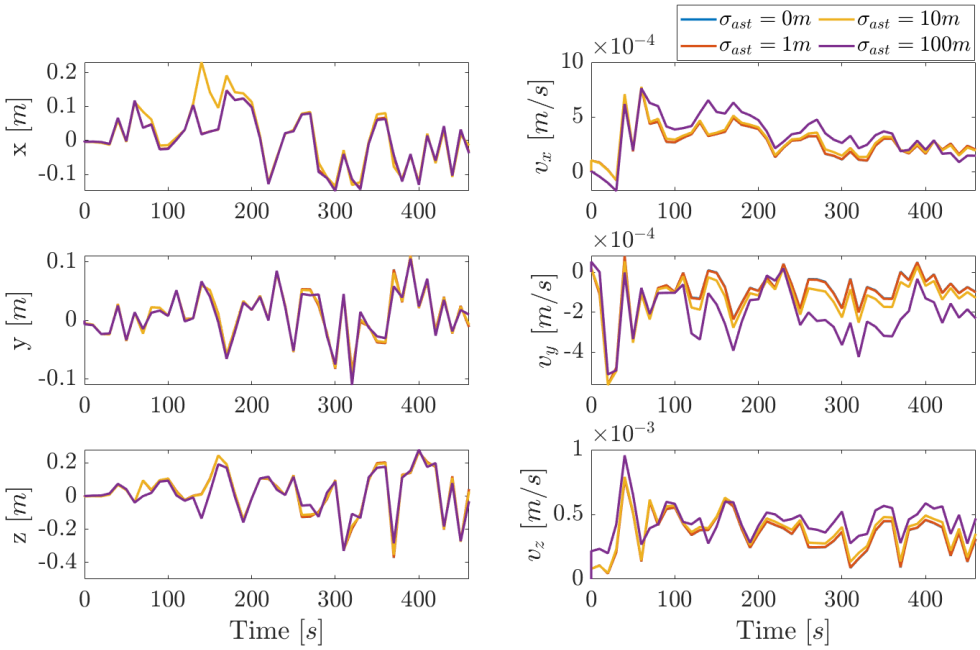


Figure 5.17. Sensitivity of the state estimation error to an error in the asteroids positions

From Fig. 5.17, it is evident that an error on the knowledge of asteroids position by the mothership, poorly affects satellites estimates. In order to appreciate the difference introduced by this inaccuracy, the error should be in the order of 100 m. For what concerns the position error, for most of the time such inaccuracy only introduces a very small additional error. However, some picks with an additional error of around 0.1 m are introduced.
Nevertheless, the state estimates does not deviate too much from the ideal case if the asteroid positions are known with an error lower than 100 m.

This result can be explained remembering the shape of the measurement set that the swarm delivers to the mothership. As the nanosatellites measure both their distance with respect to the asteroids and with respect to the mothership, these information are able to fix the incorrect information introduced by the mothership. So, this is the reason why such kind of error does not affect too much the accuracy of the results.

Chapter 6

Alternative architecture

The approach described in the previous chapters is based on the idea of exploiting the reduced size of the nanosatellites to lower mission costs. However, the drawback behind the reduced nanosatellite size is that it only allows to carry a limited amount of weight on-board and, therefore, a limited number of sensors. For this reason, in previous chapters it was assumed that range measurements only were available and that they were used to measure the distance from a nanosatellite to the mothership and to the asteroids. Developing a TOMHT algorithm using these measurements represents a solution that permits to overcome the problem of the on-board space with an affordable cost.
Nevertheless, some limitation could arise from such architecture. They will be presented in this chapter, along with an alternative measurement architecture that could allow to overcome such limitations.

6.1 Limitations of previous architecture

The main issue associated to the previously described architecture (Ch. 3.3) is related to the measurement vector and to the way measurements are performed. A nanosatellite is supposed to measure its distance both from the mothership and from the two asteroids. If a laser ranging technique is adopted in the measurement process, the nanosatellite will not measure its distance from the centre of gravity of the asteroids, it will measure the distance from the surface of the asteroids instead. In addition, asteroids are characterised by irregular shapes and irregular surfaces.
Therefore, asteroids shape should be accurately known and it should be used to implement an algorithm that, given the distance from a nanosatellite to the surface of an asteroid, estimates the distance from the nanosatellite to the centre of gravity of the asteroid.
In the algorithm presented in previous chapters, such correction has not been included and the measurements performed on-board the nanosatellite were assumed to provide the distance to the centre of gravity of the asteroid.

