Mapping And Navigation Of Small Bodies In The Presence Of Uncertainty

by

Benjamin Bercovici

MSc Aerospace Engineering, ISAE-SUPAERO 2015

MSc Aerospace Engineering, UIUC 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Aerospace Engineering Sciences

2019

This thesis entitled:
Mapping And Navigation Of Small Bodies In The Presence Of Uncertainty
written by Benjamin Bercovici
has been approved for the Department of Aerospace Engineering Sciences

Prof. Jay W. McMahon

Prof. John Evans

Prof. Christoffer Heckman

Prof. Hanspeter Schaub

Prof. Daniel Scheeres

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Bercovici, Benjamin (Ph.D., Aerospace Engineering)

Mapping And Navigation Of Small Bodies In The Presence Of Uncertainty

Thesis directed by Prof. Jay W. McMahon

Missions to small bodies in the Solar System face a number of challenges as early as their inception begins, due to the lack of information that usually characterizes asteroids and comets that have never been the target of an in-situ mission or been observed from Earth in a favorable geometry. Robust mission design to these targets can only be achieved if uncertainties affecting the a-priori knowledge - or lack thereof - in the small body shapes and dynamical environments are correctly handled. Small body shape models, customarily represented as a collection of triangular facets or generalized through higher-order elements are a function of a mesh of control points effectively defining the shape. Describing this ensemble of control points as a multidimensional random variable, obeying a Gaussian distribution of known mean and covariance, enables performing linearized uncertainty quantification in the small body's inertia parameters and gravitational field, allowing valuable insight into the small body dynamical environment to be gained, at a lesser computational cost than a traditional Monte-Carlo sampling of the shape, to the benefit of mission designers and planetary scientists alike. Moving closer to the shape, the capability to autonomously survey a small body by means of Lidar observations given little to no a-priori information is demonstrated, in addition to the capacity to deliver a consistent shape estimate accounting for underlying errors in the reconstructed shape. This consistent pair of a shape estimate augmented with its uncertainty metric allows model-based navigation to take place in a robust manner, through the use of an Iterated Extended Kalman Filter taking in position and attitude measurements from a Consider Batch Filter augmenting the measurement covariance with a commensurate consider contribution coming from the shape uncertainty model. A sensitivity analysis covering a subset of the parameter space has validated the proposed framework's robustness, paving the way for autonomous mapping and navigation of small bodies in the presence of uncertainty.

Dedication

To my family.

Wir müssen wissen

Wir werden wissen

Acknowledgements

A few lines on a page will hardly do justice to any attempt to express my gratefulness for having had Jay McMahon as my research advisor. I have been measuring the incredible luck of benefiting from your guidance and advising throughout my time at CU, on top of your amazing character as a person and lab leader. So in the low-keyest of all attempts, thanks ! On a similar note, I am indebted for life to my parents, Daniel and Nathalie, and my sisters, Clara and Deborah, who have always fostered my interests (and never quite lost their minds over my numerous garage experiments or my music tastes) and supported me over the last 27 years, no matter the times or the distance between us. My time here would also not have been the same without Alex and Stef, for the many great moments of friendship that will remain chiseled in my memories (including the most absurd yet intense Savage Worlds campaign ever played), Thibaud and Mar, whose characters and personalities made me feel like home, to Manuel, without whom I would have never heard of Nine Feet Underground, and to Samm and Jeroen, for the positiveness that they will always embody. Also, shout-out to Ann, without whom this very thesis would not exist, to Nicola, for being no less than a role model of talent and dedication, and to Paolo, for the fruitful research atmosphere that prevailed at my apartment for the last two months of my PhD. Many more people inside and outside CCAR could have had their names added to this list, and I'm no different than anyone else when faced with the cruelty of summarizing such a chunk of life over one single page. My last words of thanks will be for all my teachers and professors, without whom I would have gone nowhere, to Didier, Cyrille and Nathalie for their invaluable guidance during my short time at Airbus-DS, to Gilles, for his casual yet decisive suggestion to check out 'CU Boulder' back when I was just starting to look into PhD programs, and to NASA, for funding this PhD through the grant PDART NNX16AG50G.

Contents

Tables

Table

Figures

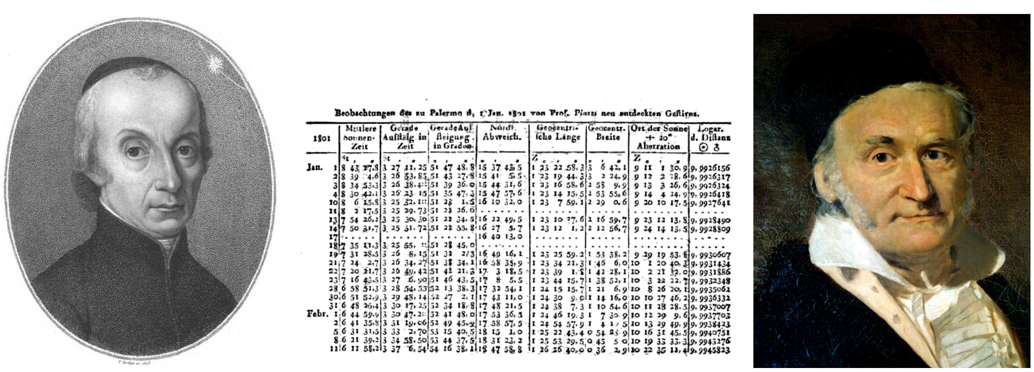
Figure

Chapter 1

Background and Motivation

1.1 Small bodies in the Solar system

The interest for small bodies in the solar system has been a recurring theme in space science. The first discovery of an asteroid (1 Ceres) was made by Giuseppe Piazzi in 1801 after he realized that the evanescent speck of light he noticed in his telescope was moving against the starry background. Unfortunately for Piazzi, the unfavorable observation geometry prevented him from holding Ceres in sight as it vanished shortly after. Carl-Friedrich Gauss heard of his demise and devised a technique akin to least-squares to fit Piazzi's observations, and successfully predicted the reappearance of Ceres nearly a year after its initial discovery.



Beobachtungen des zu Palermo d. 1. Jan. 1801 von Prof. *Piazzi* neu entdeckten Gestirns.

1801	Mittlere Sonnen-Zeit	Gerade Aufsteig. in Zeit	Gerade Aufsteigung in Graden	Nördl. Abweich.	Geocentrische Länge	Geocentr. Breite	Ort der Sonne + 20″ Aberration	Logar. d. Distanz ☉ ♂
	St ′ ″	St ′ ″	° ′ ″	° ′ ″	Z ° ′ ″	° ′ ″	Z ° ′ ″	
Jan. 1	8 43 27,8	3 27 11,25	51 47 48,8	15 37 43,5	1 23 22 58,3	3 6 42,1	9 11 1 30,9	9,9926156
2	8 39 4,6	3 26 53,85	51 43 27,8	15 41 5,5	1 23 19 44,3	3 2 24,9	9 12 2 28,6	9,9926317
3	8 34 53,3	3 26 38,4::	51 39 36,0	15 44 31,6	1 23 16 58,6	2 58 9,9	9 13 3 26,6	9,9926324
4	8 30 42,1	3 26 23,15	51 35 47,3	15 47 57,6	1 23 14 15,5	2 53 55,6	9 14 4 24,9	9,9926418
10	8 6 15,8	3 25 32,1::	51 23 1,5	16 10 32,0	1 23 7 59,1	2 29 0,6	9 20 10 17,5	9,9927641
11	8 2 17,5	3 25 29,73	51 22 26,0	. . . . . .	. . . . . .	. . . . .	. . . . . .	. . . . .
13	7 54 26,2	3 25 30,30	51 22 34,5	16 22 49,5	1 23 10 27,6	2 16 59,7	9 23 12 13,8	9,9928490
14	7 50 31,7	3 25 31,72	51 22 55,8	16 27 5,7	1 23 12 1,2	2 12 56,7	9 24 14 13,5	9,9928809
17	. . . . . .	. . . . . .	. . . . . .	16 40 13,0	. . . . . .	. . . . .	. . . . . .	. . . . .
18	7 35 11,3	3 25 55,::	51 28 45,0	. . . . . .	. . . . . .	. . . . .	. . . . . .	. . . . .
19	7 31 28,5	3 26 8,15	51 32 2,3	16 49 16,1	1 23 25 59,2	1 53 38,2	9 29 19 53,8	9,9930607
21	7 24 2,7	3 26 34,27	51 38 34,1	16 58 35,9	1 23 34 21,3	1 46 6,0	10 1 20 40,3	9,9931434
22	7 20 21,7	3 26 49,42	51 42 21,3	17 3 18,5	1 23 39 1,8	1 42 28,1	10 2 21 32,0	9,9931886
23	7 16 43,5	3 27 6,90	51 46 43,5	17 8 5,5	1 23 44 15,7	1 38 52,1	10 3 22 22,7	9,9932348
28	6 58 51,3	3 28 54,55	52 13 38,3	17 32 54,1	1 24 15 15,7	1 21 6,9	10 8 26 20,1	9,9935062
30	6 51 52,9	3 29 48,14	52 27 2,1	17 43 11,0	1 24 30 9,0	1 14 16,0	10 10 27 46,2	9,9936332
31	6 48 26,4	3 30 17,25	52 34 18,8	17 48 21,5	1 24 38 7,3	1 10 54,6	10 11 28 28,5	9,9937007
Febr. 1	6 44 59,9	3 30 47,2::	52 41 48,0	17 53 36,3	1 24 46 19,3	1 7 30,9	10 12 29 9,6	9,9937703
2	6 41 35,8	3 31 19,06	52 49 45,9	17 58 57,5	1 24 54 57,9	1 4 15,5	10 13 29 49,9	9,9938423
5	6 31 31,5	3 33 2,70	53 15 40,5	18 15 1,0	1 25 22 43,4	0 54 23,9	10 16 31 45,5	9,9940751
8	6 21 39,2	3 34 58,50	53 44 37,5	18 31 23,2	1 25 53 29,5	0 45 5,0	10 19 33 33,3	9,9943276
11	6 11 58,2	3 37 6,54	54 16 38,1	18 47 58,8	1 26 26 40,0	0 36 2,9	10 22 35 11,4	9,9945823

Figure 1.1: Giuseppe Piazzi (left) first observed Ceres in early 1801. The observations he collected spanned slightly over one month. After Ceres was lost, Carl Friedrich Gauss (right) developped an initial orbit determination technique that led to the re-discovery of Ceres a year later

Since then, thousands of small bodies have been discovered by means of optical or radar observation within the Solar system, and the catalog of detected objects keeps growing. Figure 1.2 shows the staggering increase in the number of known Near-Earth Objects (NEO) since the introduction of a number of dedicated automatic sky surveying programs.

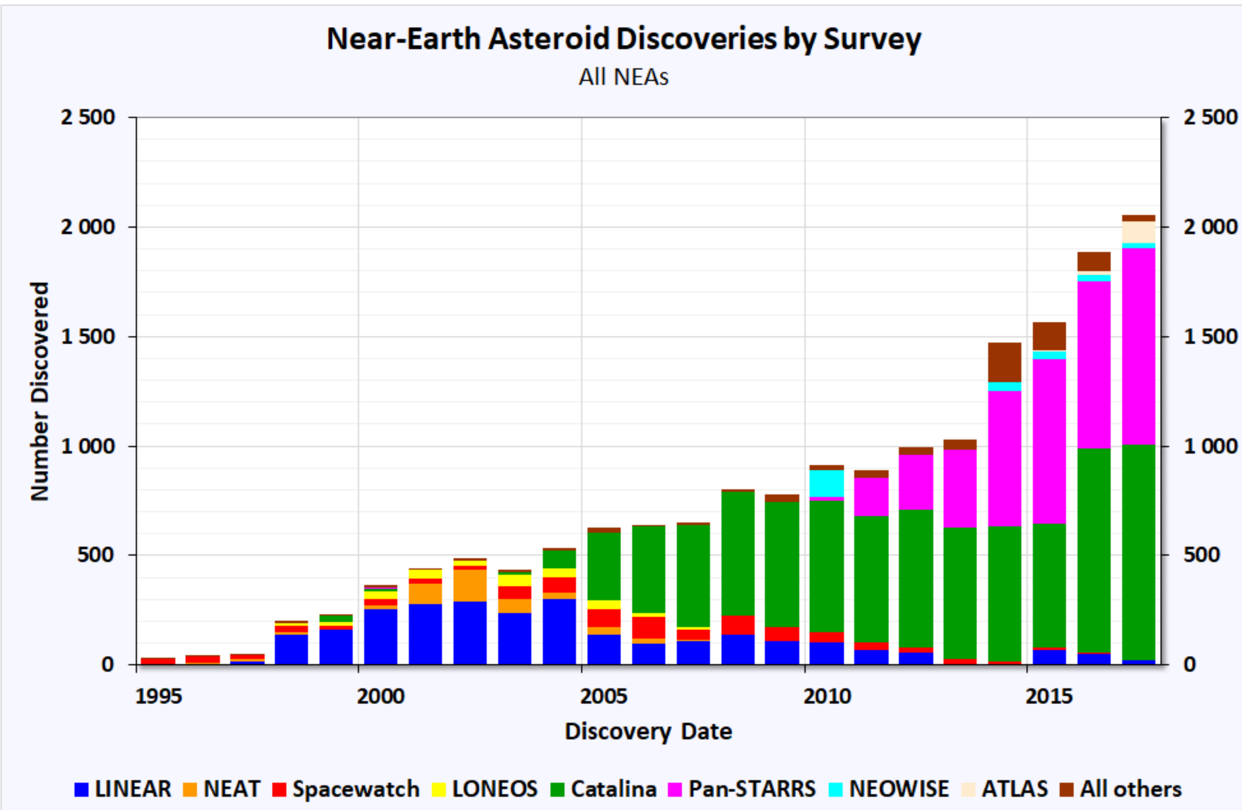


Figure 1.2: Chart of detected NEO's from 1995 to 2018 (courtesy of Alan B. Chamberlin at JPL's CNEOS)

The reasons explaining the considerable attention directed at small bodies are three-fold.

First, asteroid and comets are known to be remnants of the solar system formation [6] [7]. The asteroid main-belt, which can be found between 2.1 and 3.5 Astronomical Units from the Sun, is the most blatant artifact of early-solar system planet embryos collisions and subsequent debris scattering [8][9]. Because small bodies are less likely to be depleted of their volatile compounds as they move away from the Sun, the surface composition of main-belt asteroids could have virtually remained untouched since the creation of these bodies. Earth-based observational evidence suggests the presence of hydrated components at the surface of Ceres as well as water vapor [10] [11], but observations acquired by the Dawn spacecraft after its arrival at Ceres showed no water-ice absorption bands [12]. This apparent discrepancy (that could be explained by cryo-volcanism in the case of Ceres) could be lifted by means of more in-situ missions. Specifically, the gap between remote and in-situ small body observations can be filled by means of autonomous sample return missions, like NASA's OSIRIS-Rex [13] or JAXA's Hayabusa 2 [14], that will enable access to pristine surface or sub-surface material, and provide a unique porthole into this side of the history of the Solar System.

The second reason supporting small bodies investigation is Planetary Defense. The threat posed by small bodies to Earth dates as far back as the formation of the Solar system. The Cretacean extinction event, responsible for no less than the disappearance of the dinosaurs 66 millions years ago is strongly believed to be no else than the aftermath of an asteroid impact [15]. In more recent times, the 1908 Tunguska explosion of a 10-m large asteroid released approximately 10 to 20 megatons of explosive equivalent above the inhabited lands of Central Siberia [16]. Even more recently, the 2014 Chelyabinsk incident was a dramatic reminder that a relatively small object can have catastrophic consequences should it re-enter above a populated area [17]. Early warning and the capability of deflecting inbound objects are thus the two pillars of planetary defense [18]. But because small bodies are strongly affected by non-gravitational forces and torques, such as the Yarkovski and YORP effects [19], which are a function of the object's surface properties and mass, remote observations of asteroids do not always suffice to propagate trajectories forward in time. This uncertainty in the small body dynamics plagues long-term orbit determination, which is of utmost importance to determine whether the object actually poses a threat. Planetary Defense could therefore be benefit from in-situ missions sent out to investigate potentially hazardous objects.

Finally, in-situ resource utilization (ISRU) has recently received unprecedented attention, with the development of private ventures aiming to characterize and mine Near-Earth Asteroids of interest. Among these, M-class asteroids are estimated to be worth billions of dollars in ore (cobalt, platinum,...) [20]. Water ice is also regarded as a resource of prime interest, as it holds oxygen and hydrogen, themselves precursors to rocket fuel or life-support systems. Obviously, dedicated in-situ operations is required for these resources to become available. Unfortunately, the recent demise of Planetary Resources has left ISRU in its infancy, with a number of outstanding economical and technological challenges that still need to be addressed.

1.2 Spacecraft navigation

The enhancement of proximity spacecraft operations about small bodies necessary to enable the three major goals listed in section 1.1 relates to the ever-continuing push towards more spacecraft autonomy. The current state-of-the-art of spacecraft navigation is building upon major the breakthroughs made 60 years ago, under a fortunate set of circumstances that will remind the reader of the dramatic contribution of Gauss to Piazzi's hunt for Ceres: the seminal work of Kalman in his 1960 paper [21] provided a novel discrete-time alternative to existing frequency-based, continuous-time uncertainty propagation techniques. At the time, NASA Ames was in need for a robust and tractable scheme capable of computing circumlunar navigation solutions. The iterated linear weighted least-squares then in use at JPL [22] was too much of a numerical burden for the computer resources available at Ames (see Figure 1.3). The fortunate conjunction of the Apollo's program interest and Kalman's breakthrough which allowed the embedding of the dynamics within the sequential estimation scheme paved the way to the exponential development of dynamical state estimation. This field of research has since expanded way beyond the aerospace world. Spacecraft attitude estimation followed shortly after, once the modeling of rigid-body rotational dynamics had improved [23]. It must be noted that these early efforts were concerned with the estimation of the position and attitude state of a known spacecraft. In particular, this setup assumed that measurements akin to angles and angle-rate of change could be provided by start trackers or speed gyros directly mounted on the vehicle.

Advances made during the Apollo program would soon benefit to unmanned missions into the

Solar system. Mariner 9 was the first spacecraft to ever orbit another planet when it reached Mars in November 1971 [24]. The interplanetary navigation was conducted by means of radiometric data from the spacecraft retrieved by the Deep Space Network (DSN) stations. This mission also marked the first in-flight validation of OPtical NAVigation (OPNAV) techniques as a potential complement to radiometric measurements. This was in part motivated by the fact that a navigation solution solely obtained from radiometric measurements is affected by uncertainties in the solar system bodies ephemerids, whereas OPNAV provides a more direct measurement of the relative spacecraft-to-planet state. OPNAV proceeds by first obtaining images of dim stars and planets as seen from a typically narrow-angle. This image is then correlated with astrometric data obtained from a catalog of known stellar and planetary objects. The geometric transform obtained from the correlation encompasses the desired relative state. Although the OPNAV demonstration was found successful, it would take a few more years until it started to be used in conjunction with radiometric data: the orbit determination of Voyager 2 heavily relied on optical observations of Neptune and its satellites while the spacecraft was flying by the planetary system in 1987 [25]. It is noteworthy that man-in-the-loop data pre-processing is sometimes still required, as exemplified by the difficulty to achieve correct limb alignment when the imaged body features an atmosphere [26]. OPNAV is now the workhorse of interplanetary navigation and has been utilized in a number of missions like Cassini [26], Rosetta [27] and Galileo [28].

1.3 Small body navigation

The Near Earth Asteroid Rendezvous Shoemaker (NEAR Shoemaker) marked the first successful orbiting and landing onto an asteroid [29]. The spacecraft's mission came to an end in January 2001 when it gently touched down on the surface of asteroid 433 Eros. The technological enabler to this unique mission profile was optical landmark tracking: about 1590 surface landmarks were identified at the surface of Eros and tracked in successive images. Computer-aided crater-centerfinding over the images transmitted back to Earth allowed for the determination of Eros' rotation state along with the spacecraft's trajectory [30].

This extension of OPNAV to proximity operation has since been refined and successfully applied to all subsequent small-body bound missions in their relative navigation pipeline [31] [32]. Craters are no longer systematically tracked, as they may very well not exist at the surface of the orbited body. This evidence justified the development of the Stereo-Photo-Clinometry (SPC) approach, that relies on hybrid landmarks or L-maps formed by the combination of slope and albedo data [33]. In addition to not relying on physical features like craters, SPC is also robust to varying lighting conditions that occur naturally as the phase angle between the imaging spacecraft, the imaged small body and the Sun evolves. Finally, the L-maps extraction process is well automated, thus requiring a lesser manpower, although the Rosetta OPNAV team found that L-map generation can be failure-prone in some cases [32].

SPC forms the state-of-the-art of today's relative navigation techniques, and will be at the core of the incoming Osiris-Rex proximity operation phase [34]. Despite these successes, the applicability of OPNAV for proximity operations is bounded by a number of constraints:

- Operationally, successful OPNAV is tied to lighting conditions and sun phasing. Although SPC is robust to large lighting variations, unfavorable pole orientation may very well prevent large portions of the surface area to receive sunlight for months. Optical navigation could thus only take place over the sunlit side of the object. For instance, this issue could arise if the rotation pole of the object lies within the orbit plane.

- The reconstruction of L-maps leverages varying lighting conditions, which may or may not be naturally occurring. In addition, this procedure is subject to convergence issues as said earlier, which require significant man-hours dedicated to the monitoring of the SPC fit.

- Because SPC boils down to assembling and solving a large linear system by a batch-like procedure, the reconstruction of the L-maps implies a significant computational burden that cannot feasibly be handled on-board. For this reason, images collected by navigation cameras are typically transmitted back to Earth after possible pre-processing to diminish the overall data size.

- The relatively large size of the transmitted data (about 1.5 Megabyte per black-and-white 1024 x 1024 12-bit-per-pixel uncompressed image) implies the use of a dedicated set of large receiving stations like the DSN to retrieve the images as fast as possible [34].

In truth, these constraints can be alleviated thanks to two ingredients, which are mission time and money. Even if historical space-faring actors like NASA, JAXA, ESA and Roskosmos can afford these missions, it places an increasing stress on mission-critical resources like the DSN or its counterparts [35]. This makes a strong case in favor of more spacecraft autonomy, so as to relax the ground link requirements by letting a robotic spacecraft rely less on ground-based operations for navigation.

1.4 Sensors and methods for autonomous spacecraft operation

Spacecraft autonomy has been the object on an increased attention over 15 years. The Deep Impact mission that led to the successful intercept of comet Tempel-1 by a kinetic impactor simultaneously imaged by a flying-by spacecraft was the first realization of autonomous navigation. This incredible success relied on optical sensors on board of both spacecraft and the AutoNav software that computed the navigation solution using center-finding techniques [36]. The success of more advanced robotic missions such as satellite servicing or proximity operations about small bodies was unsurprisingly found to be heavily dependent on the capacity of spacecraft to operate autonomously as they carry out their mission [37]. Achieving science or engineering goals without reducing the mission envelope thus requires spacecraft to perform data processing and decision making without external input. This is a textbook example of where advanced state and parameter estimation techniques are needed. For instance, orbital debris mitigation can only be addressed by means of autonomous robotic servicing spacecraft if one were able to remotely determine the state, inertia, or any other relevant parameter of a non-cooperative target for which little if no apriori information is available [38]. Pioneering rendez-vous, remote inspection and stand-off of a servicer and a non-cooperative target was demonstrated in 2012 in the frame of the PRISMA experiment [1], using angle-only measurements provided by an optical camera and ground-in-the-loop processing. The final relative separation between the two spacecraft was close to 3 kilometers in average, which is too far to allow resolved observations of the target.

Relative pose information can also be provided by Lidar sensors, as a replacement or complement to optical cameras. This active sensor type can be broken down into single-beam range finders, scanning Lidars and flash Lidars. First flown on Apollo 15, single-beam range finders have since been operated about Mars [39], asteroids (433) Eros [30], (25143) Itokawa [40] and (162137) Ryugu [41]. Scanning Lidars feature a rotating mirror allowing for an effective scanning of the targeted object, with Osiris-Rex's OLA being the first scanning Lidar to be flown on a extraterrestrial body-bound mission [42]. They have also been used within the GNC suite of the ATV/HTV GNC instrument suite [43]. Finally, flash Lidars feature a laser source associated to a focal plane of photo-receptors, effectively producing a point cloud from the collection of the laser light reflected by the targeted object. The successful STORMM flight experiment that took place in 2011 marked the first demonstration of Flash Lidar as a relative navigation sensor [2]. Flash Lidar differ from optical camera in many ways: from a navigation standpoint, they provide bearing angles as well a direct measure of the range to the targeted object, whereas monocular optical sensors only provide angles if the relative dynamics cannot be leveraged to infer the range. Flash Lidar technology was thus chosen as the proximity navigation sensor-of-choice for the future Orion vehicle [44]. The maturity of Flash Lidar technology will also improve from the experience gained with NASA'S OSIRIS-Rex, as the probe carries one of Advanced Scientific Concepts' GoldenEye Flash Lidar technology demonstrator [45], in addition to the OLA scanning Lidar [42] also carried by the spacecraft.

This decision is supported by ground-based hardware-in-the-loops simulations that have demonstrated cooperative relative navigation using Lidar as the only navigation sensor [46]. This referenced work featured a known CAD model of the target satellite being orbited about. This way, point clouds collected by the Lidar instrument could be registered to the known shape model so as to infer the relative state between a chaser and the targeted spacecraft. This approach will typically be used in the incoming Restore-L mission, where a servicer spacecraft will rendezvous and dock with the Earth-observing Landsat 6 spacecraft, of known condition and shape model, so as to refuel and relocate it to a different orbit [47]. Going back to the Orion vehicle, the assumption that a CAD model of the target is available still makes perfect sense, since the ISS or another space vehicle visited by Orion would certainly be known in advance. Obviously, the fact that the shape model of the visited object is not always known poses a significant challenge. An interesting contribution relying on a probabilistic Bayesian framework was provided by Lichter and al., in which they performed shape, inertia and attitude parameter estimation of an unknown spacecraft. The shape was parametrized implicitly using voxels [48]. However, it must be noted that the observation model retained in this study was fairly optimistic, as it was assuming that a fully-registered point cloud was readily available from a formation of spacecraft carrying Lidar instruments.

Dealing with an unknown environment as a navigation proxy while estimating a dynamical state is the foundation of SLAM techniques. Simultaneous Localization and Mapping (SLAM) pertains to the problem of charting the map of an unknown environment while finding the location of the mapping sensor at the same time. This problem was initially formulated in 1991 with the goal of providing an alternative to stochastic maps that would store exhaustive spatial relationships between measurements and state estimates [49]. This notion of stochastic maps has evolved into factors graphs, that keep track of states and landmarks positions in a probabilistic graph, where the edges denote the joint density distribution of connected nodes [50]. As opposed to a filtering scheme like the different flavors of the Kalman filter, the graph variables representing the previous state estimates and observations are not marginalized at every timestep. This bookkeeping effort allows for later bundle adjustment and loop closure, enabling future observations to correct past state estimates.

The work of Tweddle, Saenz-Otero, Leonard and Miller is one of the most advanced research efforts on the topic of tumbling rigid-body state estimation using factor graphs [3]. The position, orientation, linear velocity, angular velocity, center of mass, principal axes and ratios of inertia of a tumbling target were estimated. This hardware-in-the-loop simulation took place on board the International Space Station in 2015, using the Synchronize Position Hold Engage Reorient Experimental Satellites (SPHERES) platform using body-mounted stereo cameras. Three SPHERES platform are shown on Figure 1.7. A collection of Speeded Up Robust Features (SURF) features was constructed and tracked so as to measure the position and attitude of the SPHERES platform. A dense surface map provided by all the computed SURF features was obtained towards the end of the estimation pass, yielding a good approximation of a SPHERES' shape. The listed parameters were all successfully estimated, but it was noted that this whole procedure was not suitable for real-time implementation due to the growth in the graph size. This degeneracy in the graph's size is a well-identified problem in the SLAM community. Recent developments have thus been focusing on diminishing the burden caused by the growth in the factor graphs. Marginalization and conditioning of the graph nodes are two possible approaches to tackle this issue. They both strive to make graph inference possible by removing nodes from the graph. Both techniques will result in an information loss that is compensated by the retained tractability of the inference [51] [52].

In any case, it must be noted that enforcing sparsity in the SURF features collection introduces holes and gaps in the shape model reconstruction, since this collection of features is not a closed-form shape model but merely a point-wise tiling of the surface.

At this point, the following key observations can be made:

- **A**: Flight-proven OPNAV techniques have a somewhat limited applicability due to lighting constraints, downlink, numerical stability and computational costs. In addition, monocular optical sensors provide no along-line-of-sight information unless relative dynamics are incorporated in the estimation workflow.

- **B**: SLAM-based techniques feature good estimation performance when combined with optical sensors, but are subject to limitations inherent to the tractability of the problem at hand. In addition, optical sensor-based SLAM is subject to the same operational constraints as traditional OPNAV due to their sensitivity to lighting conditions.

C: A dense collection of landmarks can be representative of the body of interest's shape, but is not a closed-form representation of the said shape, since there is no geometrical tie between the different features.

A first attempt to overcome the difficulties inherent to **A** was addressed by an alternative approach of relative navigation about a small body by Dietrich and McMahon [53]. They designed a novel, model-based relative navigation framework relying on Lidar data and the knowledge of the physical parameters of a small body of interest. Namely, the shape, center of mass and attitude state of the orbited small body were the only prerequisites for an OD solution to be produced, independently from lighting conditions or ground communication.

Solely relying on SLAM navigation presents a number of challenges, as reminded in **B**. Cotlin et al. motivate their choice to prefer a filtering-based approach over smoothing-based SLAM in the SPHERES follow-up GNC by the following: *While approaches for simultaneous localization and mapping (SLAM) such as LSD-SLAM are increasingly capable, they are not as reliable as techniques which rely on a fixed, pre-computed map. Astrobee is confined to a fixed area so the flexibility that SLAM provides is unnecessary* [54]. This observation can be translated to the small-body attitude and shape estimation problem: the small-body of interest has a finite extent in space, hence making the environment map charted by the surveying spacecraft also finite. This contrasts the typical SLAM setup where a robot travels in an environment possibly subjected to no scale or topology constraint.

Finally, the literature provides no sequential, closed-form shape estimation procedure addressing the shortcomings of SURF-based shape reconstruction stressed in **C** and tailored to small-body missions. Shape reconstruction from point clouds is a well-established research area as many powerful approaches like the Poisson Surface Reconstruction exist [4]. PSR's principle of operation is summarized on Figure 4.4. Recently, the KinectFusion algorithm leveraging Microsoft's Kinect hybrid camera system has demonstrated on-line scene reconstruction and object tracking [55], relying on a Signed Distance Field evaluated over a 3D voxel grid to capture the imaged scene. Besides the GPU processing requirements for the KinectFusion algorithm to function on-line, it must be noted that the geometry-extraction capabilities of KinectFusion are made possible by the use of an implicit shape parametrization, the signed-distance field. Relying on a general, explicit shape model parametrization achievable on-board is preferable, due to the availability of analytical expressions providing direct insight into the inertia tensor, center of mass and gravity field directly arising from an explicit, topologically closed shape model.

1.5 Small body dynamical environment and shape uncertainty

The knowledge of the shape of a small body provides considerable insight into the inertia properties of the targeted object, from its volume to a description of the neighboring gravity field. This is why reconstruction of small body shapes by means of remote observations is required for the characterization of asteroids and comets before in-situ observations can take place [56].

However, it can be noted in the literature that the uncertainty in the shapes reconstructed by such is not often described in a systematic way. In particular, an analytical connection between the suspected error in the shape and that of its inertia characteristics (volume, center of mass, inertia tensor) is seldom made. For instance, in their assessment of the binary system KW4, Scheeres et al. provide uncertainties in the primary's volume and other inertia properties but leave aside the details in their computation [57]. Similarly, the shape model of Bennu which was reconstructed by means of radar observations was provided along with uncertainties in its inertia parameters deemed as 'subjective' by its authors [58]. These uncertainties are fundamental to subsequent science and engineering applications as Scheeres et al. acknowledge in their 2016 discussion of the dynamical environment of Bennu. They recognize that variations as slight as 10 % in the polar dimensions of the object 'could substantially change important elements of the asteroids geometric and geophysical properties' [59]. Busch et al. make the same assessment in their modeling of Asteroid 10115 [60], underlining the effect of shaded areas on the determination of the asteroid's shape model and the subsequent effect of the shape uncertainties on the asteroid's dynamical environment. Further, the characterization of many small body shape models obtained by means of lightcurves, like these computed by Torppa et al., is incomplete, as it leaves the determination of the uncertainty in their rotational parameters and shape models to a 'rule of thumb' [61]. Similarly, the reconstruction of the shape of 433 Eros by Miller et al. provided inertia characteristics of the shape but without quantification of their uncertainties [62]. Muinonen has proposed the so called Gaussian shape hypothesis, which consisted in decomposing a shape of interest as a spherical expansion of random Gaussian variables [63] so as to extract its inertia moment statistics. Yet, this approach cannot handle arbitrary body shapes, as well as being well not directly related to polyhedron shape models, the workhorse of today's shape models parametrization in the small body astronomy field [64]. There is thus a gap in the literature when it comes to thoroughly characterize shape uncertainty and its relationship with the shape's inertial parameters.

The same remark extends to the lack of formal expressions for the uncertainties in the polyhedral or spherical harmonics gravity fields emanating from an uncertain shape. If mass concentrations have been used in the past to describe the uncertainty in the gravity field in proximity to a small body [65], as well as dedicated Monte-Carlo analyses to investigate the effect of uncertainty spherical harmonics coefficients on spacecraft trajectories [66], the literature shows no record of an analytical approach tailored to quantifying the formal uncertainty in these gravity fields. The following observation summarizing the identified gap can thus be made:

D : There exists no analytical means to capture the uncertainty in the inertia properties and gravity fields associated with a small body whose shape is only known in a probabilistic sense.

Addressing **D** would provide scientists and missions designers with valuable insight into the structure of the dynamical environment of an unknown small body, enabling more robust mission design in the presence of uncertainty as well as a deeper understanding of the natural dynamics taking place around or at the surface of the considered object.

1.6 Thesis statement

The work carried out in this thesis can be summarized under the following statement

The rigorous treatment of small body shape and state uncertainties in the context of remote and proximity operations is beneficial to the understanding of the dynamical environment and the robustness of science and engineering proximity operations

1.7 Thesis overview

1.7.1 Contributions

This dissertation

(1) Demonstrates the superior performance of the Iterative Closest Point algorithm parametrized in terms of Modified Rodrigues Parameters over classical Euler angles parametrizations

(2) Proposes a Simultaneous Localization and Mapping framework tailored to small body operations featuring a Lidar-equipped Spacecraft

(3) Generalizes existing Lidar Model-Based Navigation algorithms to handle higher-order Bezier shape models featuring uncertainty in their spin rate and control mesh

(4) Develops a linearized uncertainty model providing an approximation of the ray tracing statistical range error arising from a stochastic Bezier shape model of arbitrary order

(5) Designs a maximum-likelihood procedure enabling the determination of the uncertainty in surface elements from point-cloud-to-shape fitting residuals

(6) Derives a linearized uncertainty model enabling first-order uncertainty quantification in the volume, center-of-mass, inertia tensor of a stochastic small body shape

(7) Produces the expressions of the partial derivative in the constant-density Polyhedron Gravity Model relative to the shape control points, allowing linearized uncertainty quantification in the potential and acceleration of gravity about a small body to take place

1.7.2 Dissertation Outline

This dissertation is organized like so:

Chapter 2 goes over the notations pertaining to reference frames and rotations, which are paramount to the derivations in this thesis, in addition to defining the force and torque models considered herein.

Chapter 3 describes the instrument model and point-cloud processing methods used throughout this thesis. This chapter defines the idealized Flash-Lidar sensor model around which the methods developed in this thesis are built, and goes over the implementations of the Iterative-Closest-Point and Bundle adjustment algorithms that were tailored to the problem at hand.

Chapter 4 details the reconstruction and fitting of a small body shape by means of range images, and details a robust two-step shape reconstruction framework featuring Poisson Surface Reconstruction and linear fitting of Bezier triangles of arbitrary degree.

Chapter 5 develops a linearized range uncertainty model able to capture the ray-traced range uncertainty in reconstructed shapes for on-board use. This chapter proposes tuning techniques fitting the model parameters to the uncertainty indeed present in the reconstructed shape, allowing the model to be evaluated to produce a range uncertainty estimate at any location on the shape.

Chapter 6 delves into the definition of a linearized uncertainty model arising from a stochastic shape model and its application to inertia parameters statistics computation. The methods developed in this chapter ultimately permit to gain insight into the stochastics of inertia parameters (volume, center of mass and inertia tensor) arising from surface uncertainty.

Chapter 7 details the derivation of a linearized polyhedron gravity model about a shape of reference, which leads to the expressions of the analytical gravity potential variance and acceleration covariance arising from an uncertain shape. The proposed method allows for the determination of predicted gravity uncertainties around and at the surface of a small body, for a lesser computational cost that Monte-Carlo simulations.

Chapter 8 presents the generalization of Dietrich's filter to relative navigation, in addition to the estimation of the small body's attitude state and standard gravitational parameter, leveraging the shape reconstruction pipeline and the shape uncertainty model previously derived.

Chapter 9 details the performance and robustness of the proposed small body survey, mapping and navigation algorithm by combining the methods from Chapters 3, 4, 5 and 8.

Finally, Chapter 10 lists a selection of research leads that could be explored to generalize the methods and results developed in this thesis.

1.7.3 Publications

1.7.3.1 Journal Papers

Accepted

- Bercovici, B., & McMahon, J. W. (2019). Robust Autonomous Small Body Shape Reconstruction and Relative Navigation using Range Images (Accepted). Journal of Guidance, Control, and Dynamics.

- Bercovici, B., & McMahon, J. W. (2019). Inertia Parameter Statistics of An Uncertain Small Body Shape (Accepted, In Press). Icarus.

- Bercovici, B., & McMahon, J. W. (2017). Point-Cloud Processing Using Modified Rodrigues Parameters for Relative Navigation. Journal of Guidance, Control, and Dynamics, 40(12). https://doi.org/https://arc.aiaa.org/doi/abs/10.2514/1.G002787

- Venigalla, C., Baresi, N., Aziz, J. D., Bercovici, B., Brack, D. N., Dahir, A., Van wal, S. (2019). The Near-Earth Asteroid Characterization and Observation (NEACO) Mission to Asteroid (469219) 2016 HO3 (In Print). Journal of Spacecraft and Rockets.

In Preparation

- Bercovici, B., Panicucci, P. & McMahon, J. W. Analytical Uncertainty Quantification In The Polyhedron Gravity Model of An Uncertain Small Body. Celestial Mechanics and Dynamical Astronomy

- Panicucci, P., Bercovici, B. & McMahon, J. W. Analytical Uncertainty Quantification In The Spherical Harmonics Gravity Expansion of An Uncertain Small Body. Celestial Mechanics and Dynamical Astronomy

- Bercovici, B. & McMahon, J. W. Improvements to Small Body Autonomous Survey, Mapping and Model-Based Navigation. Journal of Guidance, Control, and Dynamics

1.7.3.2 Conference Proceedings

- Bercovici, B., & McMahon, J. W. (2019). Lidar-Based Autonomous Shape Reconstruction and Navigation about Small Bodies Under Uncertainty. In 42nd annual AAS Guidance, Navigation and Control Conference (pp. 112).

- Bercovici, B., & McMahon, J. W. (2019). Initial Orbit Determination About Small Bodies Using Flash Lidar And Rigid Transform Invariants. In 29th AAS/AIAA Space Flight Mechanics Meeting, Kaanapali, HI (pp. 121).

- Bercovici, B., & McMahon, J. W. (2018). Inertia Parameter Statistics of An Uncertain Small Body Shape. In 50th Meeting of the American Astronomical Society Division for Planetary Science.

- Bercovici, B., & McMahon, J. W. (2018). A Consistent Small Body Navigation Filter Using Flash-Lidar Data and Bezier Triangles. In 2018 AAS/AIAA Astrodynamics Specialist Conference (pp. 120).

- Bercovici, B., & McMahon, J. W. (2018). Autonomous Shape Determination Using Flash-Lidar Observations and Bezier patches. In Proceedings of the 41st Annual Guidance and Control Conference.

- Bercovici, B., Dietrich, A., & McMahon, J. W. (2017). Autonomous Shape Estimation and Navigation About Small Bodies Using Lidar Observations. In Proceedings of the 2017 AAS/AIAA Astrodynamics Specialist Conference (pp. 120). Stevenson.

- Venigalla, C., Baresi, N., Aziz, J., Bercovici, B., Motta, G. B., Brack, D., Van Wal, S. (2018). The Near-Earth Asteroid Characterization and Observation (NEACO) mission. In Advances in the Astronautical Sciences (Vol. 162).

- Bercovici, B., & McMahon, J. W. (2017). Improved Shape Determination for Autonomous State Estimation. In Proceedings of the 31st ISTS , Matsuyama, Japan.

- Bercovici, B., & McMahon, J. W. (2017). An improved MRP-based Iterative Closest Point-To-Plane Algorithm. In 27th AAS/AIAA Space Flight Mechanics Meeting (pp. 120). San Antonio.

- Bercovici, B., & McMahon, J. W. (2017). Autonomous Shape Determination Using Flash-Lidar Observations. In Proceedings of the 27th AAS/AIAA Space Flight Mechanics Meeting (pp. 114). San Antonio.

1.7.3.3 Posters

- Bercovici, B., & McMahon, J. W. (2018). The Small Body Geophysical Analysis Toolbox (SBGAT). AAS Division for Planetary Science Conference. Knoxville, TN

- Bercovici, B., & McMahon, J. W. (2017). The Small Body Geophysical Analysis Toolbox (SBGAT). AAS Division for Planetary Science Conference. Provo, UT

- Bercovici, B., & McMahon, J. W. (2017). The Small Body Geophysical Analysis Toolbox (SBGAT). Lunar and Planetary Science Conference. The Woodlands, TX

Figure 1.3: IBM 704 computer similar to NASA Ames' at the time of Kalman's historical paper (courtesy of NASA)

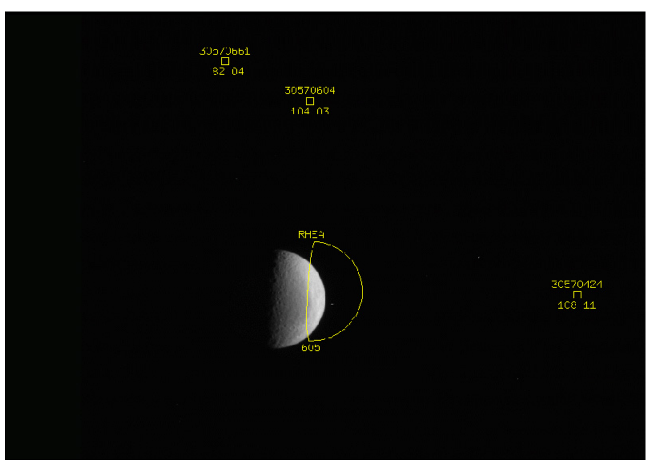
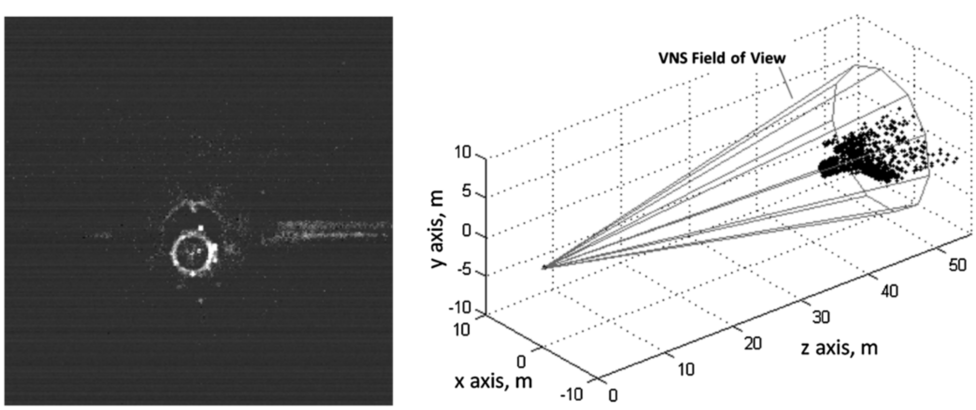


Figure 1.4: One of the last OPNAV images collected by Cassini in September 2016. The yellow overlay designates expected locations of known stars and Rhea's limb. The mismatch between the limb and the projected overlay can be tied to uncertainties in the spacecraft's trajectory as well as in the moon's ephemerids (courtesy of NASA/JPL-Caltech/Space Science Institute)

Figure 1.5: The PRISMA chaser/client spacecraft Mango and Tango [1]



a) Intensity image

b) Range image converted to 3-D point cloud

Figure 1.6: Example Flash Lidar image of the ISS collected during the STORRM flight experiment on STS-134 in May 2011 [2]

Figure 1.7: SPHERES with Astronaut Kelly on board the ISS (courtesy of the MIT Space System Laboratory)

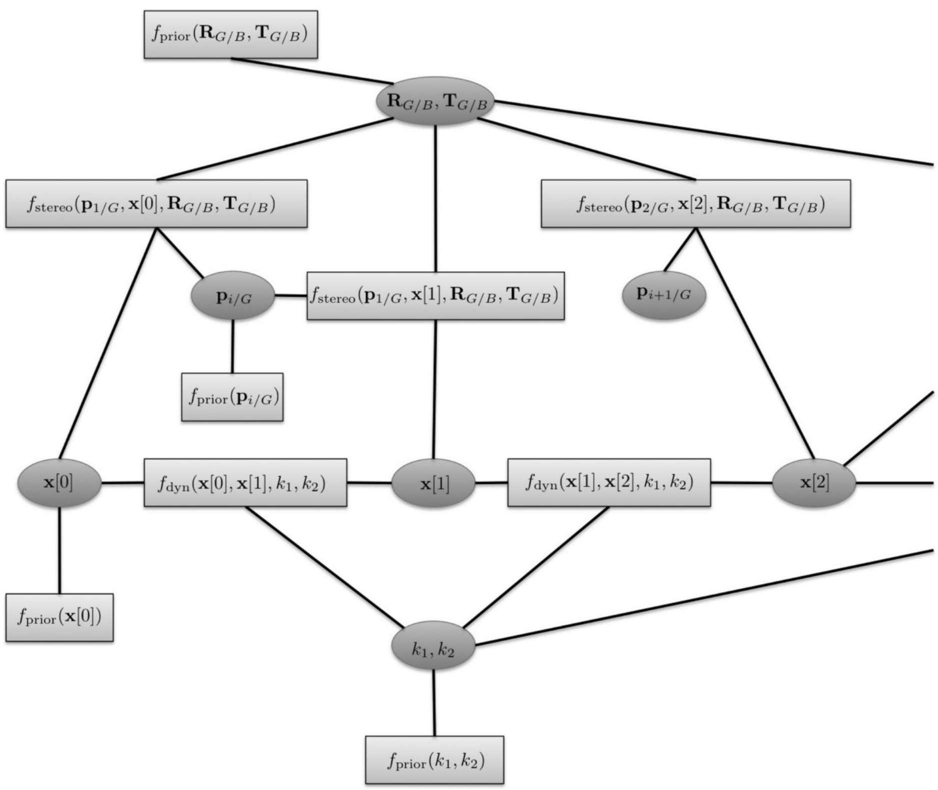


Figure 1.8: Factor-graph from [3] after three time steps. The variables and edges respectively denote random variables and the corresponding joint probability distributions

Chapter 2

Frames and dynamical models

2.1 Reference frames

Reference frames are fully defined by their origin - a position in space - and a set of three orthonormal vectors defining a proper orthonormal basis. The following paragraphs define the different reference frames used throughout this thesis. Reference frames are always written in a calligraphic font (like $\mathcal{N}$) whenever they are used in the text.

2.1.1 International Celestial Reference Frame $\mathcal{I}$

The International Celestial Reference Frame (ICRF) $\mathcal{I}$ is a quasi-inertial reference frame originating from the Solar System barycenter [67]. It was constructed by leveraging the quasi-fixed position of 212 distant extra-galactic radio sources, effectively defining a frame exhibiting virtually no noticeable rotational motion.

2.1.2 Barycentered inertial frame $\mathcal{N}$

The barycentered inertial frame $\mathcal{N}$ has its origin at the barycenter of the considered small body of interest. Its axes are inertially fixed and collinear with that of $\mathcal{I}$, such that the DCM $[\mathcal{IN}]$ is equal to the identity matrix.

2.1.3 Barycentered body-fixed principal frame $\mathcal{P}$

The principal body-fixed frame $\mathcal{P}$ has its origin at the barycenter of the considered small body of interest. Its axes are aligned with the principal axes of the inertia tensor. That is, the inertia tensor of the small body at the barycenter is diagonal when expressed in $\mathcal{P}$. Because the axes are fixed to the body, this this frame not inertially fixed.

2.1.4 Barycentered body-fixed frame $\mathcal{B}$

The barycentered body-fixed frame $\mathcal{B}$ has its origin at the barycenter of the considered small body of interest. Its axes are fixed with respect to the body topography, thus making this frame not inertially fixed. The orientation of these axes is constant with respect to those of $\mathcal{P}$, but arbitrary, so $\mathcal{B}$ and $\mathcal{P}$ do not necessarily overlap.

2.1.5 RIC frame $\mathcal{R}$

The Radial - In track - Crosstrack (RIC) frame $\mathcal{R}$ has its origin at the barycenter of the considered small body of interest. Its axes are defined from the position $\mathbf{r}$ and inertial velocity $\mathbf{v}$ of the spacecraft it is tracking. The first axis of the $\mathcal{R}$ frame is defined as the unit direction from the barycenter to the spacecraft. Its third axis is directed along $\mathbf{r} \times \mathbf{v}$, the orbit's angular momentum. Its second axis completes the triad in a counter-clockwise fashion.

2.1.6 Instrument frame $\mathcal{L}$

The instrument frame $\mathcal{L}$ has its origin at the spacecraft's barycenter. The first axis of this frame defines the instrument's bore sight. The other two axes completing the triad are unimportant to the definition of this frame. A crucial assumption made throughout this thesis is that the attitude of the spacecraft relative to the inertial frame of reference is always known. That is, the true DCMs $[\mathcal{LN}]$ and $[\mathcal{LI}]$ are assumed to be perfectly known at all times.

2.2 Frame conversions

2.2.1 Direction cosine matrix

The relative orientation of the axes of two frames $\mathcal{A}$ and $\mathcal{B}$ can be related through a Direction Cosine Matrix (DCM) noted $[\mathcal{AB}]$. DCMs are 3-by-3 real, proper orthonormal matrices, thus satisfying

$$[\mathcal{AB}]^{-1} = [\mathcal{AB}]^T \tag{2.1}$$

$$\det([\mathcal{AB}]) = +1 \tag{2.2}$$

Given a 3-by-1 vector $\mathbf{x}$, the expression of this vector in a specific frame (for instance the $\mathcal{B}$ frame) reads ${}^{\mathcal{B}}\mathbf{x}$, so as to inform the reader that the components are written in the $\mathcal{B}$ basis. The expression of the same vector in the $\mathcal{A}$ frame is then simply given by

$${}^{\mathcal{A}}\mathbf{x} = [\mathcal{AB}]\,{}^{\mathcal{B}}\mathbf{x} \tag{2.3}$$

The frame superscript is not used consistently throughout this thesis since it can be inferred from the definition of the DCM itself. Assuming that the $\mathcal{A}$ and $\mathcal{B}$ orthonormal basis vectors are respectively defined as $\hat{a}_x, \hat{a}_y, \hat{a}_z$ and $\hat{b}_x, \hat{b}_y, \hat{b}_z$, $[\mathcal{AB}]$ can be constructed from

$$[\mathcal{AB}] = \begin{bmatrix} {}^{\mathcal{B}}\hat{a}_x^T \\ {}^{\mathcal{B}}\hat{a}_y^T \\ {}^{\mathcal{B}}\hat{a}_z^T \end{bmatrix} = \begin{bmatrix} {}^{\mathcal{A}}\hat{b}_x & {}^{\mathcal{A}}\hat{b}_y & {}^{\mathcal{A}}\hat{b}_z \end{bmatrix} \tag{2.4}$$

In addition, the following notation

$$[\mathcal{AB}]^{-1} = [\mathcal{BA}] \tag{2.5}$$

holds true for all DCMs. Finally, it must be noted that a DCM describing the relative orientation of two frames is actually the unique DCM representative of this specific relative orientation. [68]

2.2.2 Rotation composition

Given three frames $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, converting the $\mathcal{A}$ frame components of ${}^{\mathcal{A}}\mathbf{x}$ to the $\mathcal{C}$ frame can be directly achieved through ${}^{\mathcal{C}}\mathbf{x} = [\mathcal{C}\mathcal{A}]^{\mathcal{A}}\mathbf{x}$, or decomposed into a succession of two rotations like so:

$$ {}^{\mathcal{C}}\mathbf{x} = [\mathcal{C}\mathcal{B}][\mathcal{B}\mathcal{A}]^{\mathcal{A}}\mathbf{x} \tag{2.6}$$

which naturally implies that $[\mathcal{C}\mathcal{A}] = [\mathcal{C}\mathcal{B}][\mathcal{B}\mathcal{A}]$. Conversely, the conversion of the $\mathcal{C}$ frame components of ${}^{\mathcal{C}}\mathbf{x}$ to the $\mathcal{A}$ frame is given by

$$ {}^{\mathcal{A}}\mathbf{x} = \left([\mathcal{C}\mathcal{B}][\mathcal{B}\mathcal{A}]\right)^{T\mathcal{C}}\mathbf{x} = [\mathcal{A}\mathcal{B}][\mathcal{B}\mathcal{C}]^{\mathcal{C}}\mathbf{x} \tag{2.7}$$

2.3 Dynamical models

2.3.1 Spherical-harmonics gravity acceleration

The gravitational potential originating from an irregular, constant-density small body of standard gravitational parameter μ can be written in terms of a spherical-harmonics expansion

$$ U = \frac{\mu}{r} + \frac{\mu}{r}\sum_{l=2}^{\infty}\sum_{m=0}^{l}\left(\frac{R}{r}\right)^{l} P_{l,m}\left(\sin\left(\phi\right)\right)\left[C_{l,m}\cos\left(m\lambda\right) + S_{l,m}\sin\left(m\lambda\right)\right] \tag{2.8}$$

r denotes the magnitude of the radius vector between the origin of the considered shape and the point at which the gravity potential must be evaluated, while $P_{l,m}$ denotes the Legendre polynomial of degree l and order m. λ and ϕ respectively denote the longitude and latitude of $\mathbf{r}$ expressed in the small body-fixed frame $\mathcal{B}$ [69]. Equation 2.8 is only valid outside of the Brillouin sphere, the sphere centered at the origin $\mathbf{r} \equiv \mathbf{0}$ circumscribing the surface from which the spherical harmonics expansion was computed. This is due to the powers of $\frac{R}{r}$ that will make the expansion diverge should this ratio become equal or greater than one.

The acceleration of gravity expressed in the $\mathcal{N}$ frame is thus given by

$$ \mathbf{a}_{\text{gravity}} = [\mathcal{B}\mathcal{N}]^{T}\nabla U = [\mathcal{B}\mathcal{N}]^{T}\frac{\partial U}{\partial^{\mathcal{B}}\mathbf{r}} \tag{2.9}$$

where the gradient of the potential has been differentiated with respect to the spacecraft's position expressed in the $\mathcal{B}$ frame.

The implementation of the acceleration arising from a gravity field spherical harmonics expansion was provided by the Small Body Geophysical Analysis Tool (SBGAT) [70]. SBGAT is also capable of evaluating the spherical harmonics coefficients over a constant-density polyhedral shape.

2.3.2 Third-body Sun gravity

The barycentered reference frame $\mathcal{N}$ in which the spacecraft is tracked is acted upon by the Sun, that exerts a point-mass gravitational pull on both the small-body and the spacecraft. As a result, the third-body gravity exerted by the Sun onto the spacecraft in the $\mathcal{N}$ frame reads

$$\mathbf{a}_{\mathrm{3rd-body}} = \mu_{\odot}\left(\frac{\mathbf{R}}{|\mathbf{R}|^3} - \frac{\mathbf{R}+\mathbf{r}}{|\mathbf{R}+\mathbf{r}|^3}\right) \tag{2.10}$$

where $\mathbf{R}$ and $\mathbf{r}$ respectively denote the small-body geocentric position and spacecraft position relative to the small-body center, both expressed in the $\mathcal{N}$ frame, and $\mu_{\odot}$ the Sun's standard gravitational parameter.

2.3.3 Solar radiation pressure

The expression of the acceleration caused by the Solar Radiation Pressure (SRP) interacting with a sphere of uniform optical properties (referred to as the *cannonball* SRP model) is given by [69]

$$\mathbf{a}_{\mathrm{SRP}} = \frac{\Phi}{c} C_r \frac{A}{m} \frac{\mathbf{R}+\mathbf{r}}{\|\mathbf{R}+\mathbf{r}\|} \tag{2.11}$$

Φ is equal to the Solar flux evaluated at the small body of interest while c represents the speed of light in vacuum. The SRP cannonball coefficient C_r is valued over the $[0,2]$ interval, and the area-to-mass ratio $\frac{A}{m}$ is set to the fixed value of 0.01 m^2/kg. $\mathbf{R}$ and $\mathbf{r}$ hold the same significance as in the expression of the third-body gravity acceleration.

The cannonball model was augmented by means of a simple eclipse-check: a ray is traced from the spacecraft position $\mathbf{r}$ towards the Sun. The spacecraft is lit and affected by SRP if this ray does not intersect with the shape. If the ray intersects, then the magnitude of the SRP force is nullified as the spacecraft is considered in the shadow of the small body.