Furthermore, the necessity of a Laser Range Finder (LRF) on-board the nanosatellites, imposes a limit on the minimum available on-board power and, consequently, on the minimum size of the platform. The power consumption of a LRF is in the order of few Watts. When dealing with nanosatellites bigger than 1U, this power level could easily

be guaranteed. However, if dealing with picosatellites (satellites with mass between 0.1 kg and 1 kg) or smaller satellites, the available on-board power that they can provide is lower than 1 W and a LRF does not represent a viable solution anymore.

For this reason, if dealing with a swarm of very small nanosatellites orbiting around a binary system, a different measurement strategy needs to be planned in order to track the orbit of the swarm.
An alternative way of following the swarm satellites could be based on the idea that the nanosatellites are able to measure their inter-satellite distance in addition to their distance from the mothership. In this scenario, the nanosatellites do not possess any information about their distance with respect to the asteroids.
This approach allows to use the inter-satellite link to measure nanosatellites distance and the LRF sensors are not needed. Therefore, the available on-board power does not represent a bottleneck anymore, and dimensions of the nanosatellites can be further reduced with respect to the previous architecture.

6.2 Analytical considerations

In this section it will be analysed the possibility for the inter-satellite measurement architecture to provide satellite tracking capabilities under an analytical point of view. There exist two main reasons why this approach could be of interest: the first one is based on the difficulties that could arise when measuring the nanosatellite-to-asteroids distances already highlighted before; the second aspect of interest is that, if there was an internal structure between the swarm satellites that had to be preserved, this method would provide a direct measurement of the inter-satellite distance.
In order to study the feasibility of this different measurement architecture, some initial considerations are here presented. The idea is to look for an analytical solution to the problem of determining nanosatellites' position and then try to understand if a MHT method could be adequate for the implementation of this different tracking approach.

The situation under investigation is depicted in Fig. 6.1, considering that 3 swarm nanosatellites are present, along with the mothership and the two asteroids. In this plot, two reference frames are depicted: the first one is the mothership's one (Ref1), that is defined by its reference attitude, whilst the second one is a reference frame relative to the swarm (Ref2). In order to be able to define the latter frame, the position of at least three nanosatellites needs to be known with respect to the mothership.
In the presence of more than 3 swarm nanosatellites, their position can be expressed through the swarm reference frame. In fact, if the position of 3 nanosatellites is known from the mothership point of view, there is no need for the other nanosatellites to measure their distance with respect to the mothership as their position can be expressed through Ref2. Then, if the rotation matrix that provides the transformation from Ref1 to Ref2 is known, their position can be expressed in the absolute reference frame of the mothership.

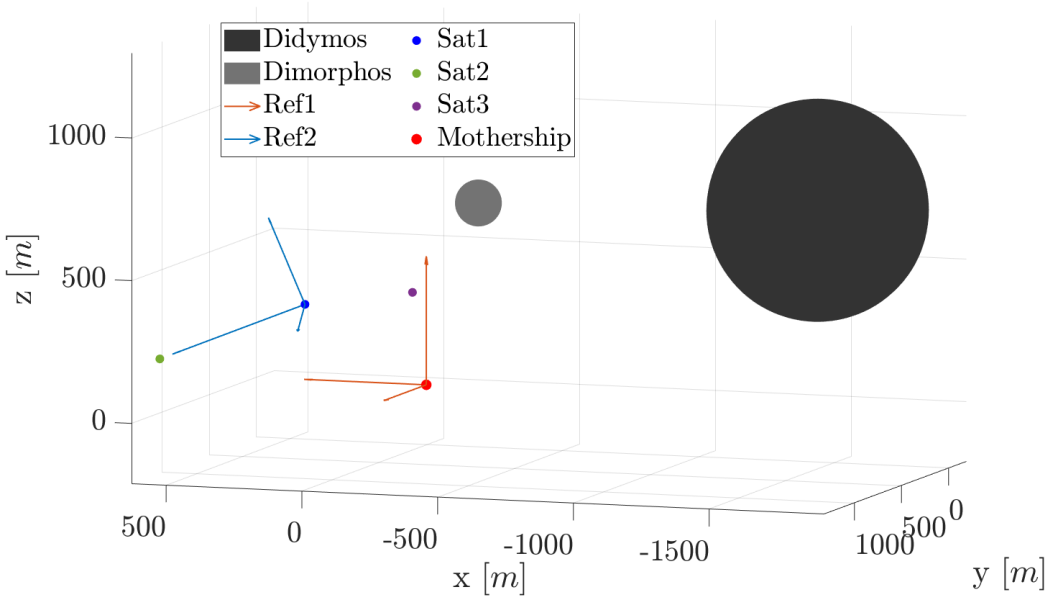


Figure 6.1. Comparison between mothership reference frame (Ref1) and swarm reference frame (Ref2)