Torque-free small-body barycentric rotation The attitude of the small body tracked through the DCM $[\mathcal{BN}]$ is governed by the torque-free Euler equation [68]

$$[I]\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} = -\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times [I]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \tag{2.12}$$

where $[I]$ denotes the inertia tensor evaluated at the small body barycenter and $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}$ the angular velocity vector, both expressed in the in the $\mathcal{B}$ frame. The kinematics of the underlying MRP parametrization of the DCM $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}}$ is given by [68]

$$\dot{\boldsymbol{\sigma}}_{\mathcal{B}/\mathcal{N}} = \frac{1}{4}\left[\left(1 - \|\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}}\|^2\right) I_3 + 2[\widetilde{\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}}}] + 2\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}}\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}}^T\right]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \tag{2.13}$$

Chapter 3

Data acquisition

3.1 Instrument model

Although the procedures developed in this thesis are suitable to any range sensor, including scanning-Lidar and Flash-Lidar systems, Flash-Lidar instruments are the focus of this work. A Flash-Lidar proceeds by acquiring a collection of 3D points, dubbed *point cloud* over the surface of the target of interest. A 3D point measured over the target of interest $\tilde{\mathbf{P}}_i$ is obtained by casting a laser ray along a measurement direction $\hat{u}_i$. The range ρ_i is measured between red the sensor origin $\mathbf{r}$ and the impact point $\tilde{\mathbf{P}}_i$. The measurement is affected by an error along the line-of-sight μ_i modeled as a Gaussian variable of zero-mean and standard deviation σ. This measurement geometry is shown on Figure 3.1, along with the sensor's field-of-view (fov) and focal length (f). In so many words,

$$\tilde{\mathbf{P}}_i = (\rho_i + \mu_i)\,\hat{u}_i + \mathbf{r} \tag{3.1}$$

Operating the instrument and imaging the target at two neighboring times yield two overlapping point clouds, that differ in content because of the motion of the targeted body relative to the instrument. The point-clouds must be aligned with respect to each other before further processing (such as shape fitting) can take place. This is done by means of an instance of the iterative closest point (ICP) algorithm. The ray-tracing procedure can be considerably accelerated by means of space-partitioning techniques, like the *kd-tree* shown on Figure 3.2. kd-trees are binary tree structures enclosing recursive rectangular boxes that capture subdomains of the enclosed shape. The complexity of an intersection query between a ray and a shape comprised of N triangular planar

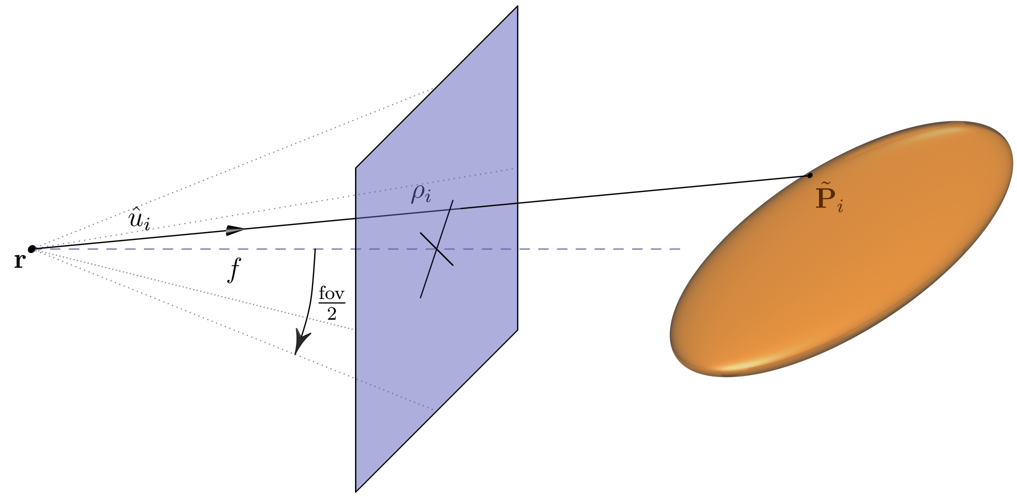


Figure 3.1: Flash Lidar instrument model

elements thus approaches $\log N$ on average for the kd-tree search as opposed to Nin the brute-force query case [71] .

3.2 ICP Registration

Point-cloud registration pertains to the computation of point pairs between a source and a destination point cloud, followed by the calculation of the rigid transform minimizing a relative distance between these point pairs [72]. The resulting point-clouds can then be combined in a common frame, effectively "stitching" them together. One can then obtain the shape model of the target, compute an estimate of the rigid transform the target has undergone between the two observations times or use the shape model in a relative navigation filter [73]. This registration process is computationally demanding, and given limited on-board resources, could thus benefit from alternative formulations improving its accuracy and its speed [74].

There exists an abundant body of literature covering several implementations of the Iterative Closest Point (ICP) algorithm dealing with three-dimensional point clouds [72] [75] [74]. The baseline method consists in parametrizing the rigid transform in terms of a translation vector and a set of Euler angles, under the assumption of small misalignment. This process results in a linear system that can be directly solved for the translation vector and the Euler angles sequence. The

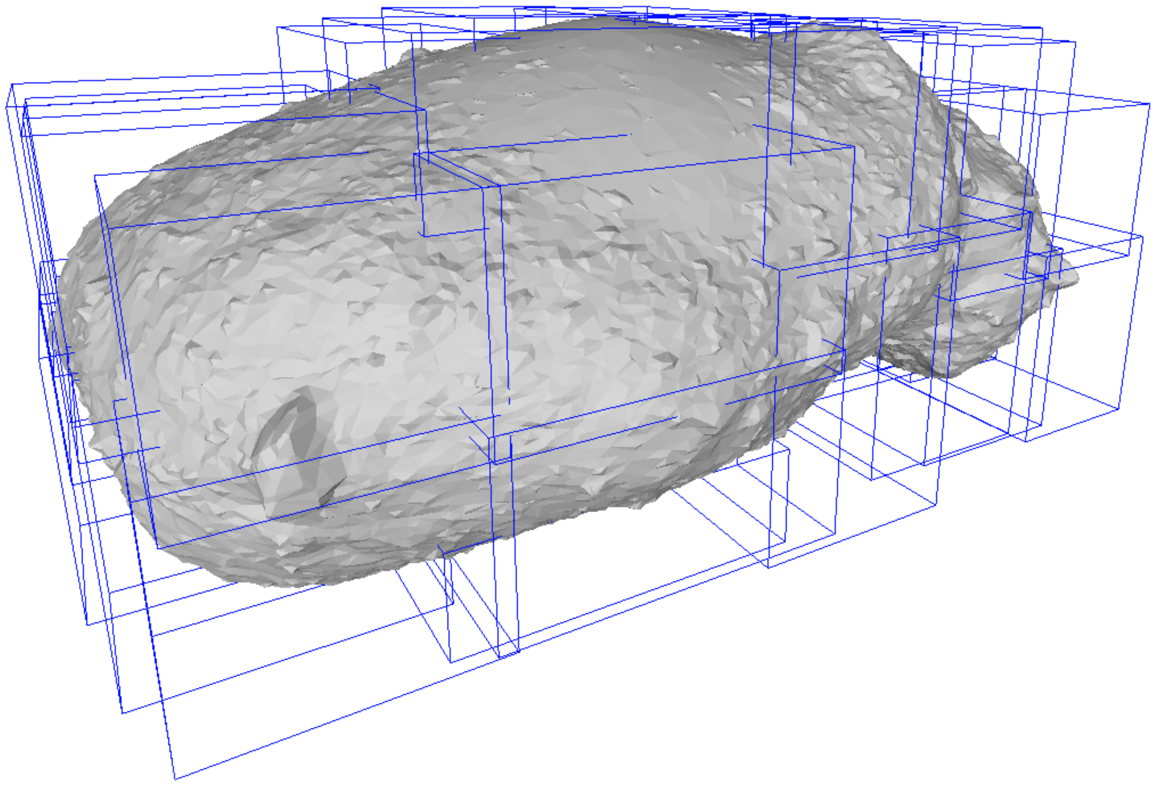


Figure 3.2: *kd-tree* leaf nodes around an asteroid shape model. In reality, the nodes would be much tighter so as to only encompass a few dozen facets each.

algorithm is then iterated until a convergence criterion is satisfied, indicating that a local minimum of the associated cost function has been found.

While Euler angles-based point cloud registration algorithms perform well when small rotations are indeed present, they are not well suited to the estimation of large rotational transforms as large linearization errors hinder fast convergence [76]. This could prevent autonomous on-board operation, as unmanned vehicles are often limited in computational power and could be undergoing time-constrained tasks. Like any 3-parameter attitude sets, Euler angles will also become singular for some critical orientations [77]. Techniques have been devised to appropriately switch away from a singular Euler angle sequence to a better-behaved sequence, but implementation issues make these unpractical [78]. One contribution of this thesis is to propose the use of an alternative, well-behaved attitude parametrization to represent the rotational component of the rigid transform: Modified Rodrigues Parameters (MRP). Their rising popularity in the spacecraft attitude control area is justified by their mathematical properties that allow tracking of large rotations as well as the existence of asymptotic stability proofs associated with this class of attitude parameters [79]. These properties should help to improve convergence of the ICP algorithms at large rotations and avoid singularity issues.

3.2.1 Summary of Modified Rodrigues Parameters

MRP are a relatively recent addition to the minimum attitude parameter set family [80]. Unlike higher-order attitude sets such as quaternions, they do not need to obey any additional algebraic constraint to effectively represent rotations. This property facilitates optimal attitude determination by means of MRP, as the underlying optimization problem is no longer constrained [81]. In addition, the MRP singularity occurring for rotations reaching 360° of magnitude can be avoided by switching to the so-called MRP shadow set before this singularity is reached [82]. The MRP and its shadow set effectively represent the "short" and the "long" rotation sequences representative of the same attitude. This makes MRP ideally suited to large rotations.

The mathematical definition of a MRP set is as follows:

$$\boldsymbol{\sigma} = \tan\left(\frac{\Phi}{4}\right)\hat{e} \tag{3.2}$$

where Φ is the principal rotation angle and $\hat{e}$ the unit vector directing the principal rotation axis [83]. Another advantage of MRP sets over Euler angles in the parametrization of a rotation resides in the very linear behavior of the application $\theta \mapsto \tan\left(\frac{\theta}{4}\right)$, enabling one to retain much more information in the first order partial derivative of the Direction Cosine Matrix (DCM) with respect to $\boldsymbol{\sigma}$ compared to its alternative formulation in terms of Euler angles. Figure 3.3 shows how the norm of Euler angles and MRP compare to purely linear extrapolations with respect to the principal rotation magnitude. For this example, the principal rotation axis direction was set to $\hat{e} = \frac{1}{\sqrt{3}}[1\ 1\ 1]^T$. It can be observed that a linear approximation of the norm of the 321 Euler angle set $[\theta_1\ \theta_2\ \theta_3]^T$ deviates by more than 5% from its true value at $\Phi = 43.6°$, whereas the norm of the corresponding MRP $\sigma = \|\boldsymbol{\sigma}\|$ can be captured linearly within this tolerance up to $\Phi = 90.9°$. This well-behaved, more linear behavior of MRP is fundamental in explaining the performance of the MRP-based registration algorithm [84].

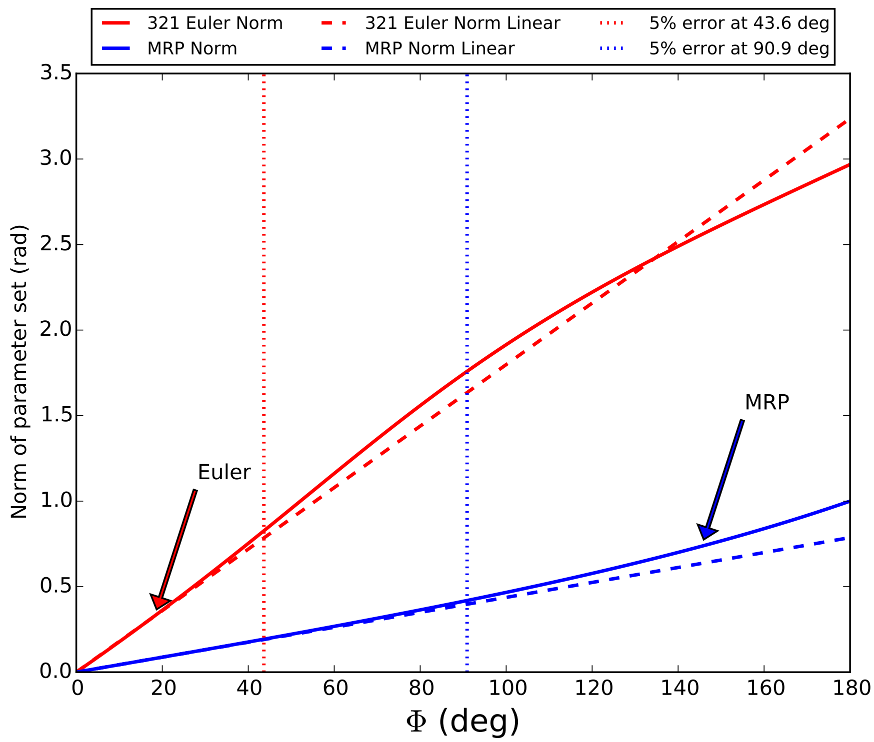


Figure 3.3: Euler 321 and MRP sets norms against the principal rotation angle Φ

3.2.2 General formulation of the iterative alignment algorithm

3.2.2.1 Notations

A rigid transform is comprised of a rotational and a translational component. This transform can be compactly expressed as $(M, \mathbf{x})$, where M stands for the orthogonal Direction Cosine Matrix (DCM) representing the transform rotation while $\mathbf{x}$ denotes the translational component. This DCM maps the coordinates of a vector expressed in a departure frame (of source frame) to an arrival frame (or destination frame). The rigid transform can be parametrized as

$$(M, \mathbf{x}) = (M(\boldsymbol{\mathcal{X}}), \mathbf{x}) \tag{3.3}$$

where $\boldsymbol{\mathcal{X}}$ is an attitude parametrization not yet defined. By consequent, the state of the rigid transform is represented by the following vector $\boldsymbol{X}$

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{\mathcal{X}} \\ \mathbf{x} \end{pmatrix} \tag{3.4}$$

3.2.2.2 Iterative Closest-Point-To-Plane cost function

A Lidar instrument returns a sequence of ranges associated with line-of-sight directions, from which point coordinates can be extracted. Given two point clouds of size N, $P = \{\mathbf{S}_i\}$ and $Q = \{\mathbf{D}_i\}$ (respectively denoted *source* and *destination* point-cloud), registration pertains to computing the rigid transform $(M, \mathbf{x})$ that best aligns P and Q. In the context of the Iterative Closest Point-to-Plane (ICP2P) framework, the cost function to minimize is [85]

$$J^2 = \sum_{i=1}^{N} \left(\hat{n}_i^T \left(M\mathbf{S}_i + \mathbf{x} - \mathbf{D}_i\right)\right)^2 \tag{3.5}$$

where $(\mathbf{S}_i, \mathbf{D}_i) \in \mathbb{R}^3 \times \mathbb{R}^3$ is the i-th pair of points to match, and $\hat{n}_i$ the unit normal vector of the destination point cloud computed at $\mathbf{D}_i$. This expression assumes that the pairs $(\mathbf{S}_i, \mathbf{D}_i)$ have already been formed so that there indeed exists a correspondence between the two points. ICP2P has been shown to converge faster than the classical Iterative Closest Point-to-Point algorithm, in addition to being better suited to matching a point cloud to an existing shape mode [86].

3.2.2.3 Batch formulation

An iterated batch filter framework can be used to compute the transform state minimizing Equation (3.5). This filter can be used to estimate a state by comparing collected and generated observations, mapped into the state estimate by means of the observation model first-order partial derivatives with respect to the state. Utilizing this filter requires the following:

- Formulating an estimated state vector $\boldsymbol{X}$.
- Expressing the observation residuals as a cost function to be minimized.
- Deriving the observation model G relating the state $\boldsymbol{X}$ to collected observations, and its associated partial derivatives.
- Deriving the dynamics of the estimated state (if any) and its associated partial derivatives.
- Choosing a reference a-priori state $\boldsymbol{X}^*$ to start iterating over.

The estimated state $\boldsymbol{X}$ and the cost criterion J^2 have already been defined in Equation (3.4) and Equation (3.5). The observation model G and its partial derivatives are derived in the following sections.

3.2.2.4 Observation model

An observation model can be extracted from the expression of the cost function. The i-th collected observation and computed observation (respectively denoted Y_i and G_i) are defined as

$$Y_i = \hat{n}_i^T \mathbf{D}_i \tag{3.6}$$

$$G_i(\boldsymbol{\mathcal{X}}, \mathbf{x}) = \hat{n}_i^T \left(M(\boldsymbol{\mathcal{X}}) \mathbf{S}_i + \mathbf{x} \right) \tag{3.7}$$

Equation (3.5) thus becomes

$$J^2 = \sum_{i=1}^{N} \left(Y_i - G_i(\boldsymbol{\mathcal{X}}, \mathbf{x}) \right)^2 \tag{3.8}$$

Introducing a reference state $\boldsymbol{X}^* = \begin{pmatrix} \boldsymbol{\mathcal{X}}^* \\ \mathbf{x}^* \end{pmatrix}$, linearizing the observation model about this reference yields

$$\begin{aligned} G_i(\boldsymbol{\mathcal{X}}, \mathbf{x}) &= G_i(\boldsymbol{\mathcal{X}}^*, \mathbf{x}^*) + \begin{pmatrix} \frac{\partial G_i}{\partial \boldsymbol{\mathcal{X}}} & \frac{\partial G_i}{\partial \mathbf{x}} \end{pmatrix}_{\boldsymbol{X} = \boldsymbol{X}^*} \cdot \begin{pmatrix} \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^* \\ \mathbf{x} - \mathbf{x}^* \end{pmatrix} + \text{H.O.T} \\ &= G_i(\boldsymbol{\mathcal{X}}^*, \mathbf{x}^*) + H_i \delta \boldsymbol{X} + \text{H.O.T} \end{aligned}$$

Retaining only the first-order partials thus provides

$$G_i(\boldsymbol{\mathcal{X}}, \mathbf{x}) = G_i(\boldsymbol{\mathcal{X}}^*, \mathbf{x}^*) + H_i \delta \boldsymbol{X} \tag{3.9}$$

With

$$\delta \boldsymbol{X} = \begin{pmatrix} \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^* \\ \mathbf{x} - \mathbf{x}^* \end{pmatrix} = \boldsymbol{X} - \boldsymbol{X}^* \quad \text{state deviation vector} \tag{3.10}$$

$$H_i = \begin{pmatrix} \frac{\partial G_i}{\partial \boldsymbol{\mathcal{X}}} & \frac{\partial G_i}{\partial \mathbf{x}} \end{pmatrix}_{\boldsymbol{X} = \boldsymbol{X}^*} \quad \text{state} - \text{observation matrix} \tag{3.11}$$

By observation of Equation (3.7), it is clear that

$$\left. \frac{\partial G_i}{\partial \mathbf{x}} \right|_{\mathbf{x} = \mathbf{x}^*} = \hat{n}_i^T \tag{3.12}$$

The derivation of the other partial differential $\left. \frac{\partial G_i}{\partial \boldsymbol{\mathcal{X}}} \right|_{\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}^*}$ is case specific and will be addressed in the following sections. Now define the i-th prefit residuals y_i and observation errors ϵ_i

$$y_i = Y_i - G_i(\boldsymbol{\mathcal{X}}^*, \mathbf{x}^*)$$

$$\epsilon_i = y_i - H_i \delta \boldsymbol{X}$$

3.2.3 Normal equations

With those quantities defined, Equation (3.5) can be rewritten as

$$J^2 = \sum_{i=1}^{N} \epsilon_i^2 = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \tag{3.13}$$

With

$$\boldsymbol{\epsilon} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_N \end{pmatrix} - \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_i \\ \vdots \\ H_N \end{pmatrix} \delta \boldsymbol{X}$$

$$= \mathbf{y} - H\delta \boldsymbol{X}$$

Hence,

$$J^2 = (\mathbf{y} - H\delta \boldsymbol{X})^T (\mathbf{y} - H\delta \boldsymbol{X}) \tag{3.14}$$

The solution $\delta \boldsymbol{X}$ minimizing J^2 has to be a stationary point of (3.14). Taking the first partial derivative of Equation (3.14) with respect to $\delta \boldsymbol{X}$ and equating it to zero thus yields the so-called normal equations

$$\Lambda \delta \boldsymbol{X} = \boldsymbol{\nu} \tag{3.15}$$

where the *information matrix* and *normal matrix* are respectively defined as

$$\Lambda = H^T H = \sum_{i=1}^{N} H_i^T H_i$$

$$\boldsymbol{\nu} = H^T \mathbf{y} = \sum_{i=1}^{N} H_i^T y_i$$

Under the condition of a fully observable system (i.e the information matrix is full rank), the solution to the normal equations is given by

$$\delta \boldsymbol{X} = \Lambda^{-1} \boldsymbol{\nu} \tag{3.16}$$

It is worth noting that alternative techniques yielding the solution to Equation (3.16) can be preferred to the explicit computation of the covariance matrix Λ^{-1}. In particular, orthogonal methods such as Householder transformations are better suited to large-dimensional and ill-conditioned problems[87]. The rigid transform state is then updated through

$$\boldsymbol{X} = \boldsymbol{X}^* + \delta\boldsymbol{X} \tag{3.17}$$

The alignment algorithm can then be iterated by repeating this process until a stopping criterion is satisfied.

3.2.4 Linearized multiplicative formulation

The work of Markley[88] has illustrated the benefits of using a multiplicative rotation parametrization instead of an additive one for the purpose of attitude estimation. Specifically, this formulation respects the formalism of rotation composition which relies on multiplication of DCMs, rather than on the addition of there respective parametrizations. This representation is achieved by rewriting the DCM evaluated at the current attitude set $\boldsymbol{\mathcal{X}}$ as

$$M(\boldsymbol{\mathcal{X}}) = M(\delta\boldsymbol{\mathcal{X}})M(\boldsymbol{\mathcal{X}}^*) \tag{3.18}$$

where $\delta\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}}^*$ are respectively the deviation parameter set and the a-priori parameter set. Note that this expression does not suppose that $\delta\boldsymbol{\mathcal{X}}$ is small. If one assumes that $\delta\boldsymbol{\mathcal{X}}$ is indeed small, one can linearize Equation (3.18) about $\delta\boldsymbol{\mathcal{X}} = \mathbf{0}$ to obtain a first-order attitude parametrization of the DCM. This expression specializes into the following two, depending on whether 321 Euler angles or MRP are used:

$$M(\boldsymbol{\sigma}) \simeq (I_3 - 4[\delta\boldsymbol{\sigma}])M(\boldsymbol{\sigma}^*)$$

$$M(\boldsymbol{\Theta}) \simeq (I_3 - [T\delta\boldsymbol{\Theta}])M(\boldsymbol{\Theta}^*)$$

where

$$\boldsymbol{\Theta} = \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 \end{pmatrix}^T \tag{3.19}$$

and

$$T = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{3.20}$$

Substituting these expressions of the DCM into the observation model, the registration problem reduces to the underdetermined equations

$$G_i(\mathbf{\Theta}, \mathbf{x}) = \hat{n}_i^T M(\Theta^*)\mathbf{S}_i + \hat{n}_i^T [M(\mathbf{\Theta}^*)\mathbf{S}_i] T \delta\mathbf{\Theta}$$

$$G_i(\boldsymbol{\sigma}, \mathbf{x}) = \hat{n}_i^T M(\sigma^*)\mathbf{S}_i - 4\mathbf{S}_i^T M^T(\boldsymbol{\sigma}^*)[\hat{n}_i]\delta\boldsymbol{\sigma}$$

The resulting equations are linear in the deviations $\delta\mathbf{\Theta}$ and $\delta\boldsymbol{\sigma}$ and can thus be readily incorporated within the same batch framework presented earlier. The contribution of all G_i to the determination of $\delta\boldsymbol{\mathcal{X}}$ results into a set of normal equations that can be solved for.

3.2.5 Point-pair matching

The complete ICP algorithm requires the computation of correspondence pairs between the source and destination points. Among the many different point-pair matching schemes that have been proposed, the *closest point - compatible normal* method described in Rusinkiewicz and al. [85] was picked. This technique proceeds by associating each source point to the closest destination point if the normals evaluated at these two points are within a tolerance angle. This metric is consistent because it validates more point-pairs as the alignment between the two point clouds get better aligned. This technique was chosen to simulate the full ICP algorithm using the different formulations previously described. Two normals were considered as "compatible" if they were separated by less than 45°.

3.2.6 Results and discussion

The performance of the two alignment algorithms were compared by means of a two-step protocol. The sensitivity of both algorithms to the point cloud sizes and the magnitude of the rigid transform to estimate was first tested by means of a Monte-Carlo simulation exploring the rigid transform space. Their efficiency when dealing with structured, realistic point clouds affected by noise was then compared. It was first assumed that a perfect pairing between the source and destination point cloud was available, effectively treating the rest of the registration algorithm as a black box providing us with the correct point matching. The final step consisted in introducing random pairing error to analyze the performance of both implementations when provided with an imperfect pair-matching. Both algorithms were initialized with a trivial a-priori rigid transform $\boldsymbol{X} = [0,0,0,0,0,0]^T$.

3.2.6.1 Concept validation by Monte-Carlo simulation using perfect point pairs

The two algorithms were first compared by means of a Monte-Carlo simulation exploring the rigid transform parameter space. This first comparison assumes that perfect pairing between the source and destination point cloud is available. This assumption will not hold in the next section.

Destination point clouds of increasing size N were generated by collecting range-measurements over a boulder-like shape and retaining a random selection of N of those measurements. Corresponding source points clouds were obtained by cloning the destination point clouds and applying them a random rigid transform, to be later estimated by the registration algorithm. These point clouds of increasing resolution are shown on Figure 3.4 with the resolution increasing from left to right, top to bottom.

The rigid transform applied to each point cloud was a combination of a rotational component M and a translational component $\mathbf{x}$. The translational component was randomly sampled from a uniform distribution over $[-0.1, 0.1]$ meters, which is of the same order of magnitude as the targeted shape's size. The rotational component was formulated in terms of a sequence of 321 Euler angles uniformly sampled in $[0, 2\pi] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times [0, 2\pi]$ respectively. $10,000$ random rigid transforms were generated this way and estimated by means of both algorithms. The point-pairs were assumed to be readily available and no noise was introduced. The convergence threshold was set to $J_c = 10^{-5}$ m.

Figure 3.7 shows the mean of the iterations difference between the two algorithms over all realizations of the Monte-Carlo simulation. The negative difference indicates a faster performance of the MRP independently from the point cloud size. Figure 3.7 denotes a faster convergence of the MRP-based algorithm consistently over all the point cloud sizes by nearly 2 iterations in average. Figure 3.6 provides the mean of residuals for the MRP-based algorithm when both algorithms converge in the same number of iterations. The MRP-based algorithm features smaller residuals in average than the Euler angles-based method, with also a smaller spread about the mean.

Figure 3.5 presents the distribution of the iteration-to-convergence count for the multiplicative ICPs. The filter appears to perform better when MRP are used to parametrize the rotational component of the rigid transform. In addition, there appears to be no influence of the point cloud size on the convergence rate once the data sets are large enough.

Additional insight can be obtained by looking at the performance of each method for a specific point cloud size. The $N = 5000$ case was thus investigated. The distribution of the 'speed' difference is clearly in favor of the MRP algorithm for this given point cloud as shown on Figure 3.7. The negative difference indicates a faster performance of the MRP algorithm by more than one iteration in average in the multiplicative case.. For all cases where Euler angles converged faster than MRPs (by no more than one iteration), the mean of the residuals of the MRP-based method was two orders of magnitude smaller than that of the Euler angles-based method.

Furthermore, the evolution of the algorithms' speed against the magnitude of the rigid transform rotation to estimate is shown on Figure 3.8. It is evident that all methods require more iterations to achieve convergence as the magnitude of the rotation becomes larger, since large rotations violate the linearization assumption that both algorithms are based on. Nevertheless, MRP perform better than Euler angles over the entire angle interval as they are systematically faster than the baseline algorithm. Their performance is even more evident when the magnitude of the rigid transform becomes large. Figure 3.9 provide a synthetic heat map showing the number of iterations required by each method to converge and the resulting residuals. It can be seen that the MRP-based algorithm spreads much less than the the Euler-angle based algorithm, as the latter is much more subject to requiring a large number of iterations to converge. This trend is confirmed by Figure 3.6, as it shows that residuals at convergence are lower in average when MRP are use compared to Euler angles when both parametrizations converge in the same number of iterations.