As for the previous configuration, the TOA equations can be exploited to try to solve this problem. The measurement vector is shown by Eq. 6.1, where each of the swarm nanosatellite is associated to a number, whereas the mothership is associated to index M. The measurement vector provides the mothership with its distance from the nanosatellite and with the inter-nanosatellite distances.

$$\mathbf{z}_k = \begin{bmatrix} \sqrt{(x_1 - x_M)^2 + (y_1 - y_M)^2 + (z_1 - z_M)^2} \\ \sqrt{(x_2 - x_M)^2 + (y_2 - y_M)^2 + (z_2 - z_M)^2} \\ \sqrt{(x_3 - x_M)^2 + (y_3 - y_M)^2 + (z_3 - z_M)^2} \\ \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \\ \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2} \\ \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2} \end{bmatrix} \tag{6.1}$$

However, in case of a swarm composed by 3 nanosatellites, the information furnished by the measurement vector in Eq. 6.1 are not enough to solve the system in Eq. 6.2 (6 equations vs 9 unknowns) and to find the coordinates of the 3 nanosatellites.

$$\begin{cases} {d_1}^2 &= (x_1 - x_M)^2 + (y_1 - y_M)^2 + (z_1 - z_M)^2 \\ {d_2}^2 &= (x_2 - x_M)^2 + (y_2 - y_M)^2 + (z_2 - z_M)^2 \\ {d_3}^2 &= (x_3 - x_M)^2 + (y_3 - y_M)^2 + (z_3 - z_M)^2 \end{cases} \tag{6.2}$$

This problem could be overcome by either adding 3 more nanosatellites (18 equations vs 18 unknowns), or by exploiting the information about the pointing direction of mothership's antenna when the signals from the nanosatellites are received. This last option would add the knowledge of the rotation matrix to pass from the mothership's reference frame to the swarm's one. In general, both these choices would add enough information to solve the problem, but an analytical solution is difficult to be found in

case of 6 nanosatellites.
Nevertheless, what we can expect is that, when implementing such architecture, if the initial satellites' position guess is not too far from the real one, the filter should be able to follow satellites' motion, as it happened for the architecture in Ch. 3.3.

Once the position of the 3 nanosatellites is known, the swarm reference frame in Fig. 6.2 can be built. It results to be useful in case more than 3 satellites are part of the swarm.

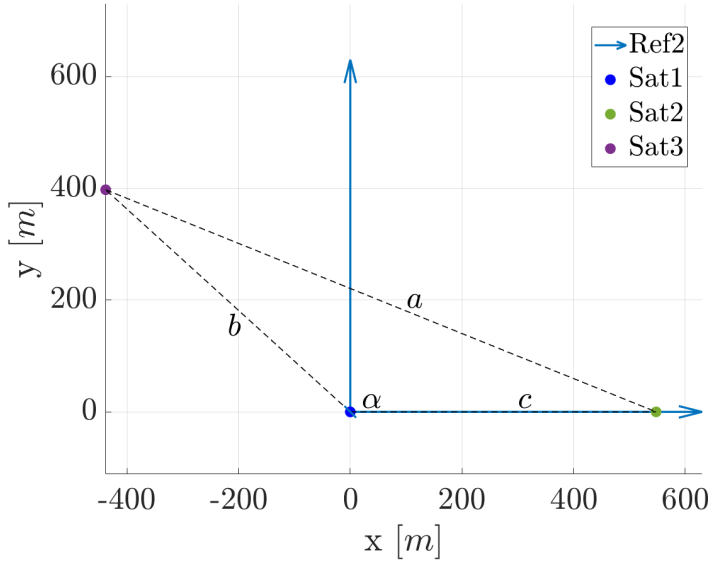


Figure 6.2. Swarm reference frame

Consider, for example, that another nanosatellite is added to the swarm and that it only measures its distance from nanosatellite 1, 2 and 3. After that such frame has been defined, nanosatellites 1, 2 and 3 have the same role that the mothership, the main and the secondary asteroid had for the previous configuration in Ch. 3.3. It means that the TOA equations, useful to find the position of a forth nanosatellite, can be solved in the same way presented in Ch. 3.3 and that they will lead to the analogous results.
Therefore, in this way, the position of the fourth nanosatellite could be either expressed in the swarm reference frame, or in the mothership one. For this reason it is not necessary that more than 3 nanosatellites measure their distance with respect to the mothership.
The swarm reference frame in Fig. 6.2 could, for example, be defined such that:

$$\begin{cases} x_1 = 0, \ y_1 = 0, \ z_1 = 0 \\ x_2 = c, \ y_2 = 0 \ z_2 = 0 \\ z_3 = 0 \end{cases} \tag{6.3}$$

where the in-plane components of satellite 3 are:

$$\alpha = acos(\frac{b^2 + c^2 - a^2}{2 \cdot b \cdot c}) \tag{6.4}$$

$$x_3 = b \cdot cos(\alpha) \tag{6.5}$$

$$y_3 = b \cdot sin(\alpha) \tag{6.6}$$

with $b = d_{13}$ and $c = d_{12}$.