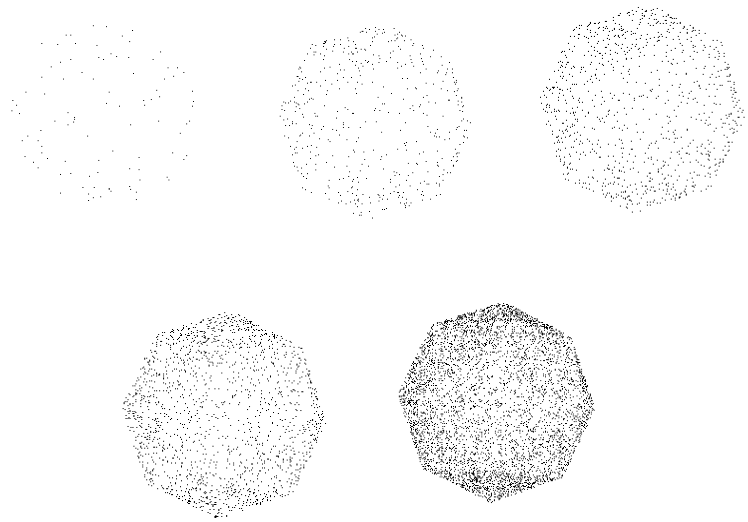


Figure 3.4: Point clouds used in the first Monte-Carlo simulation ($N = 100$, $N = 500$, $N = 1000$, $N = 1750$ and $N = 5000$)

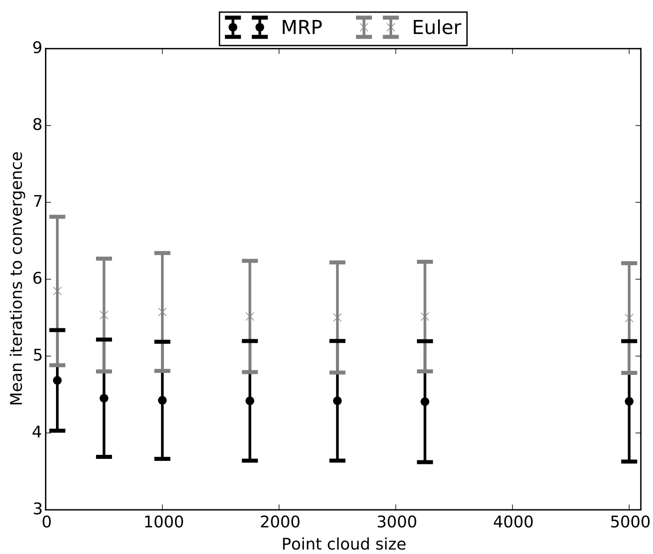


Figure 3.5: Iteration-to-convergence distribution against point cloud size for the linearized multiplicative ICPs, with one-standard deviation bounds.

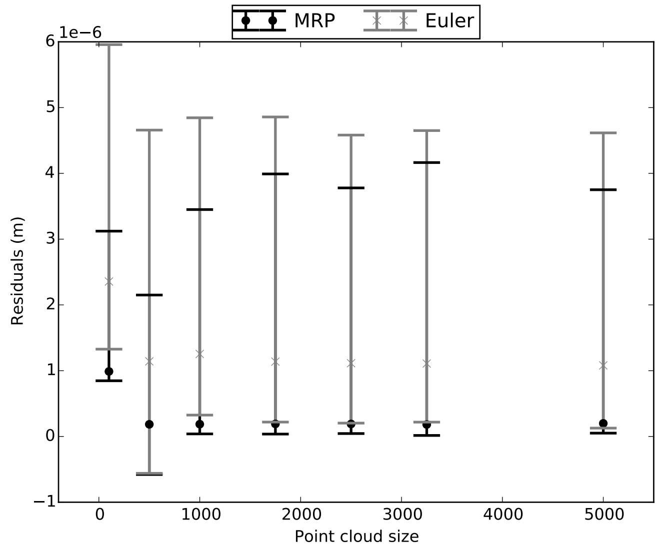


Figure 3.6: Mean of residuals over MC outcomes where multiplicative MRP and Euler angles-based ICP converge in the same number of iterations, with one-standard deviation bounds.

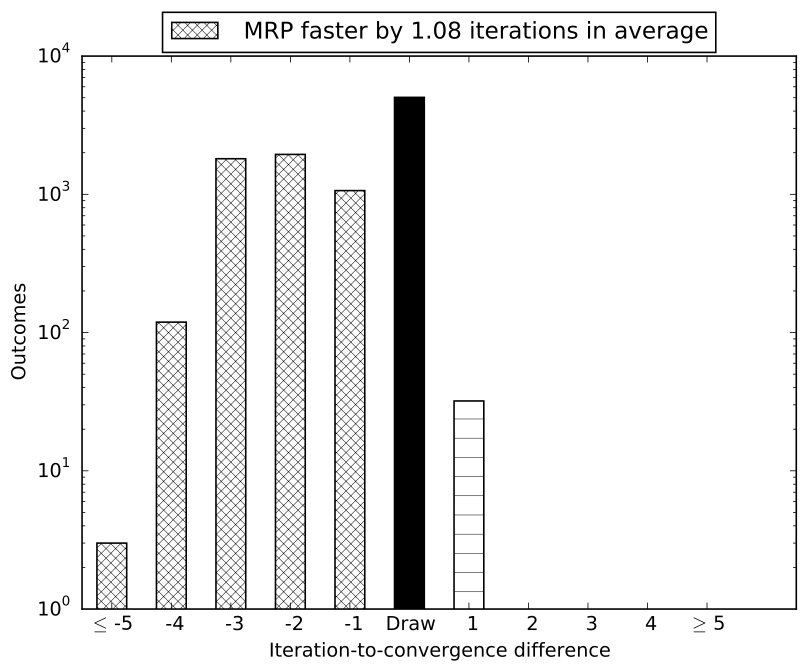


Figure 3.7: Distribution of the iteration-to-convergence difference for the multiplicative ICP($N = 5000$)

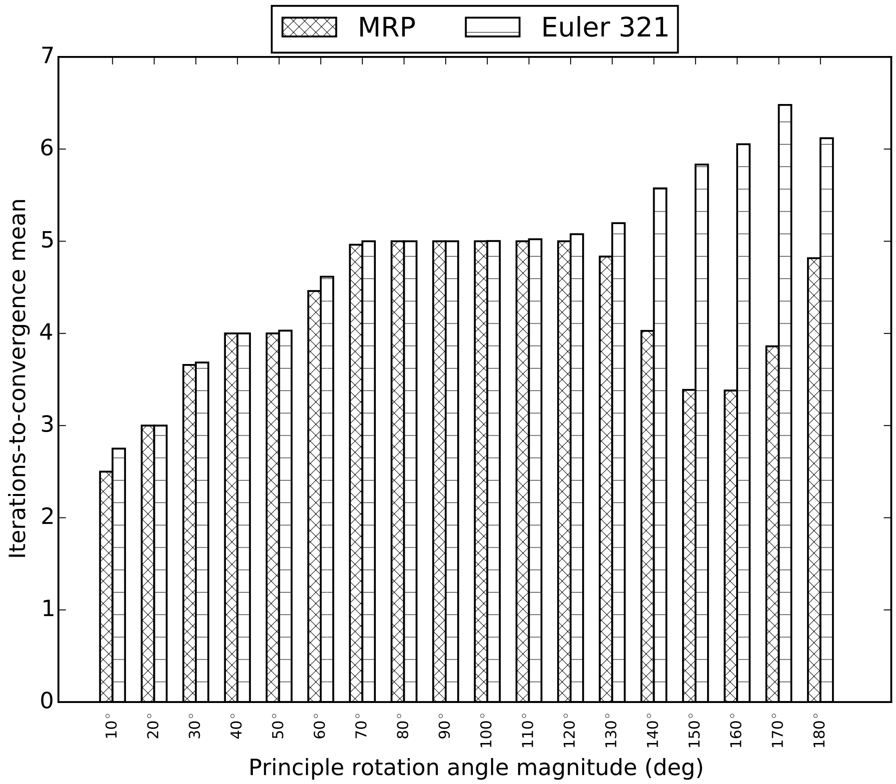


Figure 3.8: Mean iterations-to-convergence against rotational transform magnitude for the multiplicative ICP($N = 5000$)

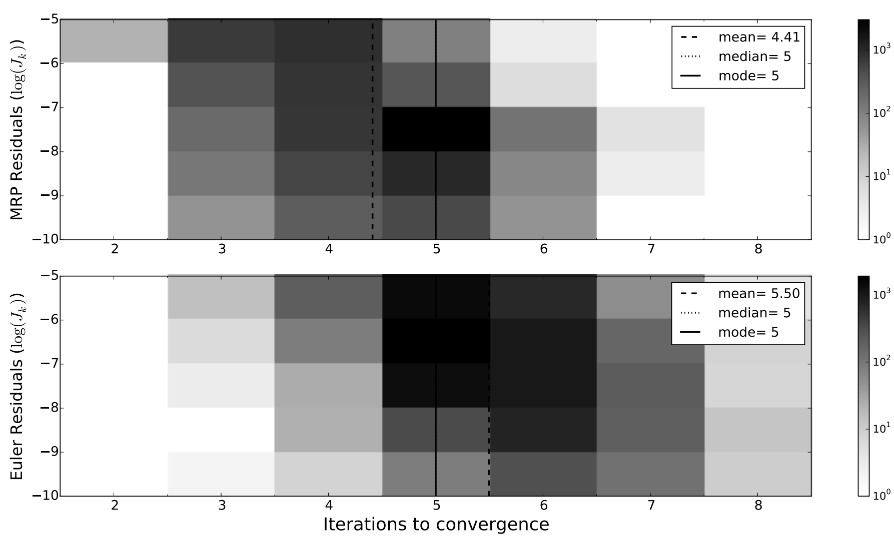


Figure 3.9: Iterations/residuals map for the multiplicative ICP ($N = 5000$)

Table 3.1: Iteration-to-convergence statistics for the linearized MRP multiplicative ICP

Point cloud size	100	500	1000	1750	2500	3250	5000
Mean (iteration)	4.68	4.45	4.42	4.42	4.42	4.41	4.41
Standard deviation (iteration)	0.65	0.76	0.76	0.78	0.78	0.79	0.78

Table 3.2: Iteration-to-convergence statistics for the linearized Euler-angles multiplicative ICP

Point cloud size	100	500	1000	1750	2500	3250	5000
Mean (iteration)	5.85	5.53	5.57	5.51	5.5	5.51	5.5
Standard deviation (iteration)	0.97	0.73	0.77	0.72	0.72	0.71	0.71

3.2.6.2 Realistic data sets with imperfect point-pair matching

Two cases illustrating realistic test scenarios for the ICP2P algorithm, featuring a deputy satellite and a derelict first stage representative of space debris were also simulated, using the realistic point-pair matching scheme described in this paper. The underlying topology of the point clouds is paramount to the convergence properties of the ICP2P algorithm[85]. Indeed, the basin of convergence of the full ICP algorithm using a point-pair matching scheme is more limited than that of the idealized ICP using perfect point-pairs. It is thus expected that the distinctive symmetries and surface features of both test cases will result in different convergence results.

3.2.6.3 Case 1: Satellite servicing scenario

A noise-free structured point cloud representing a 3D range image was generated by applying a Lidar ray-casting algorithm to a simple spacecraft shape model shown on Figure 3.10. This corresponds to a satellite servicing mission where a deputy satellite is approached by a servicer equipped with a Lidar instrument. The relative orientation between the imaged spacecraft and the Lidar yielded a destination point cloud comprised of 1573 points shown on Figure 3.11. Gaussian noise of zero mean and standard deviation $\sigma = 5$ cm was added to each range measurement.

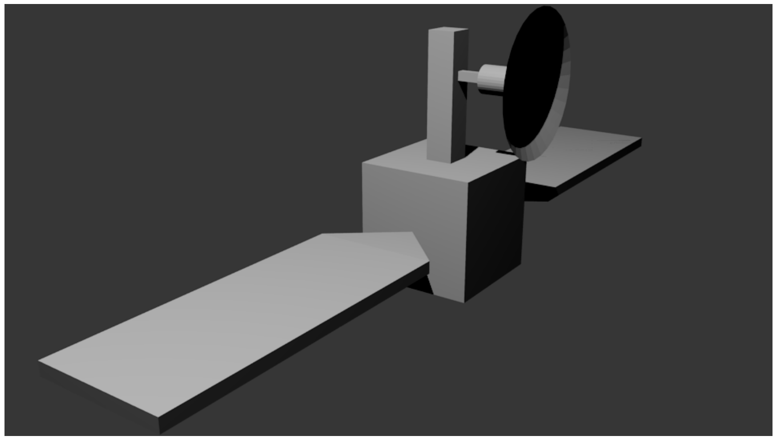


Figure 3.10: Rendered image of the spacecraft shape model

Figure 3.11: Generated noisy destination point cloud

3.2.6.4 Case 2: Orbital debris retrieval

This scenario illustrate an orbital debris-retrieval mission featuring a derelict first stage. A rendered image of the first stage is provided on Figure 3.12. The noise-free destination point cloud shown on Figure 3.13 was comprised of 2206 points. Note that the point cloud was only representing a portion of the rocket stage. Gaussian noise of zero mean and standard deviation $\sigma = 5$ cm was also added to each range measurement.

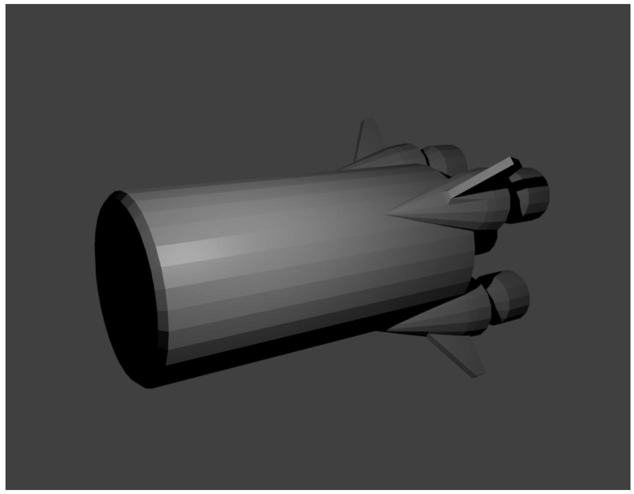


Figure 3.12: Rendered image of the shape model of the derelict rocket first stage

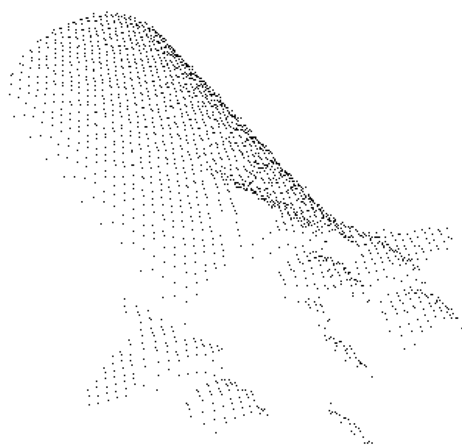


Figure 3.13: Generated noisy destination point cloud

3.2.6.5 Performance

Both tests cases were fed to a Monte-Carlo simulation running 1000 times. This time, the rotational component was formulated in terms of a sequence of 321 Euler angles uniformly sampled in $\left[0, \frac{2}{4}\pi\right] \times \left[-\frac{\pi}{8}, \frac{\pi}{8}\right] \times \left[0, \frac{2}{4}\pi\right]$ radians respectively, which is a fourth of the nominal angle intervals over which 321 Euler angles are defined. The translational component of the rigid transform was still sampled from $[-0.1, 0.1]$ meters. The justification for the smaller rotation interval is motivated by the limited radius of convergence of the pair-matching procedure. Letting the angles vary over their full domain would cause all methods to fail most of the time, because the corresponding rigid transform is outside of the basin of convergence of the pair-matching algorithm. Restricting the angles to smaller values thus ensures that the convergence basin of the pair-matching method is sufficiently populated. 1000 random rigid transforms were generated this way and estimated by means of both algorithms. Zero-mean Gaussian noise was added along the line-of-sight direction of each pixel. The convergence threshold was set to $J_c = 2.5$ cm and $J_c = 7$ cm for Case 1 (satellite) and Case 2 (rocket body) respectively. These values of J_c correspond to typical residuals when the correct rigid transform is achieved, and are depending upon the dimensions of the point clouds. Convergence was achieved when the RMS residuals J were less than J_c after 40 iterations. The source and the destination point cloud were always comprised of the same size. Their point content was representative of the same area of the targeted object, but the point clouds were not exactly overlapping due to the presence of noise.

The statistics summarizing the performance of all methods are presented in Table 3.3 and Table 3.4. It can be seen in both cases that a majority of the 1000 Monte Carlo outcomes did not converge. This was expected since the convergence basin of the closest-compatible normal pair matching algorithm puts a higher bound on the rigid transform rotation amplitude that can be captured.

Case 1 denotes a slightly better performance for the MRP parametrization. The results of Table 3.3 are tied to the topology of the point cloud, that is mostly comprised of flat areas of nearly uniform normal directions with the exception of the parabola. This geometry leads to a great number of incorrect pair-matching that cause the ICP to not converge, in addition to a number of local minima of the ICP2P cost function that cause the algorithm to get stuck. Figure 3.14 show that MRP remain faster than Euler angles, but their advantage is much less than what was obtained with the perfect point-pairing. This is another consequence of the point cloud geometry and the small radius of convergence of the point-pair matching scheme. The restricted angular span over which ICP can converge and thus prevents MRP from outperforming Euler angles when dealing with large rigid transforms.

On the other hand, Case 2 has a higher convergence count for all parametrization/formulation combinations, as seen on Table 3.4. The topology of the point cloud thus played a major role in the convergence properties, as announced. In addition, MRP did converge a minimum of 17 outcomes more than Euler angles. The basin of convergence of the pair-matching algorithm is larger in this case, which enables the MRP ICP to differentiate itself from the Euler angles-based one due to its better performance when large rotations are involved. Figure 3.15 denote a more significant speed improvement using MRP compared in Case 1, which makes perfect sense given the more-favorable point cloud geometry. The more varied distribution of surface normals across the point clouds help constraining the parameter space, granting the ICP2P with realistic point-pair matching a better performance.

Table 3.3: Number of converged outcomes for multiplicative ICP formulations, Case 1

Euler	MRP
287	293

Table 3.4: Number of converged outcomes for multiplicative ICP formulations, Case 2

Euler	MRP
513	530

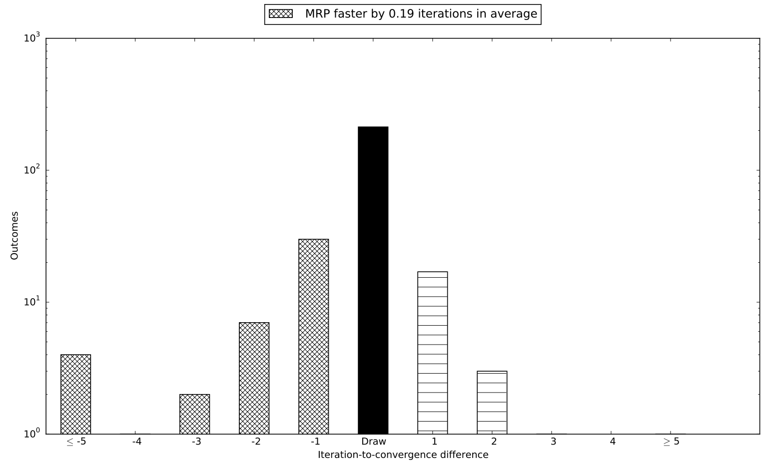


Figure 3.14: Iteration-to-convergence difference distribution for the multiplicative ICP (Case 1)

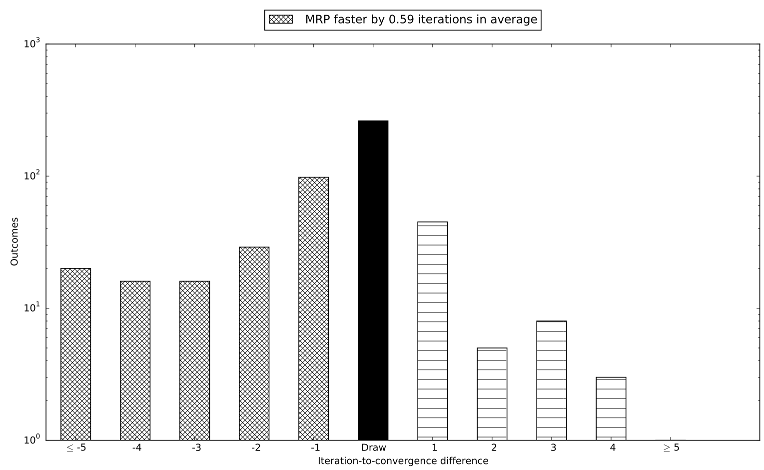


Figure 3.15: Iteration-to-convergence difference distribution for the multiplicative ICP (Case 2)

The results of the Monte-Carlo simulation using perfect point-pairs establish the faster performance of MRP compared to Euler angles. For the $N = 5000$ point cloud, the Monte-Carlo results suggest a gain of at least one or two iteration for the MRP-based algorithm compared to Euler angles. The mean iteration-to-convergence count was 5.5 for the multiplicative Euler-angle based ICP. On the other hand, the mean iteration-to-convergence count was 4.41 for the multiplicative MRP-based ICP. The performance of both methods becomes geometry-dependent when a realistic point-pair matching is used. Nevertheless, it appears that MRP do better than Euler angles, in a degree that varies with the topology of the point cloud being processed.

Table 3.5: Angular error for converged MC outcomes, multiplicative ICP (Case 1)

	Minimum error (°)	Maximum error (°)
MRP	0.14	3.29 (without outlier) / 163.23 (with outlier)
Euler angles	0.16	3.2

Table 3.6: Angular error for converged MC outcomes, multiplicative ICP (Case 2)

	Minimum error (°)	Maximum error (°)
MRP	0.018	1.01 (without outlier) / 99 (with outlier)
Euler angles	0.016	1.17

One can ensure that convergence of the ICP2P in the residuals sense implies that the rigid transform has been properly resolved. Table 3.5 and 3.6 provides the minimum and maximum value of the principal rotation angle measuring the error between the prescribed rotation and its estimate in both test cases, for converged MC outcomes. The multiplicative ICP results feature an abnormal higher-bound on the principal angle error for MRP. It appeared that the multiplicative MRP-based ICP converged only once to a spurious orientation despite satisfying residuals. This can be explained by the rejection of too many point pairs that left the ICP2P with an insufficient number of observations to process. This way, the residuals were computed over a number of observations that was too small to actually constrain the attitude state. These spurious convergent cases were obtained for relatively large rigid transform, for which the Euler angles-based ICP actually did not converge. Not accounting for this outlier in the error bounds brings the maximum error angle to a few degrees at most, confirming that the ICP2P had converged to the global minimum in the rest of the converged outcomes.

As demonstrated, the convergence of ICP2P is only guaranteed in the vicinity of a local minimum. If this local minimum happens to be the global minimum of the cost function, then the algorithm will have converged to the correct rigid transform. If not, the rigid transform that is returned by the algorithm does not properly align the point clouds. The latter may happen in the presence of symmetries and large rotations between the two point clouds to register.

This issue manifests itself for both the MRP and the Euler-angles based methods when the realistic point-pair matching scheme is used. That is, the incorrect matching of points between the source and the destination point cloud drives a number of MC outcomes to a local minimum of the ICP2P cost function, or deteriorates the convergence speed so that the global maximum is not reached within the imparted 40 iterations. This is major drawback of ICP2P compared to other registration frameworks [89].

In conclusion, the proposed MRP-based ICP alignment algorithm was proven to outperform the legacy Euler-angles based method when applied to structured points clouds. Perfect point-pair matching scenarios were explored and confirmed the better performance of the MRP algorithm in terms of convergence speed and accuracy. More generally, the MRP-based ICP was shown to converge faster than its counterpart especially as the magnitude of the rotational transform to estimate was large. MRP retain their advantage over Euler angles when coupled with a realistic point-pair matching algorithm, although the convergence basin of the ICP algorithm is then tied to the point cloud geometry.

3.3 Bundle adjustment

Successive registration of 3D points acquired by a Lidar system can lead to an erroneous structure in the reconstructed scene. If successive point-clouds are registered in a fixed stitching frame, the successive registration error will grow at least linearly with the number of point-clouds. Bundle adjustment is a powerful corrective measure that leverages overlap of non-successive data to remove such creeping misalignment errors [90]. Originating from optical camera image processing, this method roughly consists in solving for camera intrinsic parameters and relative motion minimizing an overall cost function accounting for reprojection errors. In our case where observations are 3D point-clouds, the bundle-adjustment function takes the form

$$J^2 = \sum_{k=1}^{M} \sum_{i=1}^{N_k} \left[\hat{n}_i^T M_{D_k}^T \left(M_{S_k} \mathbf{S}_{i,k} + \mathbf{x}_{S_k} - M_{D_k} \mathbf{D}_{i,D_k} - \mathbf{x}_{D_k} \right) \right]^2 \tag{3.21}$$

where the rigid transforms $(M_{S_k}, \mathbf{x}_{S_k})$ and $(M_{D_k}, \mathbf{x}_{D_k})$ define the registration of point clouds S_k and D_k relative to a reference point cloud.

3.3.1 Rigid-transforms estimation

For the problem to be well-posed, one must define a point cloud that serves as an absolute reference. That is, if a total of Q point-clouds are considered, there will be only $Q-1$ rigid transforms to determine. This becomes obvious if only two point-clouds are present: bundle adjustment then reduces to a standard ICP problem, where only one rigid transform needs to be determined. Introducing $\epsilon_{k,i}$

$$\epsilon_{k,i} = \hat{n}_i^T M_{D_k}^T \left(M_{S_k} \mathbf{S}_{i,S_k} + \mathbf{x}_{S_k} - M_{D_k} \mathbf{D}_{i,D_k} - \mathbf{x}_{D_k} \right) \tag{3.22}$$

one must linearize the quantities M_{S_k}, $\mathbf{x}_{S_k}$, M_{D_k} and $\mathbf{x}_{D_k}$ because of the coupling between rotation and translation. Defining

- $\mathcal{L}_0$: the stitching frame defined by the reference point cloud
- $\mathcal{S}_{S_k}$: the frame in which the coordinates of the S_k-th point-cloud are defined post-correction
- $\bar{\mathcal{S}}_{S_k}$: the frame in which the coordinates of the S_k-th point-cloud are defined before the bundle-adjustment correction is applied
- $\mathcal{D}_{D_k}$: the frame in which the coordinates of the D_k-th point-cloud are defined post-correction
- $\bar{\mathcal{D}}_{D_k}$: the frame in which the coordinates of the D_k-th point-cloud are defined before the bundle-adjustment correction is applied

The DCMs can be rewritten

$$M_{S_k} = [\mathcal{L}_0 \mathcal{S}_{S_k}] \tag{3.23}$$

$$M_{D_k} = [\mathcal{L}_0 \mathcal{D}_{D_k}] \tag{3.24}$$

$\epsilon_{k,i}$ can be rewritten as

$$\begin{aligned}\epsilon_{k,i} &= \hat{n}_i^T [\mathcal{D}_{D_k} \mathcal{L}_0] \left([\mathcal{L}_0 \mathcal{S}_{S_k}] \mathbf{S}_{i,S_k} + \mathbf{x}_{S_k} - [\mathcal{L}_0 \mathcal{D}_{D_k}] \mathbf{D}_{i,D_k} - \mathbf{x}_{D_k} \right) &(3.25)\\ &= \hat{n}_i^T [\mathcal{D}_{D_k} \bar{\mathcal{D}}_{D_k}][\bar{\mathcal{D}}_{D_k} \mathcal{L}_0] \left([\mathcal{L}_0 \bar{\mathcal{S}}_{S_k}][\bar{\mathcal{S}} \mathcal{S}_{S_k}] \mathbf{S}_{i,S_k} + \mathbf{x}_{S_k} - [\mathcal{L}_0 \bar{\mathcal{D}}_{D_k}][\bar{\mathcal{D}} \mathcal{D}_{D_k}] \mathbf{D}_{i,D_k} - \mathbf{x}_{D_k} \right) &(3.26)\end{aligned}$$

Under the assumption that the corrective rigid transform is small, this becomes

$$[\mathcal{S}_{S_k} \bar{\mathcal{S}}_{S_k}] \simeq I_3 - 4[\tilde{\boldsymbol{\sigma}}_{S_k}] \tag{3.27}$$

$$[\mathcal{D}_{D_k} \bar{\mathcal{D}}_{D_k}] \simeq I_3 - 4[\tilde{\boldsymbol{\sigma}}_{D_k}] \tag{3.28}$$

$$\mathbf{x}_{S_k} = \bar{\mathbf{x}}_{S_k} + \delta \mathbf{x}_{S_k} \tag{3.29}$$

$$\mathbf{x}_{D_k} = \bar{\mathbf{x}}_{D_k} + \delta \mathbf{x}_{D_k} \tag{3.30}$$

so

$$\epsilon_{k,i} \simeq \hat{n}_i^T \left(I_3 - 4[\tilde{\boldsymbol{\sigma}}_{D_k}] \right) [\bar{\mathcal{D}}_{D_k} \mathcal{L}_0] \left([\mathcal{L}_0 \bar{\mathcal{S}}_{S_k}] \left(I_3 + 4[\tilde{\boldsymbol{\sigma}}_{S_k}] \right) \mathbf{S}_{i,S_k} + \mathbf{x}_{S_k} - [\mathcal{L}_0 \bar{\mathcal{D}}_{D_k}] \left(I_3 + 4[\tilde{\boldsymbol{\sigma}}_{D_k}] \right) \mathbf{D}_{i,D_k} - \mathbf{x}_{D_k} \right) \tag{3.31}$$