6.3 Implementation considerations

For what concerns the development of a filter based on the TOMHT method and that receives the set of measurements in Eq. 6.1, it could be a not so easy task. The difficulty stands in the way a MHT algorithm works: the usual target-tracking problem does not deal with inter-target measurements. The MHT method propagates an updated state for all the plausible observation-to-track assignments and each updated state produces a predicted measurement set. If such measurement set also includes the inter-satellite measurements, it means that it should be kept into account that each predicted state for target 1 should generate a predicted distance measurement to each predicted target 2 state. For this reason, this approach would introduce a significant increase in the computational burden with respect to the case in which the measurements were taken with respect to the asteroid.
With the measurement architecture described in Ch. 3.3, each nanosatellite's predicted state only produced one measurement set, as the position of the mothership and of the asteroid are known. On the contrary, when dealing with this different measurement choice, each predicted state will produce many different measurement sets.
Assume to be in the situation represented by Fig. 6.3, in which, at a certain time instant k, each of the three nanosatellites (1, 2 and 3) has two propagated states using a MHT method.

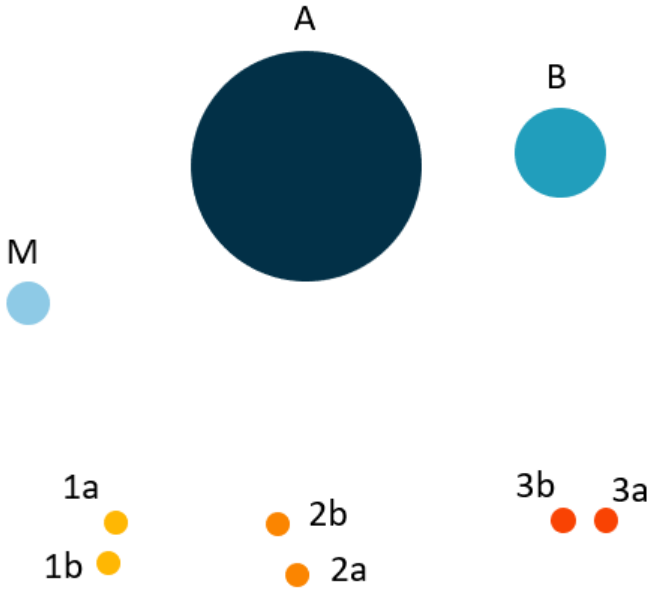


Figure 6.3. Example of nanosatellites' propagated state

Defining with A and B respectively the main and the secondary asteroid and with M the mothership, the measurement sets generated by the state $1a$ in the two different measurement architectures are shown by Tab. 6.1. Architecture 1 refers to the one in Ch. 3.3, whereas Architecture 2 refers to the inter-satellite configuration.

Architecture 1			**Architecture 2**		
1a-M	1a-A	1a-B	1a-M	1a-2a	1a-3a
			1a-M	1a-2a	1a-3b
			1a-M	1a-2b	1a-3a
			1a-M	1a-2b	1a-3b

Table 6.1. Measurement sets generated by a state in the different measurement architectures

It is clear that the computational cost of the algorithm will increase and that an additional complexity will be introduced when considering the inter-satellite measurements.

So, the algorithm shall be modified with respect to the one developed in this thesis and adapted to the different architecture. In principle, this approach should be viable and, as shown through the analytical considerations, the filter should be able to reconstruct the orbit of the nanosatellite. As the computational cost of this alternative architecture is higher with respect to the previous one, its real-time target tracking capabilities should be investigated.
Still, it is necessary to implement such algorithm and to test it in scenarios similar to those treated in this thesis.

Conclusions

At the end of this work, after the development and test of a TOMHT framework in the contest of a binary asteroids system, some observations can be discussed.
The first one is about the feasibility of this approach. This thesis was intended as a first analysis on the possibility to exploit a MHT method to determine the real-time orbits of a swarm of nanosatellites orbiting a binary asteroid system. In particular, the case in which they can only provide range measurements and in which they are closely-spaced for some intervals of their orbits has been treated.
The implemented algorithm has been tested through different scenarios that it could face during a real mission. It has been shown that a TOMHT algorithm, implemented on-board a mothership, is able to determine the orbits of the swarm nanosatellites with a reduced measurement set composed of range measurements only. The only situation that shall be avoided is the case in which the mothership is orbiting on the x-y plane in Fig. 3.1 as the filter is not able to estimate the z-component in this situation.
The knowledge of the position of the mothership has been assumed to be known, whereas it has been proven that an error on the knowledge of the position of the asteroid poorly affect the results.
The error on the state estimates varies depending on the precision of the measurements taken from the nanosatellites. However, the algorithm is able to estimate position and velocity of the nanosatellite even in case many satellites occupy the same region of space and their uncertainty overlap, without experiencing any coalescence phenomena.

The second remark is about the computational time needed to run the TOMHT algorithm. Even though it is known to be highly computationally expensive when the number of targets increase, it was proven that, with an accurate choice of the pruning window size and of the time between two subsequent observations generation, the algorithm still provides real-time target tracking capabilities.
It has also been demonstrated that the computational cost of such method is not directly related to the number of satellites itself. In fact, the time to run the TOMHT depends the number of track incompatibilities. Therefore, the computational time depends on the selected nanosatellites architecture.

Future works should deepen the tests on the configuration proposed in this thesis. In addition, the inter-satellite configuration proposed in Ch. 6 should be further explored and implemented. Firstly, because the range measurements between two satellites are less expensive to be taken with respect to the satellite-asteroid measurement. Moreover, because it could be useful, in case a relative structure inside the swarm satellites should be maintained, to measure the inter-satellite distances.