Expanding,

$$\begin{aligned}\epsilon_{k,i} \simeq{}& \hat{n}_i^T [\bar{\mathcal{D}}_{D_k} \mathcal{L}_0] \left([\mathcal{L}_0 \bar{\mathcal{S}}_{S_k}] \mathbf{S}_{i,S_k} + \bar{\mathbf{x}}_{S_k} - [\mathcal{L}_0 \bar{\mathcal{D}}_{D_k}] \mathbf{D}_{i,D_k} - \bar{\mathbf{x}}_{D_k} \right) \\ &+ \hat{n}_i^T [\bar{\mathcal{D}}_{D_k} \mathcal{L}_0] \left(4 \left[[\mathcal{L}_0 \bar{\mathcal{D}}_{D_k}][\widetilde{\mathbf{D}_{i,D_k}}] \boldsymbol{\sigma}_{D_k} - [\mathcal{L}_0 \bar{\mathcal{S}}_{S_k}][\widetilde{\mathbf{S}_{i,S_k}}] \boldsymbol{\sigma}_{S_k} \right] + \delta \mathbf{x}_{S_k} - \delta \mathbf{x}_{D_k} \right) \\ &- 4 \hat{n}_i^T [\tilde{\boldsymbol{\sigma}}_{D_k}][\bar{\mathcal{D}}_{D_k} \mathcal{L}_0] \left([\mathcal{L}_0 \bar{\mathcal{S}}_{S_k}] \mathbf{S}_{i,S_k} + \bar{\mathbf{x}}_{S_k} - [\mathcal{L}_0 \bar{\mathcal{D}}_{D_k}] \mathbf{D}_{i,D_k} - \bar{\mathbf{x}}_{D_k} \right) \end{aligned} \tag{3.32}$$

Defining a substate vector specific to $\{S_k\}$ and $\{D_k\}$

$$\delta \mathbf{X}_{S_k D_k} = \begin{pmatrix} \delta \mathbf{x}_{S_k} \\ \boldsymbol{\sigma}_{S_k} \\ \delta \mathbf{x}_{D_k} \\ \boldsymbol{\sigma}_{D_k} \end{pmatrix} \tag{3.33}$$

Such that $\epsilon_{k,i}$ becomes

$$\epsilon_{k,i} = y_{k,i} - H_{k,i}\delta\mathbf{X}_{S_k D_k} \tag{3.34}$$

with

$$y_{k,i} = \hat{n}_i^T[\bar{\mathcal{D}}_{D_k}\mathcal{L}_0]\left([\mathcal{L}_0\bar{\mathcal{S}}_{S_k}]\mathbf{S}_{i,S_k} + \bar{\mathbf{x}}_{S_k} - [\mathcal{L}_0\bar{\mathcal{D}}_{D_k}]\mathbf{D}_{i,D_k} - \bar{\mathbf{x}}_{D_k}\right) \tag{3.35}$$

$$H_{k,i} = -\begin{bmatrix} \left(\hat{n}_i^T[\bar{\mathcal{D}}_{D_k}\mathcal{L}_0]\right)^T \\ \left(-4\hat{n}_i^T[\bar{\mathcal{D}}_{D_k}\bar{\mathcal{S}}_{S_k}][\widetilde{\mathbf{S}_{i,S_k}}]\right)^T \\ \left(-\hat{n}_i^T[\bar{\mathcal{D}}_{D_k}\mathcal{L}_0]\right)^T \\ \left(4\hat{n}_i^T[\widetilde{\mathbf{D}_{i,D_k}}] - 4\left([\mathcal{L}_0\bar{\mathcal{S}}_{S_k}]\mathbf{S}_{i,S_k} + \bar{\mathbf{x}}_{S_k} - [\mathcal{L}_0\bar{\mathcal{D}}_{D_k}]\mathbf{D}_{i,D_k} - \bar{\mathbf{x}}_{D_k}\right)^T[\mathcal{L}_0\bar{\mathcal{D}}_{D_k}][\tilde{\hat{n}}_i]\right)^T \end{bmatrix}^T \tag{3.36}$$

The full state deviation over which the bundle-adjuster operates is comprised of $Q-1$ rigid transforms indexed from 1 to $Q-1$, where the rigid transform indexed at 0 is the reference one characterized by $\boldsymbol{\sigma}_0 = \mathbf{0}$ and $\mathbf{x}_0 = \mathbf{0}$ and thus left out of the bundle-adjuster.

$$\delta\mathbf{X} = \begin{bmatrix} \delta\mathbf{x}_1 \\ \boldsymbol{\sigma}_1 \\ \delta\mathbf{x}_2 \\ \boldsymbol{\sigma}_2 \\ \vdots \\ \delta\mathbf{x}_{Q-1} \\ \boldsymbol{\sigma}_{Q-1} \end{bmatrix} \tag{3.37}$$

Therefore, the final expression of $\epsilon_{k,i}$ is

$$\epsilon_{k,i} = y_{k,i} - H_{k,i}\mathcal{I}_k\delta\mathbf{X} \tag{3.38}$$

where $\mathcal{I}_k$ is a mapping matrix associating the global indices of the rigid transforms to the ones showing in the k-th point cloud pair. If either S_k or D_k correspond to the index of the reference point cloud, then the corresponding rigid transform is removed from $\delta\mathbf{X}_{S_k D_k}$ so that it only features the deviation in the rigid transform being solved for.

Going back to the cost function,

$$J^2 = \sum_{k=1}^{M} \sum_{i=1}^{N_k} \epsilon_{i,k}^2 \tag{3.39}$$

$$= \sum_{k=1}^{M} \sum_{i=1}^{N_k} (y_{k,i} - H_{k,i} \mathcal{I}_k \delta \mathbf{X})^2 \tag{3.40}$$

J^2 is minimized by the least-squares deviation, solution to

$$\frac{\partial J^2}{\partial \delta \mathbf{X}} = \mathbf{0}^T \tag{3.41}$$

which yields

$$\left(\sum_{k=1}^{M} \mathcal{I}_k^T \Lambda_k \mathcal{I}_k \right) \delta \mathbf{X} = \sum_{k=1}^{M} \mathcal{I}_k \mathbf{N}_k \tag{3.42}$$

where

$$\Lambda_k = \sum_{i=1}^{N_k} H_{k,i}^T H_{k,i} \tag{3.43}$$

$$\mathbf{N}_k = \sum_{i=1}^{N_k} H_{k,i}^T y_{k,i} \tag{3.44}$$

3.3.2 Loop closure

For the matrix on the left-hand side of Equation (3.42) to be invertible, each of the featured point-clouds must be paired with or indirectly constrained by the reference point cloud. Figure 3.16 illustrates this minimum point-cloud pairing required. However, the power of bundle adjustment best manifests itself when the reference point cloud is directly paired with a non-consecutive point-cloud. When such a loop-closure is present, the point cloud sequence ranging from the reference to the closure one is fully constrained in the reprojection error cost function, enabling one to remove substantial misalignment errors that may have accumulated over time, as shown at the bottom of Figure 3.16.

Loop closure can be detected by asserting the degree of geometric overlap between two point-clouds. Overlap can be quantified by computing tentative ICP point-pairs amongst a source/destination point cloud pair, as the number of point pairs cannot be superior to the number of points in the source point cloud [84]. Point-pairs are rejected if their intrisic ICP2P error $\hat{n}_i^T(\mathbf{S}_i - \mathbf{D}_i)$ exceeds a one-standard deviation variation from the mean of the intrisic error distribution across all pairs. If more than 80% of the tentative point-pairs are retained, then the point-clouds are considered as overlapping and paired in the bundle adjustment process. Note that there is no need to consider the entirety of the point clouds to rule out overlaps, as comparing the instrument line-of-sight directions in the estimated body frame is a quick and sound check to dismiss point clouds that are too far apart to justify a more in-depth investigation of their possible overlaps. In addition, only considering a random subset of the points in each point is usually sufficient to get an reliable overlap estimate.

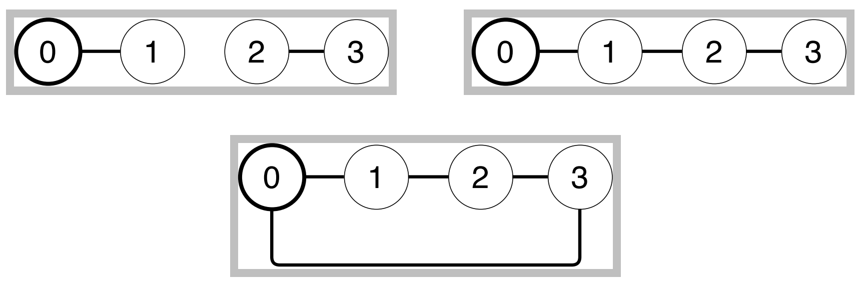


Figure 3.16: Connectivity graph of 4 points-clouds, with edges denoting considered point-cloud pairs in the bundle adjustment. Left: the problem is ill-posed because 2 and 3 are not constrained by 0. Right: the problem is well posed but will not significantly improve the ICP solution. Bottom: $(0, 3)$ provides loop closure and constrains all point-clouds, improving the registration solution

3.3.3 Clustering

Tracking point-clouds overlaps with the scheme described in the previous section unsurprisingly leads to a quick growth of the overlap graph. This growth can be partially mitigated by the addition of a clustering scheme to the bundle adjuster. The role of this clustering scheme is to enable the bundle adjuster to identify "clusters" of point-cloud pairs that bring redundant information. Consider point-cloud $i = 0$ and the point-clouds that may be overlapping with it: i_1, i_2

,..., i_P. A cluster is a grouping of the point-clouds corresponding with i such that the indices in a given cluster are within d of each other. For instance, from $P = 10$ points clouds matching with i and indexed as $i_1 = 5$, $i_2 = 8$, $i_3 = 59$, $i_4 = 61$, $i_5 = 63$, $i_6 = 64$, $i_7 = 65$, $i_8 = 66$, $i_9 = 119$, $i_{10} = 120$, setting the cluster size to $d = 4$ leaves with 4 clusters:

- Cluster 1: 5,8
- Cluster 2: 59,61,63
- Cluster 3: 64,65,66
- Cluster 4: 119,120

Since point-clouds within the same cluster as essentially redundant, only one point-cloud per cluster is retained and paired with the i-th point-cloud. For the example above, it means that only the $0-5$, $0-59$, $0-64$ and $0-119$ edges will be kept in the graph. The effect of this clustering scheme on the overlap graph growth is shown on Figure 3.17, where a simple point-cloud acquisition scenario was run without & with clustering. Setting $d = 4$ roughly divides the edge count by 4, for an equivalent alignment quality in the registered point clouds.

3.3.4 Local structures

A well known issue in SLAM is the loss of tractability should too many observations and states be considered altogether in the graph. In the case of point cloud registration, an individual observation is no else but a point cloud comprised of tens of thousands of 3D points and associated normals. When considering the overlap graph possibly keeping track of hundreds of worthy point-cloud overlaps to leverage in the bundle adjustment, there is clearly a point where the data size and computational demand will become too much of a burden for the framework to keep operating. One possible way to alleviate this hurdle is to only perform bundle adjustment on a subset of the full graph [91]. Another option more akin to filtering is to "bake-in" bundle-adjusted rigid transforms, effectively fusing together the considered point-clouds [92]. If one can guarantee good

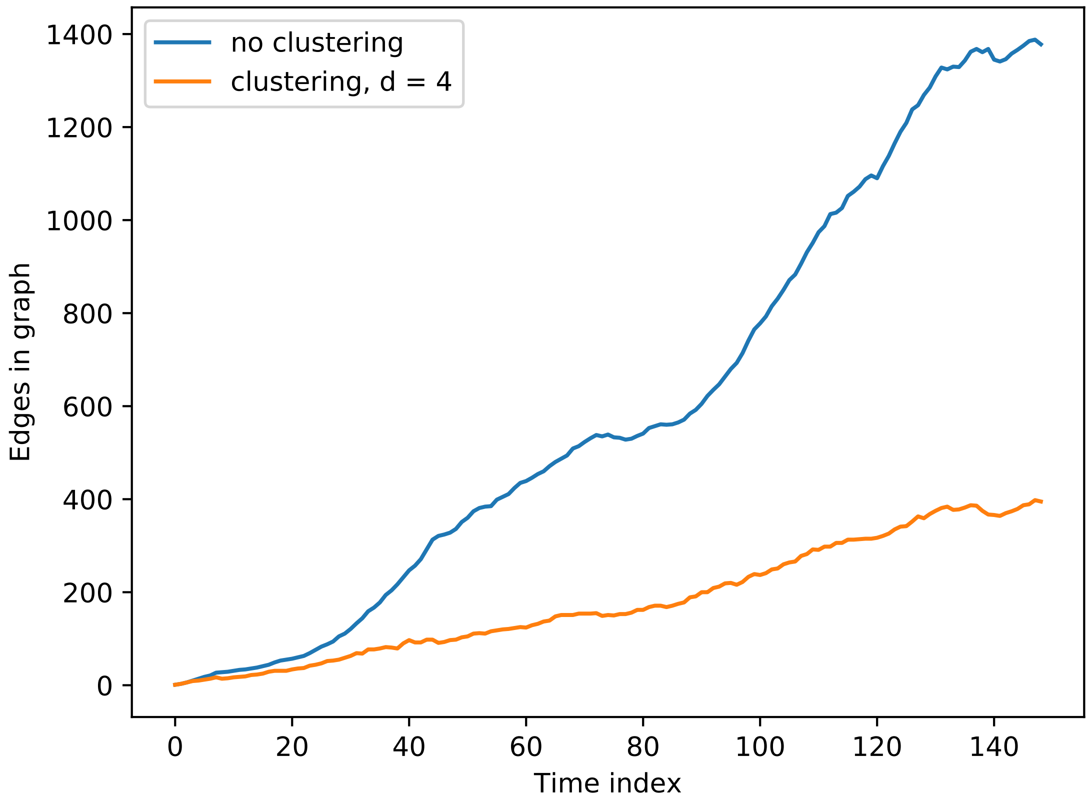


Figure 3.17: Effect of clustering on overlap graph growth

performance of the local bundle adjustment providing the rigid transforms used to align the to-be-fused point clouds, the SLAM tractability can be retained. This can be achieved by **a**) detecting loop closures over more than d point clouds (where d still refers to the cluster size defined in the previous paragraph), **b**) identifying inlier and outliers in the overlap graph, **c**) bundle-adjusting the point-clouds over the inlier overlaps and **d**) assessing when the bundle adjustment has converged.

a) has been described in 3.3.2, while **c**) is no else but the process highlighted in 3.3.1. **b**) and **d**) require a robust numerical scheme enabling the detection and flagging of "bad" edges, that is to say edges (e.g point-cloud pairs) that were considered in the bundle-adjustment but have not reached satisfying alignment. A robust, autonomous outlier detection scheme relying on Gaussian Mixtures clustering of the point-cloud pairs residuals was found to be the quickest and reliable way to carry out this task. A Gaussian mixture probability density function comprised of Z mixands describing a n-dimensional continuous variable $\mathbf{x}$is defined as

$$p(\mathbf{x}) \equiv \sum_{z=1}^{Z} \omega_i \mathcal{N}(\mathbf{x}|\mathbf{m}_i, P_i) \tag{3.45}$$

where $\mathcal{N}(\mathbf{x}|\mathbf{m}_i, P_i)$ denotes the n-dimensional Gaussian distribution of mean $\mathbf{m}_i$ and covariance P_i. The weights $\{\omega_1, \ldots, \omega_Z\}$ must add up to one for p to be a valid probability density function. Gaussian Mixtures are versatile in the sense that they can be used to fit virtually arbitrary populations through maximum-likelihood approaches such as the Expectation Maximization (EM) algorithm [93]. The heuristic the GMM clustering scheme is based on is two fold: first, it is assumed that the point-cloud pairs (the edges) are in majority satisfactory (inliers are more prevalent than outliers in the point-cloud-to-point-cloud BA residuals). Second, the probability density that is being fit must have its support within $\mathbb{R}^+$, since the distance residuals are positive numbers.

The pseudo-code describing the functioning of the outlier rejection scheme along with the bundle adjustment is detailed in Algorithm 1. Once the bundle adjustment has been run for a fixed number of iterations $N_{\text{iter}_{BA}}$ over the Q_k point clouds $\{P_1^k, \ldots, P_{Q_k}^k\}$ linked through the connectivity graph $\{\mathbf{e}_1, \ldots \mathbf{e}_{N_{e_k}}\}$, a collection of point-cloud to point-cloud residuals $\{\epsilon_1^k, \ldots, \epsilon_{N_{e_k}}^k\}$ becomes available along with the updated point-clouds themselves. These residuals are then clustered by means of an increasing number of Gaussian Mixtures, using Armadillo's *arma::gmm_diag* class [94]. If the mean of each cluster minus three standard deviations in greater than 0 (hence the Gaussian Mixture PDF has its support within $\mathbb{R}^+$), then the current Gaussian Mixture comprised of m mixands is a minimum, sufficient clustering of the point-cloud-to-point-cloud distance residuals. If not, more mixtures are added until the above condition is satisfied. If greater than the user-provided tolerance *tol*, the mean corresponding to the most-populated cluster is used to define a cutoff value for the other mixand means. Setting *tol* to a reasonable value, such as the the standard deviation in the range measurements prevents the outlier detection scheme from being overly pessimistic when assessing the quality of the bundle-adjustment. The edges belonging to clusters whose means are greater than the cutoff value are finally removed from the connectivity graph. Should the returned value of the boolean $SpuriousEdgesFound$ evaluate to True, the bundling of the Q_k point clouds into a new local structure will be triggered if Q_k is sufficiently large and the anchor point cloud P_1^k connected to the last point cloud in the sequence $P_{Q_k}^k$ either directly or through a neighboring point cloud less than d point clouds away from P_1^k.

Algorithm 1 Bundle-adjustment and Gaussian-Mixture based edge residuals clustering

procedure BundleAdjustementEdgeGMMClustering

Initialization:

  Given: Q_k point clouds $\{P_1^k, \ldots, P_{Q_k}^k\}$, connectivity graph $\{\mathbf{e}_1, \ldots \mathbf{e}_{N_{e_k}}\}$, $N_{\text{iter}_{BA}}$, tol

  $SpuriousEdgesFound \leftarrow$ False

Run Bundle Adjustment:

  $\{P_1^k, \ldots, P_{Q_k}^k\}, \{\epsilon_1^k, \ldots, \epsilon_{N_{e_k}}^k\} \leftarrow \text{BundleAdjust}\left(\{P_1^k, \ldots, P_{Q_k}^k\}, \{\mathbf{e}_1, \ldots \mathbf{e}_{N_{e_k}}\}, N_{BA}\right)$

Cluster Bundle Adjustment residuals :

  for m in $[\![1 \ldots Q_k - 1]\!]$ **do**

    $(\{\mu_1, \sigma_1\}, \ldots, \{\mu_m, \sigma_m\}), \{\mathcal{C}_{\mathbf{e}_1}, \ldots, \mathcal{C}_{\mathbf{e}_{N_{e_k}}}\} \leftarrow \text{TrainGMMModel}(\{\epsilon_1^k, \ldots, \epsilon_{Q_k}^k\})$

Check whether more mixtures are necessary :

    $badClusters \leftarrow \{\}$

    for c in $[\![1 \ldots m]\!]$ **do**

      if $\mu_i - 3\sigma_i < 0$ **then**

        $badClusters \leftarrow badClusters + \{i\}$

    if $\text{Size}\,(badClusters) == 0$ **or** $m == Q_k - 1$ **then**

      break

Compute Cutoff:

  $biggestCluster \leftarrow \max\left(\text{HistogramBinPopulation}\left(\{\mathcal{C}_{\mathbf{e}_1}, \ldots, \mathcal{C}_{\mathbf{e}_{N_{e_k}}}\}\right)\right)$

  $\text{cutoff} \leftarrow \max\left(tol, \mu_{BiggestCluster}\right)$

Remove spurious edges from the graph:

  for i in $[\![1 \ldots N_{e_k}]\!]$ **do**

    if $\mu_{\mathcal{C}_{\mathbf{e}_i}} > \text{cutoff}$ **then**

      $G_k.\text{Remove}\,(\mathbf{e}_i)$

      $SpuriousEdgesFound \leftarrow$ True

  return $\{P_1^k, \ldots, P_{Q_k}^k\}$, $\{\mathbf{e}_1, \ldots \mathbf{e}_{N'_{e_k}}\}$, $SpuriousEdgesFound$

The functioning of Algorithm 1 is illustrated on Figure 3.18. 100 simulated residuals were drawn from an underlying Gaussian Mixture distribution of means $(0.25, 0.5, 0.9)$, standard deviations $(0.1, 0.2, 0.2)$ and weights $(0.28571429, 0.57142857, 0.14285714)$. The highest weight corresponds to the inlier mixand of mean 0.5. The mixand of highest mean and lowest weight denotes outliers. It can be seen that a single ($Z = 1$) Gaussian cannot be used to identify and reject the outliers drawn from the last mixture, since the support of this mixture encroaches below 0. Satisfying clustering is detected from $Z = 3$ onwards, allowing the different clusters in the underlying data to be identified.

Figure 3.19 shows the evolution in the number of the considered point clouds Q at every time step against the total number of point-clouds effectively collected for a sample run around asteroid Itokawa. It can be seen that the number of considered point clouds and graph connectivity always remains tractable, with a local structure roughly created every d point clouds. This is not always the case, since one local structure took up to 19 point clouds to close. The structure ultimately closed when point cloud 183 was found to overlap with point cloud 169, the latter being within d point clouds of point cloud 165, the anchor point cloud for this local structure. Because the bundle-adjustment did not detect any spurious edge after 5 iterations, the local structure was flagged as complete, causing a new local structure to be formed.

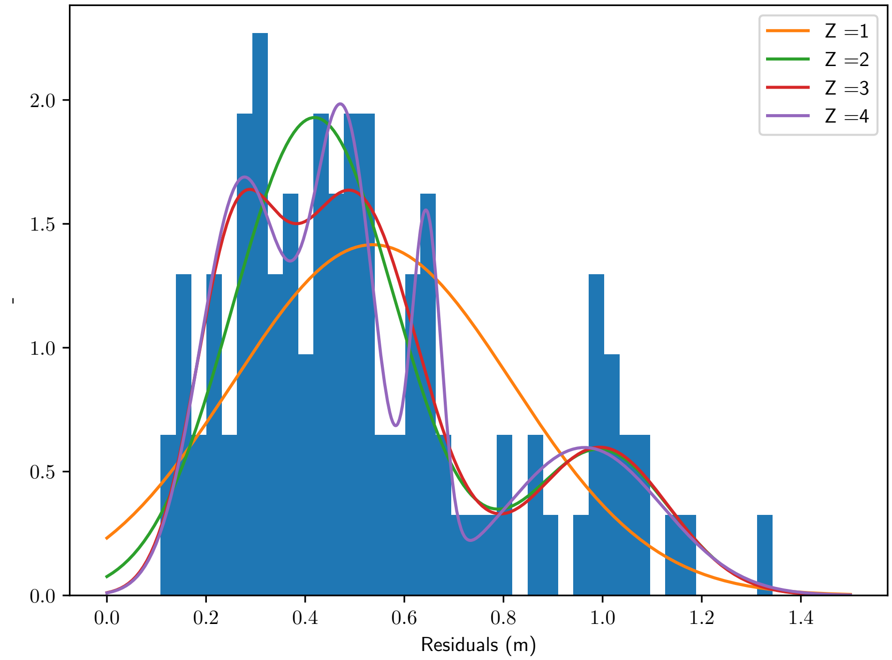


Figure 3.18: Gaussian-Mixture clustering of the simulated connectivity graph residuals

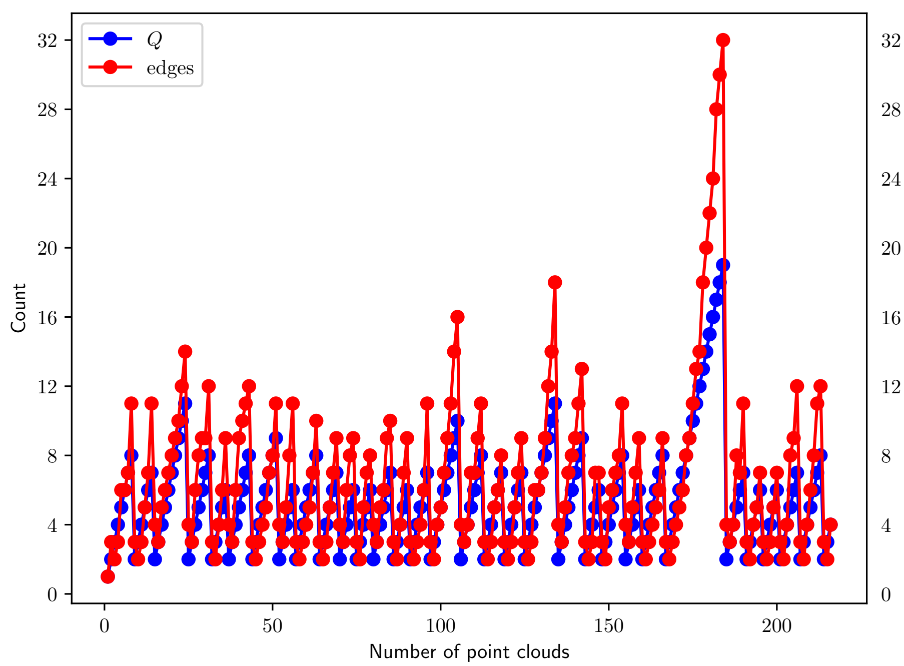


Figure 3.19: Number of considered point clouds Q and associated graph connectivity in the considered structure

Chapter 4

Shape reconstruction

4.1 Bezier shapes

4.1.1 Definition

Bezier curves and surfaces were named after the automotive engineer Pierre Bezier, a pioneer of computer-aided design at the time of his tenure at the French car manufacturing company Renault [95]. A triangular Bezier patch is a three-dimensional surface element parametrized in terms of barycentric coordinates $\boldsymbol{\chi} = (u, v, w)^T$ such that

$$u + v + w \equiv 1 \tag{4.1}$$

Given a net of control points $\mathbf{C_i} \in \mathbb{R}^3$ labeled by a triplet of indices $\mathbf{i} = ijk$ such that $|\mathbf{i}| = i + j + k = n$, any point on the patch $\mathcal{S}$ is given by

$$P(\boldsymbol{\chi}) = \sum_{|\mathbf{i}|=n} B^n_{\mathbf{i}}(\boldsymbol{\chi})\mathbf{C_i} \tag{4.2}$$

where the so-called Bernstein polynomials are defined as

$$B^n_{\mathbf{i}}(\boldsymbol{\chi}) = \frac{n!}{i!j!k!}u^i v^j w^k \tag{4.3}$$

The control mesh of a patch of degree n is thus comprised of $N_c = \frac{(n+1)(n+2)}{2}$ control points. By consequent, a Bezier patch of degree $n = 1$ is nothing but a triangular, planar facet as $N_c = \frac{2 \cdot 3}{2} = 3$ in this case. Hence, a shape model comprised solely of Bezier patches of degree one is a polyhedron. When the patch degree is strictly greater than one and all control points are not

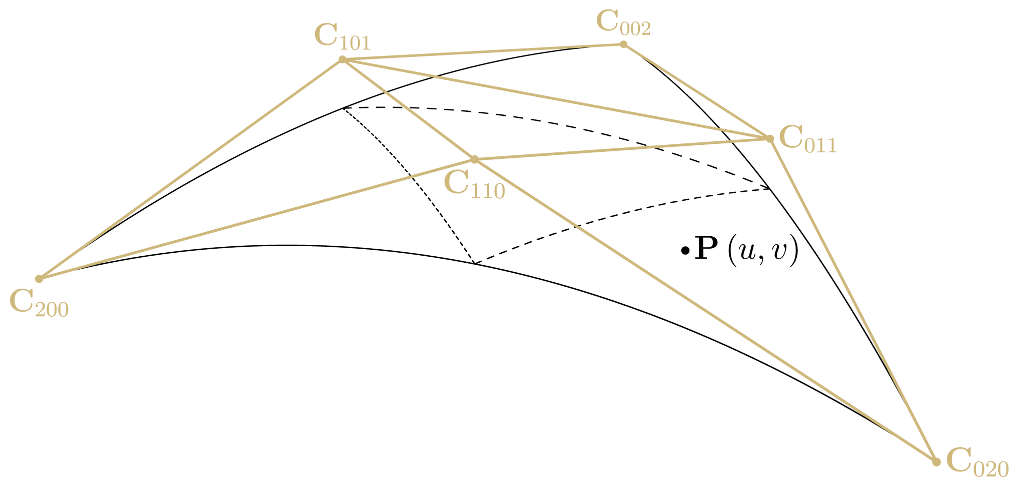


Figure 4.1: Bezier patch of degree two with its control net

coplanar, the interpolated surface is curved. An illustration of a Bezier triangle of degree two is given on Figure 4.1.

A collection of Bezier triangles can be used to represent any shape of interest, like the asteroid shape model shown in Figure 4.2. Bezier triangles of order higher than 1 are able to capture curvature within them, whereas triangles - effectively Bezier triangles of order 1 - are purely flat. This can lead to an interesting trade-off where a surface can be captured at a given level of detail by fewer high-order elements than flat triangles, as shown in Figure 4.2. In addition, higher-order surface elements can be used to enforce slope continuity across element borders, a feat that flat triangles could never provide. This property is especially interesting for motion planning where the modeled terrain must be devoid of singularities [96].