Appendix A

Additional plots

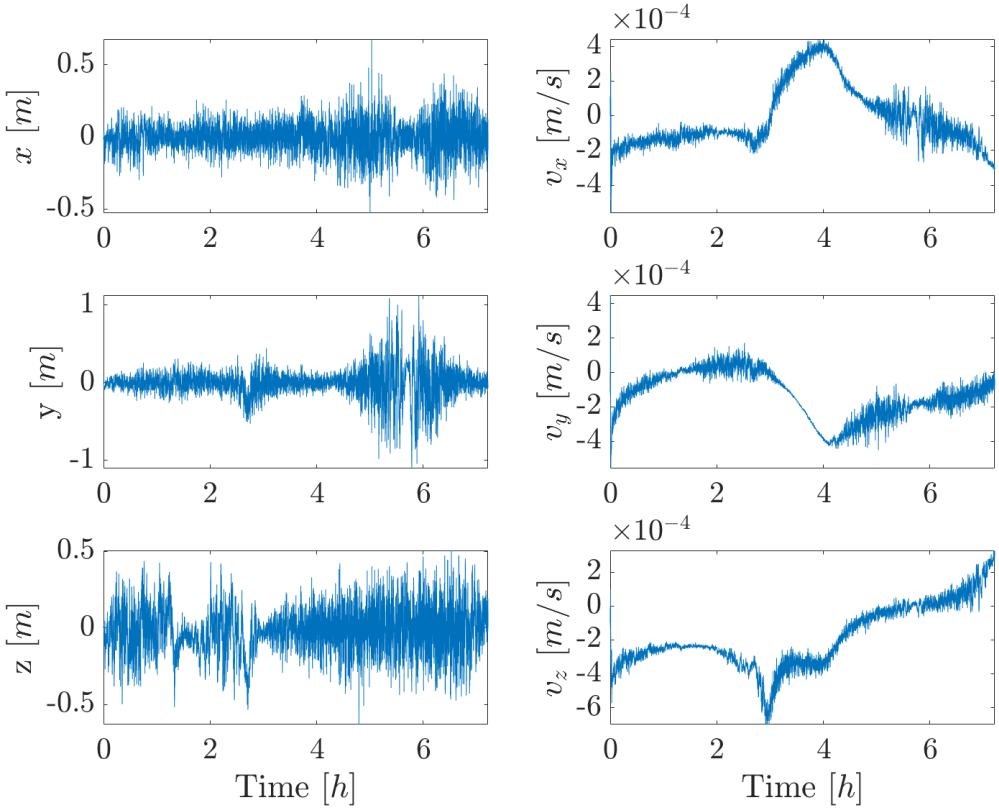


Figure A.1. Nanosatellite state estimation error for the case in Fig. 5.1

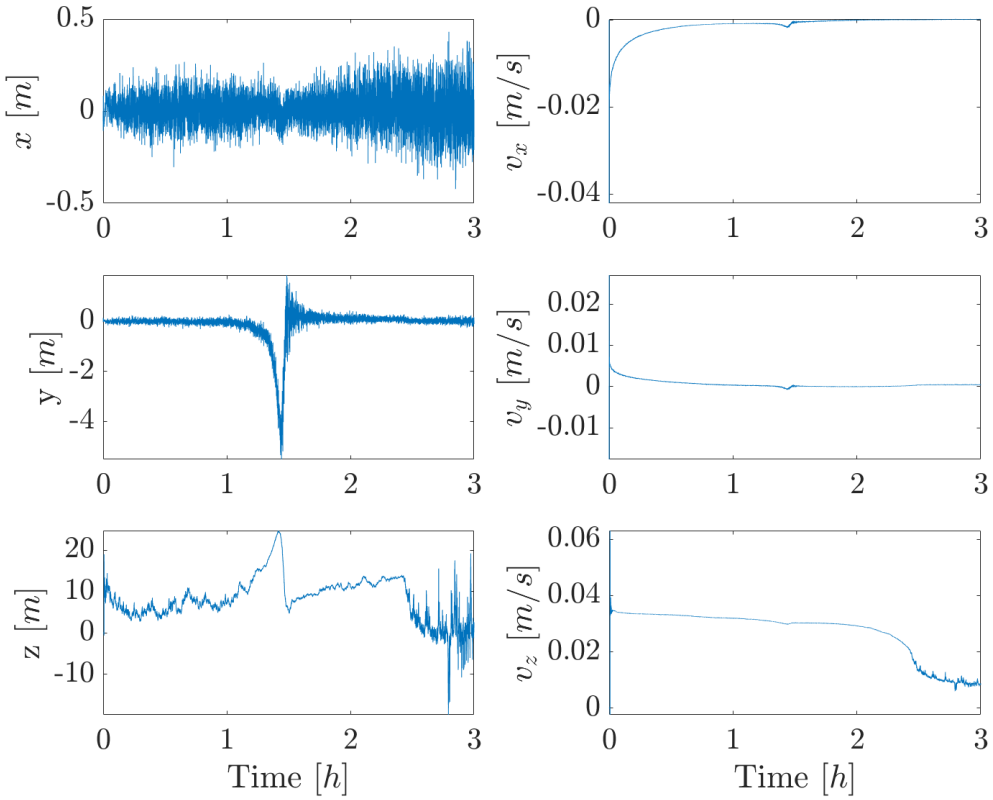
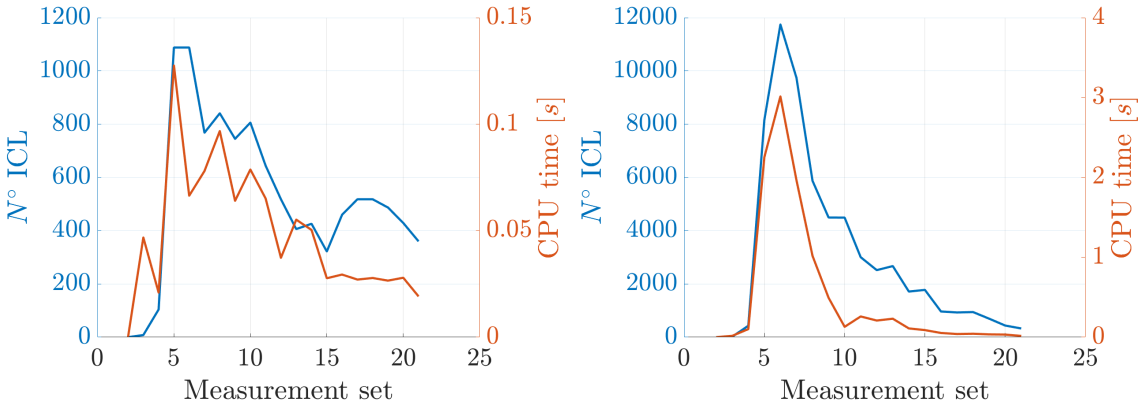
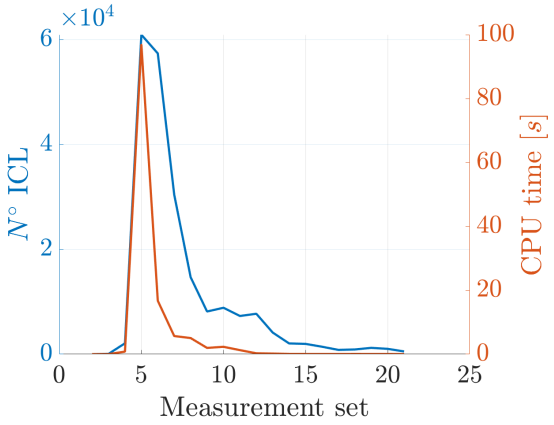


Figure A.2. Nanosatellite state estimation error for the case in Fig. 5.5

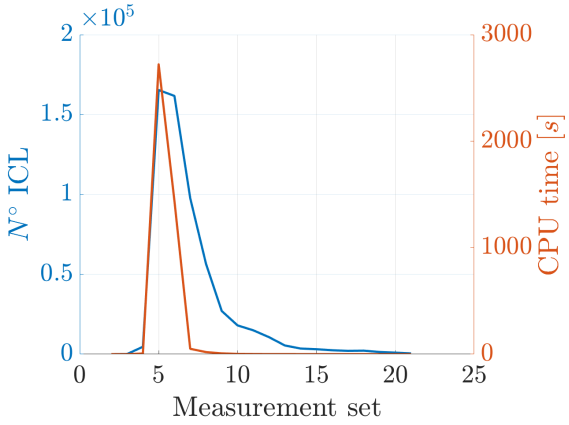


(a) 2 satellites

(b) 3 satellites



(c) 4 satellites



(d) 5 satellites

Figure A.3. CPU time vs number of incompatibilities at each time step if $N = 3$

Acronyms

ANDA	All-Neighbors Data Association
COSPAR	Committee on Space Research
COTS	Commercial Off-The-Shelf
CR3BP	Circular Restricted Three Body Problem
EKF	Extended Kalman Filter
FOV	Field of View
GNN	Global Nearest Neighbour
GNNF	Global Nearest Neighbour Filter
HOMHT	Hypothesis Oriented Multiple Hypothesis Tracking
ILP	Integer Linear Programming
KF	Kalman Filter
LEO	Low Earth Orbit
LLR	log-likelihood ratio
LRF	Laser Range Finder
JPDA	Joint Probabilistic Data Association
MAP	Multidimensional Assignment Problem
MHT	Multiple Hypothesis Tracking
MTT	Multiple Target Tracking
MWIS	Maximum Weighted Independent Set
MWISP	Maximum Weighted Independent Set Problem
NEOs	Near Earth Objects
NLLR	negative log-likelihood ratio
NNF	Nearest Neighbour Filter
NNSF	Nearest Neighbour Standard Filter
PDA	Probabilistic Data Association
PDAF	Probabilistic Data Association Filter
SRP	Solar Radiation Pressure
TOA	Time Of Arrival
TOMHT	Track Oriented Multiple Hypothesis Tracking

References

[1] NASA. "Basic Concepts and Processes for First-Time CubeSat Developers". In: *NASA* (2017).

[2] Alén Space. *A Basic Guide to Nanosatellites*. https://alen.space/basic-guide-nanosatellites/.

[3] Erik Kulu. *ompanies founded*. URL: https://www.nanosats.eu/.

[4] Elon Musk. *Starlink*. https://www.starlink.com/.

[5] S. J. Chung S.Bandyopadhyay. "Review of Formation Flying and Constellation Missions Using Nanosatellites". In: *Journal of Spacecraft and Rockets* (2016).

[6] COSPAR. "COSPAR Policy on Planetary Protection". In: *Space Research Today* (2020).

[7] Elizabeth Landau. *Cosmic Detective Work: Why We Care About Space Rocks*. https://www.nasa.gov/feature/jpl/cosmic-detective-work-why-we-care-about-space-rocks.

[8] J. R. Brophy D. D. Mazanek R. G. Merrill and R. P. Mueller. "Asteroid Redirect Mission Concept: A Bold Approach for Utilizing Space Resources". In: *Acta Astronautica* 117 (2015), pp. 163–171. DOI: 10.1016/j.actaastro.2015.06.018.

[9] NASA. *NEAR Shoemaker*. URL: https://solarsystem.nasa.gov/missions/near-shoemaker/in-depth/.

[10] ESA. *Rosetta*. URL: http://www.esa.int/Enabling_Support/Operations/Rosetta.

[11] Jet Propulsion Laboratory NASA. *Stardust*. URL: https://www.jpl.nasa.gov/missions/stardust.

[12] NASA. *Dawn*. URL: https://solarsystem.nasa.gov/missions/dawn/overview/.

[13] Kevin J. Walsh and Seth A. Jacobson. "Formation and Evolution of Binary Asteroids". In: *Asteroids IV(P. Michel et al., eds.), Univ. of Arizona, Tucson,* 10 (2015), pp. 375–393. DOI: 10.2458/azu_uapress_9780816532131-ch020.

[14] R. Serra F. Torre M. Vasile and S. Grey. "Autonomous Navigation of a Formation of Spacecraft in the Proximity of a Binary Asteroid". In: *ISTS* (2017).