4.1.2 Fitting

4.1.2.1 Squared-distance minimization

Assuming that a point cloud comprised of N elements $\tilde{\mathbf{P}}_{i=1,\dots,N}$, Liu et al. suggest fitting a Bezier shape $\mathcal{S}$ to these measurements by minimizing the following functional [97] :

$$J^2 = \sum_{i=1}^{N} \left[\hat{n}_i^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)\right]^2 = \sum_{i=1}^{N} \epsilon_i^2 = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \tag{4.4}$$

$\bar{\mathbf{P}}_i \in \mathcal{S}$ is $\tilde{\mathbf{P}}_i$'s *foot point*, the closest point to $\tilde{\mathbf{P}}_i$ belonging to $\mathcal{S}$ such that $\left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right) \times \hat{n}_i = \mathbf{0}$ as shown on Figure 4.3.

Minimizing the cost-function presented in Equation (4.4) is thus equivalent to minimizing the square of normal distances between the point cloud and the fitted surface. Relying to this normal-distance criterion is an approximation to the actual surface-to-surface distance, but results in a consistent scheme.

Let $\mathbf{X}$ be the set of all N_C control points forming the control mesh of a Bezier shape. Let $\mathbf{C}_k$ be one of such control points:

$$\mathbf{X} = \begin{pmatrix} \ldots & \mathbf{C}_k^T & \ldots \end{pmatrix}^T_{3N_C \times 1} \tag{4.5}$$

Liu et al. computed the first-order partials of J^2 with respect to $\mathbf{C}_k$ under the assumption that the normal vector $\hat{n}_i$ is invariant. This approximation is not made in this paper, as using the exact first degree partials would improve convergence properties. The i-th fitting residual reads

$$\epsilon_i = \hat{n}_i^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right) \tag{4.6}$$

Linearizing ϵ_i relative to the current control mesh $\mathbf{X}^*$ and associated $\left(\hat{n}_i^*, \bar{\mathbf{P}}_i^*\right)$, we obtain

$$\hat{n}_i^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right) \simeq \hat{n}_i^{*T} \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i^*\right) + \delta\hat{n}_i^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i^*\right) - \hat{n}_i^{*T}\delta\bar{\mathbf{P}}_i \tag{4.7}$$

$$= \mathcal{Y}_i - \mathcal{H}_i \delta\mathbf{X} \tag{4.8}$$

where $\mathcal{Y}_i = \hat{n}_i^{*T} \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i^*\right)$ and $\mathcal{H}_i$ is a row vector of size $3N_C$. The k-th contiguous 3-column block of $\mathcal{H}_i$ corresponding to the k-th control point reads

$$\mathcal{H}_i^k = \hat{n}_i^{*T} \frac{\partial \bar{\mathbf{P}}_i}{\partial \mathbf{C}_k} - \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i^*\right)^T \frac{\partial \hat{n}_i}{\partial \mathbf{C}_k} \tag{4.9}$$

Assuming that the global index k corresponds to the triplet $\mathbf{k} = ij(n-i-j)$,

$$\frac{\partial \bar{\mathbf{P}}_i}{\partial \mathbf{C}_k} = B^n_{\mathbf{k}} I_3 \tag{4.10}$$

In addition, the normal at a Bezier triangle point is given by $\hat{n}_i^* = \frac{\mathbf{P}_u^* \times \mathbf{P}_v^*}{\|\mathbf{P}_u^* \times \mathbf{P}_v^*\|}$ with $\mathbf{P}_u = \frac{\partial \bar{\mathbf{P}}}{\partial u}$ and $\mathbf{P}_v = \frac{\partial \bar{\mathbf{P}}}{\partial v}$, hence

$$\frac{\partial \hat{n}_i}{\partial \mathbf{C}_k} = \frac{1}{\|\mathbf{P}_u \times \mathbf{P}_v\|} \left(I_3 - \frac{[\tilde{\mathbf{P}_u}]\mathbf{P}_v \left(\mathbf{P}_u \times \mathbf{P}_v\right)^T}{\|\mathbf{P}_u \times \mathbf{P}_v\|^2} \right) \left([\tilde{\mathbf{P}_u}] \frac{\partial \mathbf{P}_v}{\partial \mathbf{C}_k} - [\tilde{\mathbf{P}_v}] \frac{\partial \mathbf{P}_u}{\partial \mathbf{C}_k} \right) \tag{4.11}$$

Writing

$$\mathbf{Y} = \begin{pmatrix} \mathcal{Y}_1 \\ \vdots \\ \mathcal{Y}_N \end{pmatrix} \tag{4.12}$$

$$H = \begin{bmatrix} \mathcal{H}_1 \\ \vdots \\ \mathcal{H}_N \end{bmatrix} \tag{4.13}$$

The residuals become

$$\boldsymbol{\epsilon} = \mathbf{Y} - H\mathbf{X} \tag{4.14}$$

The deviation in the control mesh $\delta\mathbf{X}$ minimizing J^2 is the solution to

$$\left(H^T H\right) \delta\mathbf{X} = H^T \delta\mathbf{Y} \tag{4.15}$$

where $\delta\mathbf{Y} = \boldsymbol{\epsilon}$. Unfortunately, $H^T H$ is systematically poorly conditioned. This stems from the fact that a foot point measurement only carries information in the surface normal direction. As such, tangential displacements of the control points are poorly constrained from the foot points alone.

There are several ways to mitigate this issue, among acquiring more measurements, re-parametrizing the problem, or adding another component to the cost function. Although acquiring more measurements is straightforward and could help in some specific geometry cases, it does not address the fundamental observability deficiency previously discussed. Re-parametrizing the problem is a possibility, as one could for instance constrain $\delta\mathbf{X}$ to line up with observable directions, such as the surface normals at the control points. A similar goal can be achieved by augmenting the cost function J^2 with another term. This approach is certainly the easiest from an implementation standpoint, although the tuning flexibility it brings can become an issue itself.

Let a new shape-fitting cost function J' be defined as

$$J'^2 = J^2 + \delta\mathbf{X}^T \mathcal{Q} \delta\mathbf{X} \tag{4.16}$$

where J is the original cost function corresponding to Equation (4.15). The weighing matrix $\mathcal{Q}$ is a tuning parameter for us to choose. $\mathcal{Q} = \alpha I$ with $\alpha \in \mathbb{R}^+$ is a classical pick, which effectively penalizes the deviation directions isotropically. If simple, this approach is not well suited to the present case for two reasons. First, the deviation of the control points along their normal is already well captured in $H^T H$. Second, robustly picking a proper value of α is not a straightforward task.

Looking back at the observability issue we are dealing with, it makes sense to only penalize the *tangential* motion of the control points. That is, for every control point $\mathbf{C_k}$, the norm of $\left(I - \hat{n}_\mathbf{k}\hat{n}_\mathbf{k}^T\right)\delta\mathbf{C_k}$ should be minimized. Noting that the matrix $I - \hat{n}_\mathbf{k}\hat{n}_\mathbf{k}^T$ is representative of an orthogonal projection onto the orthogonal of $\hat{n}_\mathbf{k}$, the simplification $\left(I - \hat{n}_\mathbf{k}\hat{n}_\mathbf{k}^T\right)^2 = I - \hat{n}_\mathbf{k}\hat{n}_\mathbf{k}^T$ can be applied.

Minimizing this unobservable component of the control points deviations can thus be achieved if $\mathcal{Q}$ takes the form

$$\mathcal{Q} = \alpha \begin{bmatrix} \left(I - \hat{n}_1\hat{n}_1^T\right) & 0 & \vdots & 0 \\ 0 & \left(I - \hat{n}_2\hat{n}_2^T\right) & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \left(I - \hat{n}_{N_C}\hat{n}_{N_C}^T\right) \end{bmatrix} \tag{4.17}$$

where

$$\alpha \equiv \frac{N}{N_P} \tag{4.18}$$

Accounting for this penalty on the tangential motion of each control point, the normal equations becomes

$$\left(H^T H + \mathcal{Q}\right)\delta\mathbf{X} = H^T \delta\mathbf{Y} \tag{4.19}$$

Provided a sufficient number of foot points falling into each patch, the above equation has a unique solution. Patches that are not seen in any measurements have their control points removed from the fitting problem, so as to ensure a non-singular fitting at all times.

Information matrix sparsity

The information matrix $H^T H + \mathcal{Q}$ is $3N_C \times 3N_C$, with N_C potentially in the thousands. The assembly and solving of Equation 4.19 could lead to a potential bottleneck. However, the local nature of the control points in the shape definition makes the information matrix extremely sparse: each row of H matches a given measurement $\tilde{P}_i$. The non-zero components on this row correspond to the N_c control points of the patch where the measurement's foot point $\bar{P}_i$ falls into. In addition, the information matrix is obviously symmetric. From an implementation standpoint, this makes the information matrix ideally represented by a sparse matrix for which a Cholesky decomposition can be computed, which drastically alleviates the computational burden.

4.1.2.2 Finding the foot points

The computation of each $\bar{\mathbf{P}}_i$ can be carried out like so: the normals $\hat{n}$ over a Bezier batch are spanned by

$$\hat{n} \propto \frac{\partial \bar{P}}{\partial u} \times \frac{\partial \bar{P}}{\partial v} \tag{4.20}$$

Obviously, we have found the desired $\bar{\mathbf{P}}_i$ matching a measurement $\tilde{\mathbf{P}}_i$ when the following is satisfied

$$\left(\frac{\partial \bar{P}}{\partial u}\right)^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right) = 0 \tag{4.21}$$

$$\left(\frac{\partial \bar{P}}{\partial v}\right)^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right) = 0 \tag{4.22}$$

A first-order expansion of the above equations about the current barycentric coordinate estimate $\boldsymbol{\chi}^*$ yields

$$\left(\frac{\partial \bar{P}}{\partial u}\right)^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)\Big|^* - \frac{\partial \mathbf{P}_i}{\partial u}\frac{\partial \mathbf{P}}{\partial \boldsymbol{\chi}}\delta\boldsymbol{\chi} + \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)^T \frac{\partial^2 \mathbf{P}}{\partial \boldsymbol{\chi}\partial u}\delta\boldsymbol{\chi} = 0 \tag{4.23}$$

$$\left(\frac{\partial \bar{P}}{\partial v}\right)^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)\Big|^* - \frac{\partial \mathbf{P}_i}{\partial v}\frac{\partial \mathbf{P}}{\partial \boldsymbol{\chi}}\delta\boldsymbol{\chi} + \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)^T \frac{\partial^2 \mathbf{P}}{\partial \boldsymbol{\chi}\partial v}\delta\boldsymbol{\chi} = 0 \tag{4.24}$$

The deviation to the foot point barycentric coordinates $\delta\boldsymbol{\chi}$ is the solution to

$$\begin{bmatrix} \frac{\partial \mathbf{P}_i}{\partial u}\frac{\partial \mathbf{P}}{\partial \boldsymbol{\chi}} - \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)^T \frac{\partial^2 \mathbf{P}}{\partial \boldsymbol{\chi}\partial u} \\ \frac{\partial \mathbf{P}_i}{\partial u}\frac{\partial \mathbf{P}}{\partial \boldsymbol{\chi}} - \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)^T \frac{\partial^2 \mathbf{P}}{\partial \boldsymbol{\chi}\partial u} \end{bmatrix} \delta\boldsymbol{\chi} = \begin{pmatrix} \left.\left(\frac{\partial \bar{P}}{\partial u}\right)^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)\right|^* \\ \left.\left(\frac{\partial \bar{P}}{\partial v}\right)^T \left(\tilde{\mathbf{P}}_i - \bar{\mathbf{P}}_i\right)\right|^* \end{pmatrix} \tag{4.25}$$

The foot point computation is initialized by finding the control point that is the closest to a given measurement $\tilde{\mathbf{P}}_i$. The patches owning this control point are tested for the existence of the foot point $\bar{\mathbf{P}}_i$, starting with the initial guess $u^* = v^* = \frac{1}{3}$.The above deviation is iteratively computed and added to $\boldsymbol{\chi}^*$ for a fixed number of iterations, unless convergences occurs first. If the above algorithm converges to barycentric coordinates that belong to the unit triangle, then the foot point to the current measurement has been found. If the search over the patch subset is inconclusive, the measurement is rejected.

4.1.2.3 A-priori generation

A key point in surface fitting is the utmost importance of the initial a-priori shape provided to the shape fitter. Because the fitting process is essentially a linearization of the equations at hand, large deviations between the desired shape and its first guess will lead to the divergence of the shape fitter.

When a point cloud providing a sufficient broad coverage of the target is available, the problem is well-posed. Powerful techniques such as Poisson surface reconstruction (PSR) [4] can be used to reconstruct an underlying indicator function whose gradient corresponds to the point cloud normals. Then, an algorithm akin to Marching Cubes [98] provides a tesselated surface (typically, a polyhedron) from points that are sampled from the indicator function isosurface. However, these techniques are better suited to offline processing than sequential shape fitting, in addition to being quite burdensome computationally.

For the developed method to be applicable to arbitrary small body shapes, from nearly-ellipsoidal bodies to convoluted topographies, the a-priori Bezier shape model is generated by applying PSR to a bundle-adjusted, sparsified, globally covering point cloud obtained by successive ICP calls and bundle-adjustment of collected point clouds. Once this point cloud satisfies a prescribed criterion, it is passed to a PSR pipeline that generates a polyhedron. This polyhedron is then decimated until it is comprised of a prescribed number of edges. Finally, the decimated polyhedron is converted into an equivalent Bezier shape model of degree one, and ultimately elevated to degree two by adding 3 control points to each triangular facet. The decimation of the PSR output is made necessary by the subsequent training of the uncertainty model: given a fixed number of training points, too many surface elements will yield a poor training performance as too few points fall in each surface element. This whole process guarantees that the a-priori shape model is well-behaved and within the basin of convergence of the iterative shape fitter.

4.2 Poisson Surface Reconstruction

Poisson Surface Reconstruction is a relatively recent technique developed by Kazhdan et al. in 2006 to produce watertight surfaces from oriented point sets [4][99]. The present section provides a quick overview of PSR, leaving out derivation and implementation details. Given a point cloud augmented with oriented surface normals at each point, PSR proceeds by reconstructing an underlying indicator function χ evaluating to 0 inside the shape and 1 outside of it. $\hat{n} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is the application yielding the outward-oriented surface normal at every sample point. The goal of shape reconstruction is thus to find the scalar function χ such that its gradient equates the sampled normals at the sampled points. Solving the weak formulation of this problem leads to the Poisson equation

$$\Delta\chi = \nabla \cdot \hat{n} \tag{4.26}$$

where Δ is the Laplacian operator.

Once the indicator function has been computed, the surface of the shape model is effectively extracted by finding the iso-value best representative of the shape. This process is highlighted on Figure 4.4 This iso-value is found by averaging the indicator function at the provided sample points. A polyhedron is then constructed from the sampled iso-surface.

A number of PSR implementations exist in computer graphics. The Computational Geometry Algorithms Library (CGAL) is a software library of computational geometry algorithms [100]. CGAL's PSR implementation infers the polyhedron mesh resolution from the point cloud density and a user-provided upper bound on the radius of surface Delaunay balls, defined as a ball circumscribing a facet, centered on the surface and empty of vertices. Setting this criterion to too small of a value return a very high resolution mesh, which would have to be decimated to a lower resolution before being converted to a Bezier shape and fitted. On the contrary, setting this criterion to too high of a value would yield a low resolution polyhedron, possibly too coarse to serve as a good-enough a-priori for the Bezier shape fitter to converge. It thus appears there must be a balance between these two extreme behaviors. A value of 5 for this criterion appeared as a good compromise. In addition, ensuring that the input point clouds is relatively uniformly sampled allows to provide only 1% of the input points to the PSR pipeline. This allows the PSR pipeline to return a satisfying a-priori shape model from $1\% \cdot 2M \equiv 20000$ input points in a few seconds.

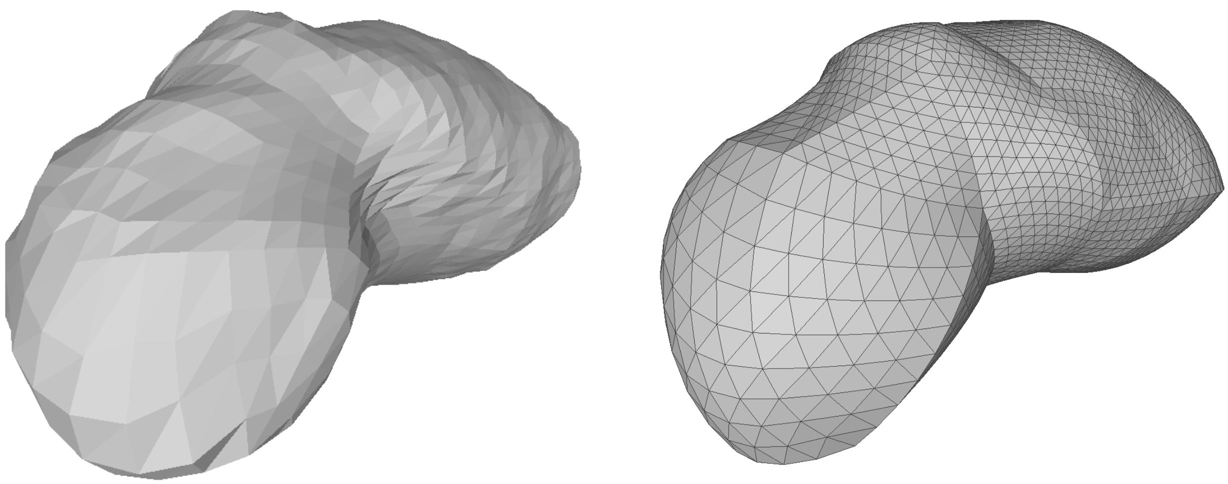


Figure 4.2: Left: Itokawa shape model comprised of 3072 Bezier 3-control point, order 1 triangles (1538 control points total). Right: Itokawa shape model comprised of 85 6-control point, order-2 Bezier triangles, with one such Bezier triangle being highlighted along with its control mesh

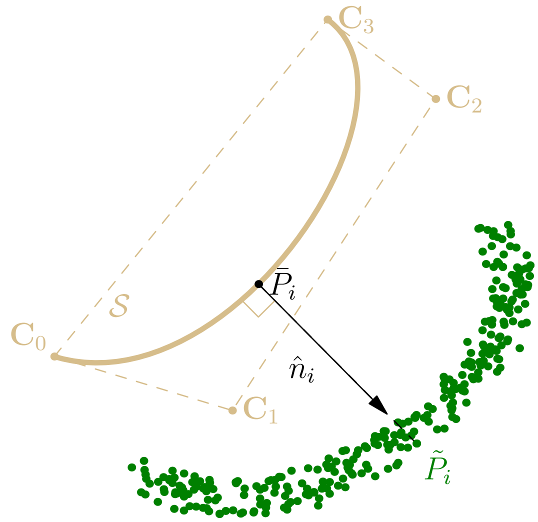


Figure 4.3: Illustration of the projection of $\tilde{\mathbf{P}}_i$ onto its foot point $\bar{\mathbf{P}}_i$ and associated normal $\hat{n}_i$

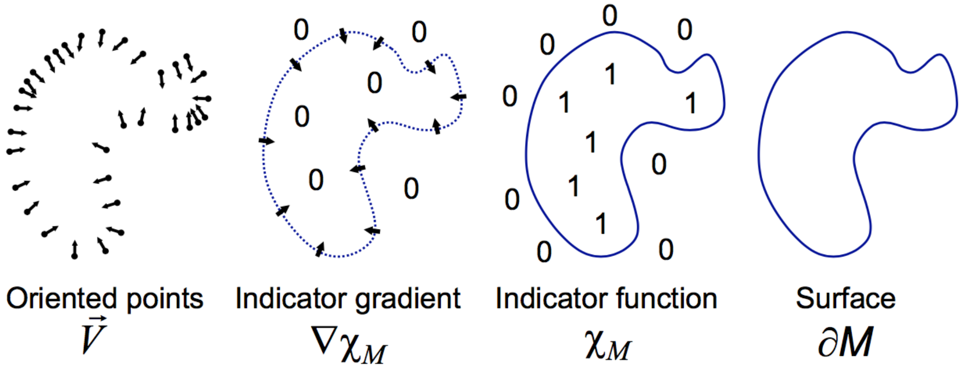


Figure 4.4: Illustration of the Poisson Surface Reconstruction (PSR) technique [4]. The inputs consists in a set of fully registered 3D points along with a corresponding set of oriented normal vectors $\hat{n}$. The normal vectors themselves are taken as the gradient of the so-called indicator function χ. The variational problem $\Delta\chi = \nabla \cdot \hat{n}$ is then assembled and solved. The surface itself is finally extracted from an indicator function isosurface instantiated over the input points.

Chapter 5

Shape uncertainty model

5.1 Range uncertainty caused by a Gaussian control mesh

In what follows, the control mesh coordinates $\mathbf{X}$ of a given Bezier triangle as a Gaussian vector of mean $\bar{\mathbf{X}}$ and covariance

$$P_{\mathbf{X}} = \mathrm{E}\left(\left[\mathbf{X} - \bar{\mathbf{X}}\right]\left[\mathbf{X} - \bar{\mathbf{X}}\right]^T\right) \tag{5.1}$$

$$= \begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & P_{\mathbf{C_i C_i}} & \cdots & P_{\mathbf{C_i C_j}} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & P_{\mathbf{C_j C_i}} & \cdots & P_{\mathbf{C_j C_j}} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}_{3N_c \times 3N_c} \tag{5.2}$$

with

$$P_{\mathbf{C_i C_j}} = \mathrm{E}\left(\left[\mathbf{C_i} - \bar{\mathbf{C}}_\mathbf{i}\right]\left[\mathbf{C_j} - \bar{\mathbf{C}}_\mathbf{j}\right]^T\right) \tag{5.3}$$

A computed range measurement over the reconstructed shape model reads

$$\rho = \hat{u}^T \left(\mathbf{P}\left(\boldsymbol{\chi}\right) - \mathbf{S}\right) \tag{5.4}$$

Carrying out simple algebra, the expected value of the range measurement is

$$\mathrm{E}\left(\rho\right) = \hat{u}^T \left(\bar{\mathbf{P}}\left(\bar{\boldsymbol{\chi}}\right) - \mathbf{S}\right) \tag{5.5}$$

$$\mathrm{E}\left(\left[\rho - \bar{\rho}\right]^2\right) = \hat{u}^T P_{\mathbf{P}} \hat{u} \tag{5.6}$$

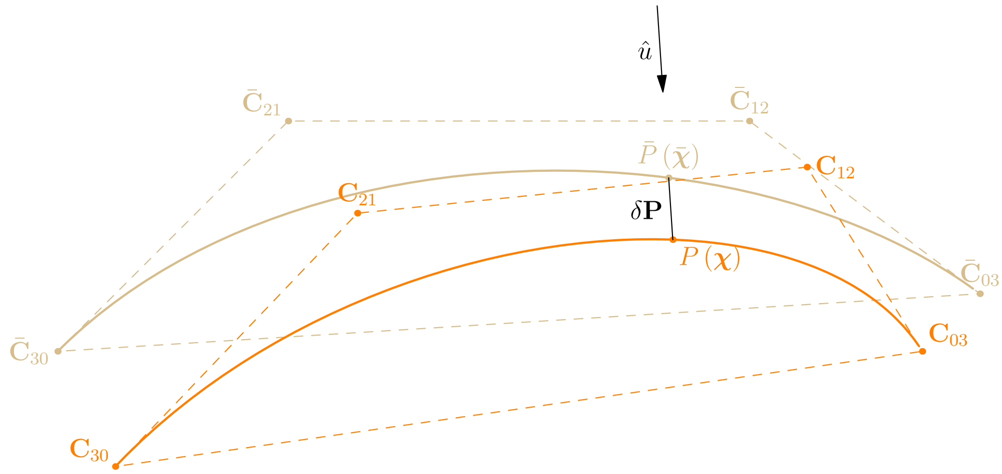


Figure 5.1: Shift in location of impacted point $\mathbf{P}$ along the measurement direction $\hat{u}$ as the control mesh $\mathbf{X} = \left[\mathbf{C}_0^T, \mathbf{C}_1^T, \mathbf{C}_2^T, \mathbf{C}_2^T\right]^T$ shifts away from its mean. The barycentric coordinates of $\mathbf{P}$ and $\bar{\mathbf{P}}$ are different

where

$$\mathrm{E}\left(\mathbf{P}\left(\boldsymbol{\chi}\right)\right) = \bar{\mathbf{P}}\left(\bar{\boldsymbol{\chi}}\right) = \sum_{|\mathbf{i}|=n} B_{\mathbf{i}}^n\left(\bar{\boldsymbol{\chi}}\right)\bar{\mathbf{C}}_{\mathbf{i}} \tag{5.7}$$

The derivation of the covariance $P_{\mathbf{P}} = \mathrm{E}\left(\left[\mathbf{P}\left(\boldsymbol{\chi}\right) - \bar{\mathbf{P}}\left(\bar{\boldsymbol{\chi}}\right)\right]\left[\mathbf{P}\left(\boldsymbol{\chi}\right) - \bar{\mathbf{P}}\left(\bar{\boldsymbol{\chi}}\right)\right]^T\right)$ can be carried out assuming that the change in the barycentric coordinates is small: $\|\delta\boldsymbol{\chi}\| = \|\boldsymbol{\chi} - \bar{\boldsymbol{\chi}}\| << 1$. Introduce the notation $\delta\mathbf{C}_{\mathbf{i}} = \mathbf{C}_{\mathbf{i}} - \bar{\mathbf{C}}_{\mathbf{i}}$. The coordinates $\boldsymbol{\chi}$ positioning a point on the disturbed surface spanned by corrupted control points $\mathbf{X}$ are not the same as the $\bar{\boldsymbol{\chi}}$ from the original surface. Expanding the difference in the positions,

$$\begin{aligned}
\mathbf{P}\left(\boldsymbol{\chi}\right) - \bar{\mathbf{P}}\left(\bar{\boldsymbol{\chi}}\right) &= \sum_{|\mathbf{i}|=n}\left(B_{\mathbf{i}}^n\left(\boldsymbol{\chi}\right)\mathbf{C}_{\mathbf{i}} - B_{\mathbf{i}}^n\left(\bar{\boldsymbol{\chi}}\right)\bar{\mathbf{C}}_{\mathbf{i}}\right) && (5.8)\\
&\simeq \sum_{|\mathbf{i}|=n}\left(\left[B_{\mathbf{i}}^n\left(\bar{\boldsymbol{\chi}}\right) + \frac{\partial B_{\mathbf{i}}^n}{\partial\boldsymbol{\chi}}\sum_{|\mathbf{k}|=n}\frac{\partial\boldsymbol{\chi}}{\partial\mathbf{C}_{\mathbf{k}}}\delta\mathbf{C}_{\mathbf{k}}\right]\left[\bar{\mathbf{C}}_{\mathbf{i}} + \delta\mathbf{C}_{\mathbf{i}}\right] - B_{\mathbf{i}}^n\left(\bar{\boldsymbol{\chi}}\right)\bar{\mathbf{C}}_{\mathbf{i}}\right) && (5.9)\\
&\simeq \sum_{|\mathbf{i}|=n}\left(B_{\mathbf{i}}^n\left(\bar{\boldsymbol{\chi}}\right)\delta\mathbf{C}_{\mathbf{i}} + \bar{\mathbf{C}}_{\mathbf{i}}\frac{\partial B_{\mathbf{i}}^n}{\partial\boldsymbol{\chi}}\sum_{|\mathbf{k}|=n}\frac{\partial\boldsymbol{\chi}}{\partial\mathbf{C}_{\mathbf{k}}}\delta\mathbf{C}_{\mathbf{k}}\right) && (5.10)
\end{aligned}$$

The term

$$\mathbf{Z} = \sum_{|\mathbf{k}|=n}\frac{\partial\boldsymbol{\chi}}{\partial\mathbf{C}_{\mathbf{k}}}\delta\mathbf{C}_{\mathbf{k}} \tag{5.11}$$

looks challenging at first, because the partial derivatives $\frac{\partial \boldsymbol{\chi}}{\partial \mathbf{C_k}}$ do not have an easy closed-form. However, one can leverage the fact that the new point $\mathbf{P}(\boldsymbol{\chi})$ must be found along the direction of the measurement $\hat{u}$, as shown on Figure 5.1. Thus

$$\hat{u} \times \left(\mathbf{P}(\boldsymbol{\chi}) - \bar{\mathbf{P}}(\bar{\boldsymbol{\chi}})\right) = \mathbf{0} \tag{5.12}$$

This implies

$$\hat{u} \times \sum_{|\mathbf{i}|=n} \left(B^n_{\mathbf{i}}(\bar{\boldsymbol{\chi}})\, \delta \mathbf{C_i} + \bar{\mathbf{C}}_{\mathbf{i}} \frac{\partial B^n_{\mathbf{i}}}{\partial \boldsymbol{\chi}} \mathbf{Z} \right) = \mathbf{0} \tag{5.13}$$