[15] S. N. and D. S. "Autonomous Swarming for Simultaneous Navigation and Asteroid Characterization". In: *AAS/AIAA Astrodynamics Specialist Conference* (2015).

[16] M. Vetrisando and M. Vasile. “Autonomous Navigation of a Formation of Spacecraft in the Proximity of a Binary Asteroid”. In: *ISTS* (2017).

[17] J. Parker S. Hesar J. M. and G. Born. “Small Body Gravity Field Estimation Using LIAISON Suplemented Optical Navigation”. In: *AAS GNC conference* (2015).

[18] M. Vasile F. Torre and S. Grey. “Angles-Only Navigation in the Proximity of a Binary Asteroid System”. In: *Journal of Guidance, Control and Dynamics* (2020).

[19] Bar-Shalom. “Tracking methods in a multitarget environment”. In: *IEE Transactions on Automatic Control* AC-23 (1978), pp. 618–626.

[20] Rasmussen C. and Hager G.D. “Probabilistic data association methods for tracking complex visual objects”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (2001), pp. 560–576.

[21] Leonard J.J. and Durrant-Whyte H.F. “Application of multi-target tracking to sonar based mobile robot navigation”. In: *Proc. 29th Conf. Dec. Control., Hawaii* 23 (1990), pp. 3118–3123.

[22] Acton S.T. Goobic A.P. Welser M.E. and Ley K. “Biomedical application of target tracking in clutter”. In: *Dept. EE, Virginia Univ. Charlottesville* (2001).

[23] Bar-Shalom and T. E. Fortmann. “Algorithms for Tracking A Single Target In Clutter”. In: *Tracking and Data Association* (1998), pp. 119–185.

[24] S. Blackman and R. Popoli. “Desgin and Analysis of Modern Tracking Systems”. In: Artech House Publishers, 1999. Chap. 6.6.

[25] Y. Bar-Shalom and E. Tse. “Tracking in a Cluttered Environment with Probabilistic Data Association”. In: *Automatica* II (1975), pp. 451–460.

[26] Fortnamm T. E. et al. “Sonar tracking of Multiple Targets Usinga Joint Probabilistic Data Association”. In: *IEEE J of Oceanic Engineering* OE-8 (1983), pp. 173–184.

[27] R. G. Sea R. A. Singer and K. B. Housewright. “Derivation and evaluation of improved tracking filters for use in dense multi-target environment”. In: *IEE Trans. Inform: Theory* IT-20 (1974), pp. 423–432.

[28] Donald B. Reid. “An Algorithm for Tracking Multiple Targets”. In: *IEEE Transactions on Automatic Control* AC-24 (1979).

[29] T. Kurien. “Issues in the Design of Practical Multitarget Tracking Algorithms”. In: *Multitarget-Multisensor Tracking: Application and Advances* ch. 3 (1990).

[30] R. J. Dempster S. S. Blackman and R.W. Reed. “Demonstration of Multiple Hypothesis Tracking (MHT) Practical Real-Time Implementation Feasibility”. In: *Proceedings of SPIE* 4473 (2001), pp. 470–475.

[31] S. S. Blackman Y. Bar-Shalom and R. J. Fitzgerald. “Dimensionless Score Function for Multiple Hypothesis Tracking”. In: *IEEE Transactions on Aerospace and Electronic Systems* 43 (2007), pp. 392–400.

[32] D. J. Papageorgiou and M. R. Salpukas. "The maximum weight independent set problem for data association in multiple hypothesis tracking". In: *Optimization and Cooperative Control Strategies* (2009).

[33] *Mixed-integer linear programming (MILP)*. URL: `https://it.mathworks.com/help/optim/ug/intlinprog.html`.

[34] ESA. *Target asteroid*. URL: `https://www.esa.int/Safety_Security/Hera/Target_asteroid2`.

[35] R. Serra M. Vasile F. Torre and Stuart Grey. "Autonomous Orbit Determination and Navigation for Formations of Cubesats Beyond LEO". In: *Acta Astronautica* (2017).

[36] K. Arras G.Grisetti C. Stachniss and W. Burgard. "Robotics 2 - Data Association". In: *University of Freiburg* ().

[37] Selva Prabhakaran. *Mahalanobis Distance – Understanding the math with examples (python)*. URL: `https://www.machinelearningplus.com/statistics/mahalanobis-distance/`.

[38] A. Ciptadi C. Kim F. Li and J. Rehg. "Multiple Hypothesis Tracking Revisited". In: *Georgia Institute of Technology, Oregon State University* (2015).

[39] MATLAB. *version R2019b*. Natick, Massachusetts: The MathWorks Inc., 2019.