A linear system

$$A\mathbf{Z} = B \tag{5.14}$$

can be formed, where

$$A = [\widetilde{\hat{u}}] \sum_{|\mathbf{i}|=n} \bar{\mathbf{C}}_{\mathbf{j}} \frac{\partial B^n_{\mathbf{j}}}{\partial \boldsymbol{\chi}} = [\widetilde{\hat{u}}]T \tag{5.15}$$

$$\mathbf{B} = -[\widetilde{\hat{u}}] \sum_{|\mathbf{j}|=n} B^n_{\mathbf{j}}(\bar{\boldsymbol{\chi}})\, \delta \mathbf{C_j} \tag{5.16}$$

A is a 3-by-2 matrix and thus have a rank that is at most 2. T is a 3×2 matrix whose columns are tangent to the surface at the impact point $\bar{\boldsymbol{\chi}}$. The rank of A thus becomes strictly less than 2 when $\hat{u}$ is within the plane spanned by the columns of T. This corresponds to a 90° incidence between $\hat{u}$ and the surface normal, indicating a measurement acquired tangentially to the surface. This uncertainty model hence naturally penalizes measurements acquired in such least-observable directions. A least-squares solution to that system is thus

$$\mathbf{Z} = \left(A^T A\right)^{-1} A^T B \tag{5.17}$$

If $\operatorname{rank}(A)$ is exactly equal to 2, then $\left(A^T A\right)^{-1}$ exists. The difference in the positions becomes

$$\mathbf{P}(\boldsymbol{\chi}) - \bar{\mathbf{P}}(\bar{\boldsymbol{\chi}}) = \sum_{|\mathbf{i}|=n} \left(B^n_{\mathbf{i}}(\bar{\boldsymbol{\chi}})\, \delta \mathbf{C_i} + \bar{\mathbf{C}}_{\mathbf{i}} \frac{\partial B^n_{\mathbf{i}}}{\partial \boldsymbol{\chi}} \left(A^T A\right)^{-1} A^T B \right) \tag{5.18}$$

Skipping a few unnecessary steps of algebra, the following compact expression can be obtained

$$P_{\mathbf{P}} = \mathrm{E}\left(\left[\mathbf{P}(\boldsymbol{\chi}) - \bar{\mathbf{P}}(\bar{\boldsymbol{\chi}})\right]\left[\mathbf{P}(\boldsymbol{\chi}) - \bar{\mathbf{P}}(\bar{\boldsymbol{\chi}})\right]^T\right) \tag{5.19}$$

$$= W^T P_{\mathbf{X}} W \tag{5.20}$$

with

$$W = \mathcal{M}(I_3 + \mathcal{K}) \tag{5.21}$$

where $\mathcal{M}$ is a $3N_c \times 3$ matrix given by

$$\mathcal{M} = \begin{bmatrix} B^n_{\mathbf{i}_1}(\bar{\boldsymbol{\chi}}) I_3 & \ldots & B^n_{\mathbf{i}_h}(\bar{\boldsymbol{\chi}}) I_3 & \ldots & B^n_{\mathbf{i}_{N_c}}(\bar{\boldsymbol{\chi}}) I_3 \end{bmatrix}^T \tag{5.22}$$

such that

$$\mathcal{M}^T P_{\mathbf{X}} \mathcal{M} = \sum_{|\mathbf{i}|=n} \sum_{|\mathbf{k}|=n} B^n_{\mathbf{i}}(\bar{\boldsymbol{\chi}}) B^n_{\mathbf{k}}(\bar{\boldsymbol{\chi}}) P_{\mathbf{C_i C_k}} \tag{5.23}$$

and $\mathcal{K}$ a 3×3 matrix given by

$$\mathcal{K} = [\widetilde{\hat{u}}] A \left(A^T A\right)^{-1} \sum_{|\mathbf{k}|=n} \left(\mathbf{C_k} \frac{\partial B^n_{\mathbf{k}}}{\partial \boldsymbol{\chi}}\right)^T \tag{5.24}$$

The variance of the range measurement along $\hat{u}$ solely due to an error in the control mesh $\mathbf{X}$ modeled by $P_{\mathbf{X}}$ is thus given by

$$\sigma^2 = \mathrm{E}\left([\rho - \bar{\rho}]^2\right) = \hat{u}^T \mathbf{P_P} \hat{u} = \hat{u}^T W^T P_{\mathbf{X}} W \hat{u} = \mathbf{v}^T P_{\mathbf{X}} \mathbf{v} \tag{5.25}$$

This effectively provides an uncertainty model that can be evaluated at any time to obtain a predicted confidence in a given range measurement. For this prediction to be consistent and useful to a navigation filter, the model must be tuned over the range data that was utilized in the shape reconstruction phase.

5.2 Tuning of the uncertainty model

5.2.1 Log-likelihood maximization

In order to be used within the navigation filter, the uncertainty model described in the previous section must be tuned so as to be representative of the shape uncertainties that are effectively present. In other words, $P_{\mathbf{X}}$ must be determined from the range data. Assume that N range residuals ϵ_i evaluated between 3D points $\tilde{P}_i$ and their projected foot point $\bar{P}_i$ along the surface normal $\hat{n}_i$ are available. Let L be the log-likelihood function of all measurements, itself a function of the hyper-parameter $P_{\mathbf{X}}$. L reads

$$L = \log p\left(\epsilon_1, \ldots, \epsilon_N; P_{\mathbf{X}}\right) \tag{5.26}$$

where $p\left(\epsilon_1, \ldots, \epsilon_N; P_{\mathbf{X}}\right)$ is the joint distribution of all measurements. Arguing that the measurements are actually independent, it becomes

$$L = \log \prod_{i=1}^{N} p\left(\epsilon_i; P_{\mathbf{X}}\right) = \sum_{i=1}^{N} \log p\left(\epsilon_i; P_{\mathbf{X}}\right) \tag{5.27}$$

with

$$p\left(\epsilon_i; P_{\mathbf{X}}\right) = \mathcal{N}_{\epsilon_i}\left(0, \sigma_i^2\right) \tag{5.28}$$

$$\sigma_i^2 = \mathbf{v}_i^T P_{\mathbf{X}} \mathbf{v}_i \tag{5.29}$$

$$\mathbf{v}_i = W_i \hat{u}_i \tag{5.30}$$

Although the ϵ_i all depend upon $P_{\mathbf{X}}$, they are non-identically distributed since the σ_i^2 are measurement-dependent, due to their relationship with the direction measurement $\hat{u}_i$ and impact coordinates. Traditional log-likelihood maximization looks for the best estimate of an hyper parameter by finding its value canceling the gradient of the log-likelihood. In the present case, this amounts to finding $P_{\mathbf{X}}$ such that

$$\frac{\partial L}{\partial P_{\mathbf{X}}} = \underline{\underline{\mathbf{0}}} \tag{5.31}$$

Since

$$\mathcal{N}_{\epsilon_i}\left(0, \sigma_i^2\right) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2}\left(\frac{\epsilon_i}{\sigma_i}\right)^2} \tag{5.32}$$

expanding the log-likelihood gives

$$\sum_{i=1}^{N} \log\left(\mathcal{N}_{\epsilon_i}\left(0, \sigma_i^2\right)\right) \propto -\sum_{i=1}^{N} \left(\log \sigma_i^2 + \frac{\epsilon_i^2}{\sigma_i^2}\right) \tag{5.33}$$

So setting the Jacobian of L with respect of $P_{\mathbf{X}}$ to $\underline{\underline{\mathbf{0}}}$ yields

$$\sum_{i=1}^{N}\left(\frac{1}{\sigma_i^2}-\frac{\epsilon_i^2}{\sigma_i^4}\right)\frac{\partial\sigma_i^2}{\partial P_{\mathbf{X}}}=\underline{\underline{\mathbf{0}}} \tag{5.34}$$

Since $\sigma_i^2 = \mathbf{v}_i^T P_{\mathbf{X}} \mathbf{v}_i$, it is actually straightforward to show that the (k,l) component of the matrix $\frac{\partial\sigma_i^2}{\partial P_{\mathbf{X}}}$ is

$$\left[\frac{\partial\sigma_i^2}{\partial P_{\mathbf{X}}}\right]_{k,l}=\left[\mathbf{v}_i\mathbf{v}_i^T\right]_{k,l}=\mathbf{v}_i(k)\mathbf{v}_i(l) \tag{5.35}$$

In other words,

$$\frac{\partial\sigma_i^2}{\partial P_{\mathbf{X}}}=\mathbf{v}_i\mathbf{v}_i^T \tag{5.36}$$

So the sought-for covariance $P_{\mathbf{X}}$ satisfies the following matrix equality

$$\sum_{i=1}^{N}\left(\frac{1}{\sigma_i^2}-\frac{\epsilon_i^2}{\sigma_i^4}\right)\mathbf{v}_i\mathbf{v}_i^T=0 \tag{5.37}$$

Unfortunately, Equation (5.37) cannot be solved by means of classical gradient-based methods where one seeks an increment $\delta P_{\mathbf{X}}$ to apply to $P_{\mathbf{X}}$. Indeed, while one would expect $\delta P_{\mathbf{X}}$ to increase L, it is clear from Equation (5.37) that the gradient of L cancels itself as $\|P_{\mathbf{X}}\| \to \infty$. By consequent, the applied $\delta P_{\mathbf{X}}$ would make $P_{\mathbf{X}}$ diverge to infinity. This would effectively cancel Equation (5.37), but L would not be maximized. Alternative optimization techniques are thus necessary.

5.2.2 Particle-Swarm-Optimization

Because gradient-based descent is unusable, the tuning of $P_{\mathbf{X}}$ relies on a population-based stochastic method known as *Particle-Swarm-Optimization* [101]. Assuming the goal is to find the global extremum of $J : \mathbf{x} \in \mathbb{R}^p \mapsto J(\mathbf{x}) \in \mathbb{R}$ over $\mathbb{R}^p$, PSO proceeds by sampling a number of test values in $\mathbb{R}^p$, the *particles*. J is evaluated at each particle, before an information exchange phase takes place. At the end of this phase, the particles' position in state-space are updated based on: the *global* best state found by the population, the *local* best state found by each particle, and an *inertia* term accounting for the motion of the particles within the state-space. The benefit of this information exchange is illustrated on Figure 5.2, where the optimizer is applied to the so-called Ackley's function [102], a highly non-convex bivariate function featuring a number of local minima.

The example on Figure 5.2 had its arguments translated so that the global minimum lies at $(1, 1)$. The optimizer successfully finds the global minimum, despite the sparse population which was only comprised of 100 particles. This way, the optimizer does not stall if local minima are encountered, provided that the state space is sufficiently populated.

PSO requires the state-space to be bounded to a region of interest. In the present case, the state to optimize is $\mathbf{L}_{P_\mathbf{X}}$, a parametrization of $P_\mathbf{X}$, such that the observed residuals are best explained from a maximum likelihood standpoint. The state space is thus bounded by

$$\mathbf{L}_{P_\mathbf{X},\min} \leq \mathbf{L}_{P_\mathbf{X}} \leq \mathbf{L}_{P_\mathbf{X},\max} \tag{5.38}$$

where $\leq$ is a component-wise operator. Here, the function to maximize is no else but the log-likelihood evaluated over all the measurements

$$L = -\sum_{i=1}^{N} \left(\log \sigma_i^2 + \frac{\epsilon_i^2}{\sigma_i^2} \right) \tag{5.39}$$

5.2.3 Search space initialization

The search space can be narrowed-down by finding a realistic initial guess for $P_\mathbf{X}$, or rather its associated parametrization $\mathbf{L}_{P_\mathbf{X}}$: a crude yet satisfying guess for $P_\mathbf{X}$ is

$$P_\mathbf{X} = \alpha I_{N_c} \tag{5.40}$$

which implies

$$\sigma_i^2 = \alpha \mathbf{v}_i^T \mathbf{v}_i \tag{5.41}$$

Plugging this into Equation (5.37),

$$\sum_{i=1}^{N} \epsilon_i^2 \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i \mathbf{v}_i^T \mathbf{v}_i} = \alpha \sum_{i=1}^{N} \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i} \tag{5.42}$$

This matrix equation can be turned into a vector equality by flattening the matrices on the left-hand and right-hand side. One can then extract a least-squares approximation of α.

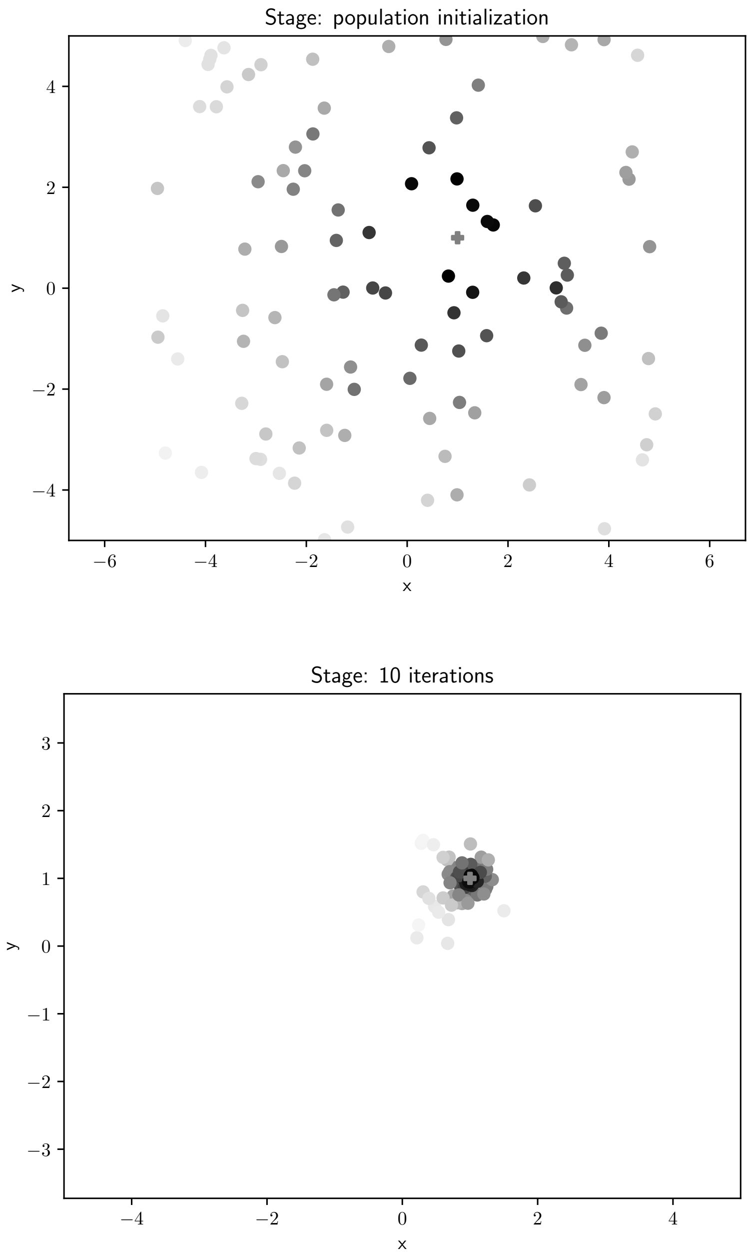


Figure 5.2: Illustration of the PSO optimizer seeking to minimize the translated two-dimensional Ackley's function with 100 particles at initialization (top) and after ten iterations (bottom)

5.2.4 Covariance parametrization

The covariance in the p-th patch's control mesh $P_{\mathbf{X}_p}$ reads in its most general form

$$P_{\mathbf{X}_p} = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & P_{\mathbf{C_i}\mathbf{C_i}} & \dots & P_{\mathbf{C_i}\mathbf{C_j}} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & P_{\mathbf{C_j}\mathbf{C_i}} & \dots & P_{\mathbf{C_j}\mathbf{C_j}} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}_{3N_c \times 3N_c} \tag{5.43}$$

Obviously, the PSO optimizer used to train a patch covariance should not operate on $3N_c \times 3N_c$ states as this could result in evaluating the log-likelihood using non-positive, symmetric definite covariance matrices. To ensure that each $P_{\mathbf{X}_p}$ is consistent, the following is enforced:

1. The control points within the same patch are uncorrelated, such that

$$P_{\mathbf{C_i}\mathbf{C_j}} = 0 \;\text{ if } \mathbf{C_i} \neq \mathbf{C_j} \tag{5.44}$$

2. Each non-zero partition $P_{\mathbf{C_i}\mathbf{C_j}}$ is parametrized as

$$P_{\mathbf{C_i}\mathbf{C_i}} = e^{\lambda_\mathbf{i}} I_3 \tag{5.45}$$

This parametrization of $P_{\mathbf{X}_p}$, $\mathbf{L}_{P_{\mathbf{X}_p}}$, that takes the form

$$\mathbf{L}_{P_{\mathbf{X}_p}} = \begin{pmatrix} \lambda_1 & \dots & \lambda_\mathbf{i} & \dots & \lambda_{N_c} \end{pmatrix}^T \tag{5.46}$$

will result in a positive symmetric definite $P_{\mathbf{X}_p}$ through Equation (5.45).

5.3 Uncertainty Model Validation

5.3.1 Validation of model evaluation

The uncertainty model derived in 5.1 can be validated in a simple case considering a single quadratic patch of control mesh $\bar{\mathbf{X}}$ where the 6 control points are assigned prescribed covariances grouped in a patch covariance $P_{\mathbf{X}}$. The mean control mesh coordinates are provided in Table

Table 5.1: Mean control point coordinates and uncertainties

Control point	Coordinates	Standard deviation	Units
$\bar{\mathbf{C}}_{200}$	$\begin{pmatrix}0 & 0 & 0\end{pmatrix}^T$	0.0585	–
$\bar{\mathbf{C}}_{110}$	$\begin{pmatrix}1 & 0 & 0.5\end{pmatrix}^T$	0.0337	–
$\bar{\mathbf{C}}_{101}$	$\begin{pmatrix}0 & 1 & 0.3\end{pmatrix}^T$	0.0695	–
$\bar{\mathbf{C}}_{020}$	$\begin{pmatrix}2 & 0 & 0\end{pmatrix}^T$	0.0717	–
$\bar{\mathbf{C}}_{011}$	$\begin{pmatrix}1.5 & 0.5 & 1\end{pmatrix}^T$	0.0410	–
$\bar{\mathbf{C}}_{002}$	$\begin{pmatrix}0.5 & 2 & 0\end{pmatrix}^T$	0.1046	–

5.1, along with the standard deviation corresponding to each point covariance. Perturbed control meshes are drawn from the Gaussian distribution $\mathcal{N}_{\mathbf{X}}\left(\bar{\mathbf{X}}, P_{\mathbf{X}}\right)$ and ray traced from a fixed origin and direction. The distribution of ranges should obey the Gaussian distribution of standard deviation given by Equation (5.25). Figure 5.3 illustrates the nominal patch along with one possible outcome from the mesh distribution. 50,000 perturbed patches were generated and ray-traced along the same direction $\hat{u}$ and origin $\mathbf{S}$. $\mathbf{S}$ was positioned at $\begin{pmatrix}2 & 0 & 5\end{pmatrix}^T$ above the nominal patch. $\hat{u}$ was set such that the ray was directed from $\mathbf{S}$ towards the center of the nominal patch. The kernel density estimate of the range residuals is shown on Figure 5.4. The standard deviation in the range residuals matches the predicted one within 0.014%. The difference that can be observed between the residuals distribution and the Gaussian distribution it should conform with stems from the linearization error inherent to the uncertainty model.

5.3.2 Validation of model tuning

The tuning of the uncertainty model is demonstrated over the same patch as in 5.3.1. To this end, 50,000 range measurements are collected over the different outcomes of the quadratic patch drawn in 5.3.1. These measurements $\{\tilde{\mathbf{P}}_i\}$ are processed so as to extract the corresponding foot points $\{\bar{\mathbf{P}}_i\}$. The tuning point set can be seen on Figure 5.3. The point pairs are then passed to the uncertainty model trainer described in 5.2 to reconstruct the patch covariance. The PSOPT trainer is iterated 30 times with 500 particles. Table 5.2 summarizes the tuning performance. It can be seen that most of the uncertainty is well captured by the model, with the exception of the covariance of

Table 5.2: Comparison of tuned versus true control point standard deviations

Control point	Trained standard deviation	True standard deviation	Error (%)
$\bar{\mathbf{C}}_{\mathbf{200}}$	0.0551	0.0585	−5.8377
$\bar{\mathbf{C}}_{\mathbf{110}}$	0.0452	0.0337	34.1458
$\bar{\mathbf{C}}_{\mathbf{101}}$	0.0701	0.0695	0.8
$\bar{\mathbf{C}}_{\mathbf{020}}$	0.0696	0.0717	−3.0139
$\bar{\mathbf{C}}_{\mathbf{011}}$	0.0393	0.0410	−4.1186
$\bar{\mathbf{C}}_{\mathbf{002}}$	0.1051	0.1046	0.5493

$\bar{\mathbf{C}}_{\mathbf{110}}$ which is overestimated. Note that $\bar{\mathbf{C}}_{\mathbf{110}}$ was the smallest error component effectively present, and its overestimation is the direct consequence of the larger error sources present in the dataset that are more easily observable.

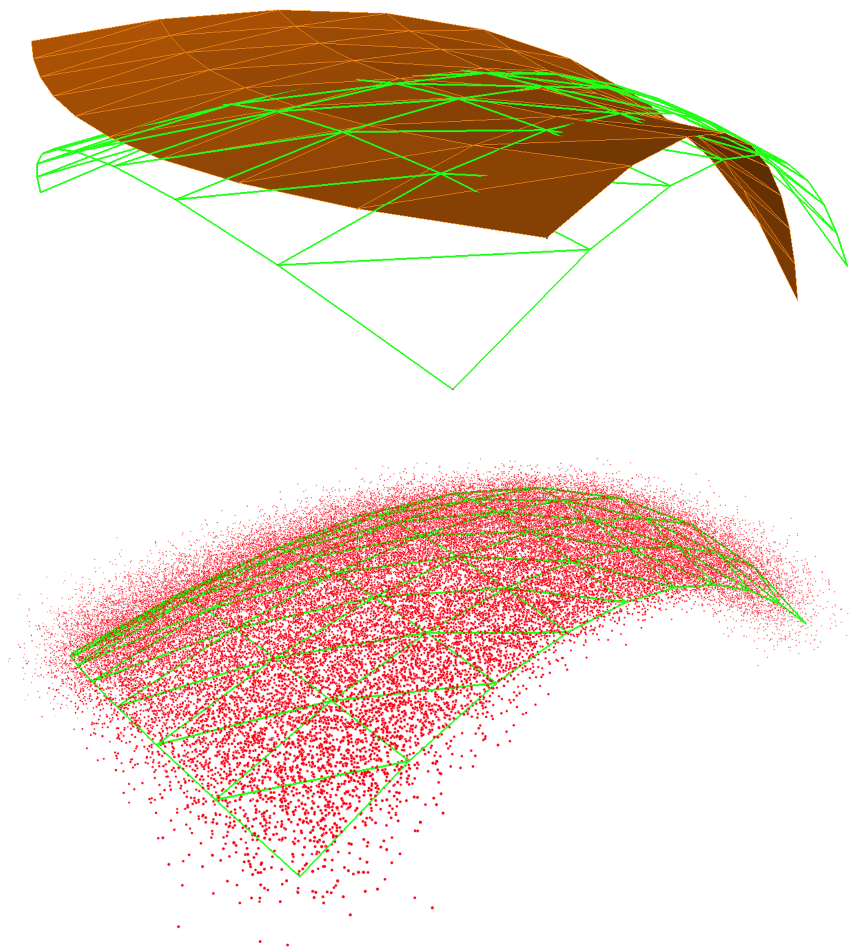


Figure 5.3: Top: tesselated nominal control mesh (green) and one outcome of the Gaussian mesh used in the Monte-Carlo validation of the uncertainty model (orange). Bottom: tuning dataset $\{\mathbf{P}_i\}$ overlaid over the true mesh

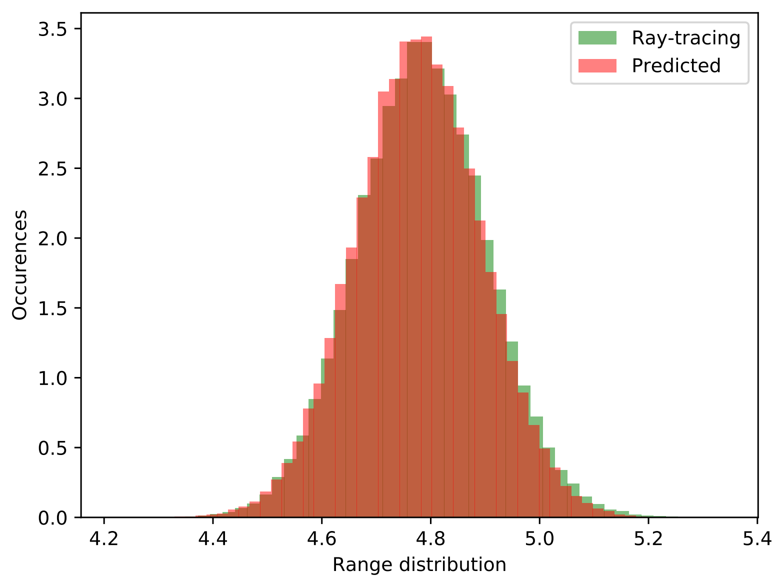


Figure 5.4: Distribution of numerical range residuals versus their predicted distribution from the analytical uncertainty model

Chapter 6

Uncertainty in Shape Inertias Arising From An Uncertain Shape

6.1 Methods

6.1.1 Inertia quantities

In what follows, 'Bezier tetrahedron' is used to described the closed volume subtended between a Bezier triangle τ and an arbitrary origin $\mathbf{O}$. Such a Bezier tetrahedron is shown in Figure 6.1.

6.1.1.1 Volume of a Bezier shape

The signed volume of a single Bezier Tetrahedron $\Delta\mathcal{V}$ is defined as

$$\Delta V = \iiint_{\delta\mathcal{V}} \mathrm{d}V$$

Using the divergence theorem, this becomes

$$\Delta V = \frac{1}{3} \oiint_{\delta\mathcal{V}} \mathbf{r}^T \mathrm{d}\mathbf{S}$$

where $\delta\mathcal{V}$ is the closed surface enclosing the volume of interest and $\mathbf{r}$ the position vector of the point of interest. It can noted that some simplifications may take place over the Bezier tetrahedron. First, it is clear than only the surface element corresponding to the Bezier triangle will have a non-zero contribution to this integral. Indeed, the position vector $\mathbf{r}$ originating from the origin and tracing the outline of the Bezier triangle is always tangent to the Bezier tetrahedron surface. As a result,

$\mathbf{r}^T \mathrm{d}\mathbf{S}$ cancels over these regions since $\mathrm{d}\mathbf{S}$ is normal to the surface. Hence the volume integral reduces to

$$\Delta V = \frac{1}{3} \iint_{\mathcal{T}} \mathbf{r}^T \mathrm{d}\mathbf{S}$$

where $\mathcal{T}$ is the Bezier triangle spanning the volume element, and

$$\mathbf{r}\left(\boldsymbol{\chi}\right) = \sum_{|\mathbf{i}|=n} B_{\mathbf{i}}^{n}\left(\boldsymbol{\chi}\right) \mathbf{C}_{\mathbf{i}}$$

$$\mathrm{d}\mathbf{S} \equiv \frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} = \sum_{|\mathbf{j}|,|\mathbf{k}|=n} \frac{\partial B_{\mathbf{j}}^{n}}{\partial u} \frac{\partial B_{\mathbf{k}}^{n}}{\partial v} \mathbf{C}_{\mathbf{j}} \times \mathbf{C}_{\mathbf{k}} \mathrm{d}u \mathrm{d}v$$

Therefore, the volume of the Bezier tetrahedron is given by

$$\Delta V = \frac{1}{3} \iint_{\mathcal{T}} \left(\sum_{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|=n} B_{\mathbf{i}}^{n} \frac{\partial B_{\mathbf{j}}^{n}}{\partial u} \frac{\partial B_{\mathbf{k}}^{n}}{\partial v} \left(\boldsymbol{\chi}\right) \mathbf{C}_{\mathbf{i}}^{T} \left[\mathbf{C}_{\mathbf{j}} \times \mathbf{C}_{\mathbf{k}}\right] \right) \mathrm{d}u \mathrm{d}v$$

which can be transformed into

$$\Delta V = \sum_{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|=n} \alpha_{\mathbf{ijk}} \mathbf{C}_{\mathbf{i}}^{T} \left[\mathbf{C}_{\mathbf{j}} \times \mathbf{C}_{\mathbf{k}}\right] \tag{6.1}$$

where

$$\alpha_{\mathbf{ijk}} = \frac{1}{3} \iint_{\mathcal{T}} \left(B_{\mathbf{i}}^{n} \frac{\partial B_{\mathbf{j}}^{n}}{\partial u} \frac{\partial B_{\mathbf{k}}^{n}}{\partial v} \right) \mathrm{d}u \mathrm{d}v = \frac{1}{3} \int_{0}^{1} \int_{0}^{1-v} \left(B_{\mathbf{i}}^{n} \frac{\partial B_{\mathbf{j}}^{n}}{\partial u} \frac{\partial B_{\mathbf{k}}^{n}}{\partial v} \right) \mathrm{d}u \mathrm{d}v \tag{6.2}$$

The $\alpha_{\mathbf{ijk}}$ can be computed analytically after the index triplets have been defined

$$\mathbf{i} = i, j, (n - i - j) \tag{6.3}$$

$$\mathbf{j} = k, l, (n - k - l) \tag{6.4}$$

$$\mathbf{k} = m, p, (n - m - p) \tag{6.5}$$

Then the Bezier surface can be written as

$$\mathbf{r}\left(\boldsymbol{\chi}\right) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} B_{i,j,n-i-j}^{n}\left(\boldsymbol{\chi}\right) \mathbf{C}_{i,j,n-i-j}$$

by consequence,

$$\alpha_{\mathbf{ijk}} = \frac{1}{3}\int_0^1\int_0^{1-u}\left(B^n_{i,j,n-i-j}\frac{\partial B^n_{k,l,n-k-l}}{\partial u}\frac{\partial B^n_{m,p,n-m-p}}{\partial v}\right)\mathrm{d}v\mathrm{d}u$$
$$= \frac{1}{3}\int_0^1\int_0^{1-u}\left(B^n_{i,j}\frac{\partial B^n_{k,l}}{\partial u}\frac{\partial B^n_{m,p}}{\partial v}\right)\mathrm{d}v\mathrm{d}u$$

where the third index is dropped from the expression of the Bernstein polynomials because of its redundancy. Expanding the Bezier derivatives,

$$\alpha_{\mathbf{ijk}} = \frac{n^2}{3}\int_0^1\int_0^{1-u} B^n_{i,j}\left(B^{n-1}_{k-1,l} - B^{n-1}_{k,l}\right)\left(B^{n-1}_{m,p-1} - B^{n-1}_{m,p}\right)\mathrm{d}v\mathrm{d}u$$

which becomes after expanding the products

$$\begin{aligned}\alpha_{\mathbf{ijk}} = \frac{n^2}{3}\binom{n}{i,j}\int_0^1\int_0^{1-u}\Bigg[&\binom{n-1}{k-1,l}\binom{n-1}{m,p-1}u^{k+m+i-1}v^{l+j+p-1}(1-u-v)^{3n-i-j-k-l-m-p}\\ &-\binom{n-1}{k-1,l}\binom{n-1}{m,p}u^{k+m+i-1}v^{l+j+p}(1-u-v)^{3n-i-j-k-l-m-p-1}\\ &-\binom{n-1}{k,l}\binom{n-1}{m,p-1}u^{k+m+i}v^{l+j+p-1}(1-u-v)^{3n-i-j-k-l-m-p-1}\\ &+\binom{n-1}{k,l}\binom{n-1}{m,p}u^{k+m+i}v^{l+j+p}(1-u-v)^{3n-i-j-k-l-m-p-2}\Bigg]\mathrm{d}v\mathrm{d}u\end{aligned}$$

Introducing

$$S^b_a = \int_0^1 u^a(1-u)^b\,\mathrm{d}u = \sum_{i=0}^{b}\binom{b}{i}\frac{(-1)^i}{i+a+1} \tag{6.6}$$

and the shortcut

$$|\mathbf{ijk}| = i+j+k+l+m+p \tag{6.7}$$

the coefficients finally become

$$\alpha_{\mathbf{ijk}} = \frac{n^2}{3}\binom{n}{i,j}\left[\binom{n-1}{k-1,l}\binom{n-1}{m,p-1} S_{l+j+p-1}^{3n-|\mathbf{ijk}|} S_{k+m+i-1}^{3n-i-k-m} \right.$$
$$- \binom{n-1}{k-1,l}\binom{n-1}{m,p} S_{l+j+p}^{3n-|\mathbf{ijk}|-1} S_{k+m+i-1}^{3n-i-k-m}$$
$$- \binom{n-1}{k,l}\binom{n-1}{m,p-1} S_{l+j+p-1}^{3n-|\mathbf{ijk}|-1} S_{k+m+i}^{3n-i-k-m-1}$$
$$\left. + \binom{n-1}{k,l}\binom{n-1}{m,p} S_{l+j+p}^{3n-|\mathbf{ijk}|-2} S_{k+m+i}^{3n-i-k-m-1}\right] \tag{6.8}$$

One will also note that the $\alpha_{\mathbf{ijk}}$ are completely independent from the control mesh, and can thus be computed ahead of time. The total volume of a Bezier shape formed by a collection of interconnected Bezier triangles can be found by summing the output of Equation (6.1) over all surface elements:

$$V = \sum_{e=1}^{N_e} \Delta V_e = \sum_{e=1}^{N_e} \sum_{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|=n} \alpha_{\mathbf{ijk}} \mathbf{C}_{\mathbf{i}}^{eT} \left[\mathbf{C}_{\mathbf{j}}^{e} \times \mathbf{C}_{\mathbf{k}}^{e}\right] \tag{6.9}$$

6.1.1.2 Center of mass of a Bezier shape model

The computation of the center of mass of the Bezier shape follows the same principle, assuming that the shape being dealt with is of constant density. The center of mass of a given Bezier tetrahedron is given by

$$\Delta \mathbf{c}_{\mathrm{m}} = \frac{1}{\Delta V} \iiint_{\delta \mathcal{V}} \mathbf{r} \mathrm{d}V$$

If the volume element is a Bezier tetrahedron, one can parametrize the position vector originating from $\mathbf{O}$ as

$$\mathbf{r} = a\mathbf{P}(u,v)$$

As a result,

$$
\begin{aligned}
\mathrm{d}V &= \begin{vmatrix} \frac{\partial \mathbf{r}}{\partial \mathbf{a}} & \frac{\partial \mathbf{r}}{\partial \mathbf{u}} & \frac{\partial \mathbf{r}}{\partial \mathbf{v}} \end{vmatrix} \mathrm{d}a\mathrm{d}u\mathrm{d}v \\
&= a^2 \sum_{|\mathbf{k}|,|\mathbf{l}|,|\mathbf{m}|=n} B^n_{\mathbf{k}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{m}}}{\partial \mathbf{v}} \mathbf{C}^T_{\mathbf{k}} \left(\mathbf{C}_{\mathbf{l}} \times \mathbf{C}_{\mathbf{m}}\right) \mathrm{d}a\mathrm{d}u\mathrm{d}v
\end{aligned}
$$

So

$$
\begin{aligned}
\mathbf{\Delta c}_{\mathrm{m}} &= \frac{1}{\Delta V} \int\limits_0^1 \int\limits_0^1 \int\limits_0^{1-u} \mathbf{r} a^2 \sum_{|\mathbf{j}|,|\mathbf{k}|=n} \sum_{|\mathbf{l}|=n} B^n_{\mathbf{j}} \frac{\partial B^n_{\mathbf{k}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{v}} \mathbf{C}^T_{\mathbf{j}} \left(\mathbf{C}_{\mathbf{k}} \times \mathbf{C}_{\mathbf{l}}\right) \\
&\quad \cdot \mathrm{d}v\mathrm{d}u\mathrm{d}a \qquad (6.10) \\
&= \frac{1}{\Delta V} \int\limits_0^1 \int\limits_0^1 \int\limits_0^{1-u} a^3 \sum_{|\mathbf{i}|=n} B^n_{\mathbf{i}} \mathbf{C}_{\mathbf{i}} \sum_{|\mathbf{j}|,|\mathbf{k}|=n} \sum_{|\mathbf{l}|=n} \\
&\quad \cdot B^n_{\mathbf{j}} \frac{\partial B^n_{\mathbf{k}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{v}} \mathbf{C}^T_{\mathbf{j}} \left(\mathbf{C}_{\mathbf{k}} \times \mathbf{C}_{\mathbf{l}}\right) \mathrm{d}v\mathrm{d}u\mathrm{d}a \qquad (6.11) \\
&= \frac{1}{\Delta V} \int\limits_0^1 \int\limits_0^1 \int\limits_0^{1-u} a^3 \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|=n}} B^n_{\mathbf{i}} \mathbf{C}_{\mathbf{i}} B^n_{\mathbf{j}} \frac{\partial B^n_{\mathbf{k}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{v}} \mathbf{C}^T_{\mathbf{j}} \left(\mathbf{C}_{\mathbf{k}} \times \mathbf{C}_{\mathbf{l}}\right) \\
&\quad \cdot \mathrm{d}v\mathrm{d}u\mathrm{d}a \qquad (6.12) \\
&= \frac{1}{4\Delta V} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|=n}} \int\limits_0^1 \int\limits_0^{1-u} B^n_{\mathbf{i}} B^n_{\mathbf{j}} \frac{\partial B^n_{\mathbf{k}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{v}} \mathbf{C}_{\mathbf{i}} \mathbf{C}^T_{\mathbf{j}} \left(\mathbf{C}_{\mathbf{k}} \times \mathbf{C}_{\mathbf{l}}\right) \mathrm{d}v\mathrm{d}u \qquad (6.13) \\
&= \frac{1}{\Delta V} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|=n}} \gamma_{\mathbf{ijkl}} \mathbf{C}_{\mathbf{i}} \mathbf{C}^T_{\mathbf{j}} \left(\mathbf{C}_{\mathbf{k}} \times \mathbf{C}_{\mathbf{l}}\right) \qquad (6.14)
\end{aligned}
$$

where

$$
\gamma_{\mathbf{ijkl}} = \frac{1}{4} \int\limits_0^1 \int\limits_0^{1-v} \left(B^n_{\mathbf{i}} B^n_{\mathbf{j}} \frac{\partial B^n_{\mathbf{k}}}{\partial u} \frac{\partial B^n_{\mathbf{l}}}{\partial v} \right) \mathrm{d}u\mathrm{d}v \qquad (6.15)
$$

Define the handy shortcut

$$
|\mathbf{ijkl}| = i + j + k + l + m + p + q + r \qquad (6.16)
$$

the closed-form expression of these coefficients can be computed in a way similar to the $\alpha_{\mathbf{ijk}}$:

$$\gamma_{\mathbf{ijkl}} = \frac{n^2}{4} \binom{n}{i, j} \binom{n}{k, l} \int_0^1 \int_0^{1-u} \left[\binom{n-1}{m-1, p} \binom{n-1}{q, r-1} \right.$$

$$\cdot\, u^{i+k+m+q-1} v^{j+l+p+r-1} (1-u-v)^{4n-|\mathbf{ijkl}|}$$

$$- \binom{n-1}{m-1, p} \binom{n-1}{q, r} u^{i+k+m+q-1} v^{j+l+p+r} (1-u-v)^{4n-|\mathbf{ijkl}|-1}$$

$$- \binom{n-1}{m, p} \binom{n-1}{q, r-1} u^{i+k+m+q} v^{j+l+p+r-1} (1-u-v)^{4n-|\mathbf{ijkl}|-1}$$

$$\left. + \binom{n-1}{m, p} \binom{n-1}{q, r} u^{i+k+m+q} v^{j+l+p+r} (1-u-v)^{4n-|\mathbf{ijkl}|-2} \right]$$

$$\cdot\, \mathrm{d}v \mathrm{d}u$$

which becomes after integrating

$$
\begin{aligned}
\gamma_{\mathbf{ijkl}} &= \frac{n^2}{4}\binom{n}{i,j}\binom{n}{k,l}\Bigg[\binom{n-1}{m-1,p}\binom{n-1}{q,r-1}S^{4n-|\mathbf{ijkl}|}_{j+l+p+r-1}S^{4n-i-k-m-q}_{i+k+m+q-1} \\
&- \binom{n-1}{m-1,p}\binom{n-1}{q,r}S^{4n-|\mathbf{ijkl}|-1}_{j+l+p+r}S^{4n-i-k-m-q}_{i+k+m+q-1} \\
&- \binom{n-1}{m,p}\binom{n-1}{q,r-1}S^{4n-|\mathbf{ijkl}|-1}_{j+l+p+r-1}S^{4n-i-k-m-q-1}_{i+k+m+q} \\
&+ \binom{n-1}{m,p}\binom{n-1}{q,r}S^{4n-|\mathbf{ijkl}|-2}_{j+l+p+r}S^{4n-i-k-m-q-1}_{i+k+m+q}\Bigg]
\end{aligned} \tag{6.17}
$$

$$
\begin{aligned}
&= \frac{n^2}{4}\binom{n}{i,j}\binom{n}{k,l}\Bigg[\binom{n-1}{m-1,p} \\
&\cdot\left(\binom{n-1}{q,r-1}S^{4n-|\mathbf{ijkl}|}_{j+l+p+r-1} - \binom{n-1}{q,r}S^{4n-|\mathbf{ijkl}|-1}_{j+l+p+r}\right)S^{4n-i-k-m-q}_{i+k+m+q-1} \\
&- \binom{n-1}{m,p}\left(\binom{n-1}{q,r-1}S^{4n-|\mathbf{ijkl}|-1}_{j+l+p+r-1} - \binom{n-1}{q,r}S^{4n-|\mathbf{ijkl}|-2}_{j+l+p+r}\right) \\
&\cdot S^{4n-i-k-m-q-1}_{i+k+m+q}\Bigg]
\end{aligned} \tag{6.18}
$$

The center of mass of a Bezier shape comprised of Bezier triangles is thus given by

$$
\mathbf{c}_m = \frac{1}{V}\sum_{e=1}^{N_e}\Delta V_e \Delta\mathbf{c}_{m,e} = \frac{1}{V}\sum_{e=1}^{N_e}\sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|=n}}\gamma_{\mathbf{ijkl}}\mathbf{C}^e_{\mathbf{i}}\mathbf{C}^{eT}_{\mathbf{j}}\left(\mathbf{C}^e_{\mathbf{k}}\times\mathbf{C}^e_{\mathbf{l}}\right) \tag{6.19}
$$

6.1.1.3 Inertia of a Bezier shape model

Define $\widetilde{[\mathbf{y}]}$ as the matrix representation of the linear mapping $\mathbf{x}\mapsto\mathbf{y}\times\mathbf{x}$, the inertia tensor of the Bezier tetrahedron about an arbitrary origin $\mathbf{O}$ is given by [68]

$$
[\Delta I]_{\mathbf{O}} = -\iiint_{\delta\mathcal{V}}[\tilde{\mathbf{r}}][\tilde{\mathbf{r}}]\mathrm{d}m
$$

Under the assumption of a constant density ρ,

$$[\Delta I]_{\mathbf{O}} = -\rho \iiint\limits_{\delta\mathcal{V}} [\tilde{\mathbf{r}}][\tilde{\mathbf{r}}]\mathrm{d}V$$

The inertia tensor becomes

$$\begin{aligned}
[\Delta I]_{\mathbf{O}} = -\rho \int_0^1 \int_0^1 \int_0^{1-u} [\tilde{\mathbf{r}}][\tilde{\mathbf{r}}] a^2 \sum_{|\mathbf{k}|,|\mathbf{l}|,|\mathbf{m}|=n} B^n_{\mathbf{k}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{m}}}{\partial \mathbf{v}} \mathbf{C}^T_{\mathbf{k}} \left(\mathbf{C}_{\mathbf{l}} \times \mathbf{C}_{\mathbf{m}}\right) \\
\cdot \mathrm{d}v\mathrm{d}u\mathrm{d}a \\
= -\rho \int_0^1 \int_0^1 \int_0^{1-u} a^2 \sum_{|\mathbf{i}|=n} \sum_{|\mathbf{j}|=n} B^n_{\mathbf{i}} B^n_{\mathbf{j}} [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\mathbf{C}}_b \mathbf{j}] a^2 \\
\cdot \sum_{|\mathbf{k}|,|\mathbf{l}|,|\mathbf{m}|=n} B^n_{\mathbf{k}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{m}}}{\partial \mathbf{v}} \mathbf{C}^T_{\mathbf{k}} \left(\mathbf{C}_{\mathbf{l}} \times \mathbf{C}_{\mathbf{m}}\right) \mathrm{d}v\mathrm{d}u\mathrm{d}a \\
= -\rho \int_0^1 \int_0^1 \int_0^{1-u} a^4 \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|,|\mathbf{m}|=n}} B^n_{\mathbf{i}} B^n_{\mathbf{j}} [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\mathbf{C}}_{\mathbf{j}}] B^n_{\mathbf{k}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{m}}}{\partial \mathbf{v}} \mathbf{C}^T_{\mathbf{k}} \left(\mathbf{C}_{\mathbf{l}} \times \mathbf{C}_{\mathbf{m}}\right) \\
\cdot \mathrm{d}v\mathrm{d}u\mathrm{d}a
\end{aligned}$$

After swapping the sums with the integrals and integrating over a,

$$[\Delta I]_{\mathbf{O}} = -\rho \frac{1}{5} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|=n}} \sum_{|\mathbf{m}|=n} [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\mathbf{C}}_{\mathbf{j}}] \mathbf{C}^T_{\mathbf{k}} \left(\mathbf{C}_{\mathbf{l}} \times \mathbf{C}_{\mathbf{m}}\right)$$

$$\cdot \int_0^1 \int_0^{1-u} B^n_{\mathbf{i}} B^n_{\mathbf{j}} B^n_{\mathbf{k}} \frac{\partial B^n_{\mathbf{l}}}{\partial \mathbf{u}} \frac{\partial B^n_{\mathbf{m}}}{\partial \mathbf{v}} \mathrm{d}v\mathrm{d}u \tag{6.20}$$

$$= \rho \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|=n}} \sum_{|\mathbf{m}|=n} [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\mathbf{C}}_{\mathbf{j}}] \mathbf{C}^T_{\mathbf{k}} \left(\mathbf{C}_{\mathbf{l}} \times \mathbf{C}_{\mathbf{m}}\right) \kappa_{\mathbf{ijklm}} \tag{6.21}$$

where

$$\begin{aligned}
\kappa_{\mathbf{ijklm}} &= -\frac{1}{5}\int\limits_0^1\int\limits_0^{1-u} B_{\mathbf{i}}^n B_{\mathbf{j}}^n B_{\mathbf{k}}^n \frac{\partial B_{\mathbf{l}}^n}{\partial \mathbf{u}} \frac{\partial B_{\mathbf{l}}^n}{\partial \mathbf{v}} \mathrm{d}v\mathrm{d}u \\
&= -\frac{n^2}{5}\binom{n}{i,j}\binom{n}{k,l}\binom{n}{m,p}\int\limits_0^1\int\limits_0^{1-u}\Bigg[\\
&\cdot\binom{n-1}{q-1,r}\binom{n-1}{s,t-1} u^{i+k+m+q+s-1} v^{j+l+p+r+t-1}(1-u-v)^{5n-|\mathbf{ijklm}|} \\
&-\binom{n-1}{q-1,r}\binom{n-1}{s,t} u^{i+k+q+m+s-1} v^{j+l+p+r+t}(1-u-v)^{5n-|\mathbf{ijklm}|-1} \\
&-\binom{n-1}{q,r}\binom{n-1}{s,t-1} u^{i+k+m+q+s} v^{j+l+p+r+t-1}(1-u-v)^{5n-|\mathbf{ijklm}|-1} \\
&+\binom{n-1}{q,r}\binom{n-1}{s,t} u^{i+k+m+q+s} v^{j+l+p+r+t}(1-u-v)^{5n-|\mathbf{ijklm}|-2}\Bigg] \\
&\cdot \mathrm{d}v\mathrm{d}u
\end{aligned}$$

Define

$$\mathbf{i} = i, j, (n-i-j) \tag{6.22}$$

$$\mathbf{j} = k, l, (n-k-l) \tag{6.23}$$

$$\mathbf{k} = m, p, (n-m-p) \tag{6.24}$$

$$\mathbf{l} = q, r, (n-q-r) \tag{6.25}$$

$$\mathbf{m} = s, t, (n-s-t) \tag{6.26}$$

$$|\mathbf{ijklm}| = i+j+k+l+m+p+q+r+s+t \tag{6.27}$$

the final expression of the $\kappa_{\mathbf{ijklm}}$ becomes

$$\kappa_{\mathbf{ijklm}} = -\frac{n^2}{5}\begin{pmatrix} n \\ i,j \end{pmatrix}\begin{pmatrix} n \\ k,l \end{pmatrix}\begin{pmatrix} n \\ m,p \end{pmatrix}\Bigg[\begin{pmatrix} n-1 \\ q-1,r \end{pmatrix}$$

$$\cdot\left(\begin{pmatrix} n-1 \\ s,t-1 \end{pmatrix} S^{5n-|\mathbf{ijklm}|}_{j+l+p+r+t-1} - \begin{pmatrix} n-1 \\ s,t \end{pmatrix} S^{5n-|\mathbf{ijklm}|-1}_{j+l+p+r+t}\right) S^{5n-i-k-m-q-s}_{i+k+m+q+s-1}$$

$$-\begin{pmatrix} n-1 \\ q,r \end{pmatrix}\left(\begin{pmatrix} n-1 \\ s,t-1 \end{pmatrix} S^{5n-|\mathbf{ijklm}|-1}_{j+l+p+r+t-1} - \begin{pmatrix} n-1 \\ s,t \end{pmatrix} S^{5n-|\mathbf{ijklm}|-2}_{j+l+p+r+t}\right)$$

$$\cdot\, S^{5n-i-k-m-q-s-1}_{i+k+m+q+s}\Bigg] \tag{6.28}$$

and the inertia tensor of the complete Bezier shape is given by

$$[I]_{\mathbf{O}} \equiv \sum_{e=1}^{N_e}[\Delta I]_{\mathbf{O},e} = \rho\sum_{e=1}^{N_e}\sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|,|\mathbf{m}|=n}}[\tilde{\mathbf{C}}^e_{\mathbf{i}}][\tilde{\mathbf{C}}^e_{\mathbf{j}}]\mathbf{C}^{eT}_{\mathbf{k}}\left(\mathbf{C}^e_{\mathbf{l}} \times \mathbf{C}^e_{\mathbf{m}}\right)\kappa_{\mathbf{ijklm}} \tag{6.29}$$

6.1.2 Inertia quantities statistics

In what follows, it is assumed that the control mesh of a given shape obeys a normal distribution of mean $\bar{\mathbf{C}}$ and known covariance. In addition, the amplitude of the shape randomness is assumed to be sufficiently small so that the first variation in each control point coordinates satisfies

$$\delta\mathbf{C}_{\mathbf{i}} \equiv \mathbf{C}_{\mathbf{i}} - \bar{\mathbf{C}}_{\mathbf{i}} << \bar{\mathbf{C}}_{\mathbf{i}}$$

in the L2-norm sense.

6.1.2.1 Uncertainty in volume given Gaussian control mesh

The first variation in the *total* volume reads

$$
\begin{aligned}
\delta V &\simeq V - \bar{V} \\
&\equiv \sum_{e=1}^{N_e} \sum_{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|=n} \alpha_{\mathbf{ijk}} \Big(\delta \mathbf{C}_{\mathbf{i}}^{eT} \left[\bar{\mathbf{C}}_{\mathbf{j}}^{e} \times \bar{\mathbf{C}}_{\mathbf{k}}^{e} \right] \\
&\quad + \bar{\mathbf{C}}_{\mathbf{i}}^{eT} \left[\delta \mathbf{C}_{\mathbf{j}}^{e} \times \bar{\mathbf{C}}_{\mathbf{k}}^{e} \right] + \bar{\mathbf{C}}_{\mathbf{i}}^{eT} \left[\bar{\mathbf{C}}_{\mathbf{j}}^{e} \times \delta \mathbf{C}_{\mathbf{k}}^{e} \right] \Big) \\
&= \sum_{e=1}^{N_e} \sum_{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|=n} \alpha_{\mathbf{ijk}} \begin{pmatrix} \mathbf{v}_{\mathbf{jk}}^{e} \\ \mathbf{v}_{\mathbf{ki}}^{e} \\ \mathbf{v}_{\mathbf{ij}}^{e} \end{pmatrix}^T \begin{pmatrix} \delta \mathbf{C}_{\mathbf{i}}^{e} \\ \delta \mathbf{C}_{\mathbf{j}}^{e} \\ \delta \mathbf{C}_{\mathbf{k}}^{e} \end{pmatrix}
\end{aligned}
$$

where $\mathbf{v}_{\mathbf{jk}}^{e} = \bar{\mathbf{C}}_{\mathbf{j}}^{e} \times \bar{\mathbf{C}}_{\mathbf{k}}^{e}$. The variance in the total volume becomes

$$
\begin{aligned}
\sigma_V^2 = &\sum_{e,f=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|, \\ |\mathbf{m}|,|\mathbf{p}|,|\mathbf{l}|=n}} \alpha_{\mathbf{ijk}} \alpha_{\mathbf{lmp}} \\
&\cdot \begin{pmatrix} \mathbf{v}_{\mathbf{jk}}^{e} \\ \mathbf{v}_{\mathbf{ki}}^{e} \\ \mathbf{v}_{\mathbf{ij}}^{e} \end{pmatrix}^T \begin{bmatrix} P_{\mathbf{C}_{\mathbf{i}}^{e}\mathbf{C}_{\mathbf{l}}^{f}} & P_{\mathbf{C}_{\mathbf{i}}^{e}\mathbf{C}_{\mathbf{m}}^{f}} & P_{\mathbf{C}_{\mathbf{i}}^{e}\mathbf{C}_{\mathbf{p}}^{f}} \\ P_{\mathbf{C}_{\mathbf{j}}^{e}\mathbf{C}_{\mathbf{l}}^{f}} & P_{\mathbf{C}_{\mathbf{j}}^{e}\mathbf{C}_{\mathbf{m}}^{f}} & P_{\mathbf{C}_{\mathbf{j}}^{e}\mathbf{C}_{\mathbf{p}}^{f}} \\ P_{\mathbf{C}_{\mathbf{k}}^{e}\mathbf{C}_{\mathbf{l}}^{f}} & P_{\mathbf{C}_{\mathbf{k}}^{e}\mathbf{C}_{\mathbf{m}}^{f}} & P_{\mathbf{C}_{\mathbf{k}}^{e}\mathbf{C}_{\mathbf{p}}^{f}} \end{bmatrix} \begin{pmatrix} \mathbf{v}_{\mathbf{mp}}^{f} \\ \mathbf{v}_{\mathbf{pl}}^{f} \\ \mathbf{v}_{\mathbf{lm}}^{f} \end{pmatrix}
\end{aligned} \tag{6.30}
$$

where $P_{\mathbf{C}_{\mathbf{i}}^{e}\mathbf{C}_{\mathbf{l}}^{f}} = \mathrm{E}\left(\delta \mathbf{C}_{\mathbf{i}}^{e} \delta \mathbf{C}_{\mathbf{l}}^{fT} \right)$ captures the uncertainty in the considered control points.

6.1.2.2 Uncertainty in center of mass given Gaussian control mesh

A first order expansion of the center of mass expression about the mean value of the control points such that $\bar{\mathbf{c}}_m = \mathbf{0}$ gives

$$
\begin{aligned}
\delta \mathbf{c}_m &\equiv \mathbf{c}_m - \bar{\mathbf{c}}_m \\
&\simeq \frac{1}{\bar{V}} \sum_{e=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|=n}} \gamma_{\mathbf{ijkl}} \Big[\bar{\mathbf{C}}^e_{\mathbf{i}} \bar{\mathbf{C}}^{eT}_{\mathbf{j}} \left(\bar{\mathbf{C}}^e_{\mathbf{k}} \times \delta \mathbf{C}^e_{\mathbf{l}} + \delta \mathbf{C}^e_{\mathbf{k}} \times \bar{\mathbf{C}}^e_{\mathbf{l}} \right) \\
&+ \left(\bar{\mathbf{C}}^e_{\mathbf{i}} \delta \mathbf{C}^{eT}_{\mathbf{j}} + \delta \mathbf{C}^e_{\mathbf{i}} \bar{\mathbf{C}}^{eT}_{\mathbf{j}} \right) \bar{\mathbf{C}}^e_{\mathbf{k}} \times \bar{\mathbf{C}}^e_{\mathbf{l}} \Big] - \frac{\delta V}{\bar{V}_t} \underbrace{\bar{\mathbf{c}}_m}_{\mathbf{0}} \\
&= \frac{1}{\bar{V}} \sum_{e=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|=n}} \gamma_{\mathbf{ijkl}} \Big[\bar{\mathbf{C}}^e_{\mathbf{i}} \left(\mathbf{v}^{eT}_{\mathbf{jk}} \delta \mathbf{C}^e_{\mathbf{l}} + \mathbf{v}^{eT}_{\mathbf{lj}} \delta \mathbf{C}^e_{\mathbf{k}} \right) \\
&+ \left(\bar{\mathbf{C}}^e_{\mathbf{i}} \delta \mathbf{C}^{eT}_{\mathbf{j}} + \delta \mathbf{C}^e_{\mathbf{i}} \bar{\mathbf{C}}^{eT}_{\mathbf{j}} \right) \mathbf{v}^e_{\mathbf{kl}} \Big] \\
&\simeq \frac{1}{\bar{V}} \sum_{e=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|=n}} \gamma_{\mathbf{ijkl}} \begin{bmatrix} \bar{\mathbf{C}}^{eT}_{\mathbf{j}} \mathbf{v}^e_{\mathbf{kl}} I_3 \\ \mathbf{v}^e_{\mathbf{kl}} \bar{\mathbf{C}}^{eT}_{\mathbf{i}} \\ \mathbf{v}^e_{\mathbf{lj}} \bar{\mathbf{C}}^{eT}_{\mathbf{i}} \\ \mathbf{v}^e_{\mathbf{jk}} \bar{\mathbf{C}}^{eT}_{\mathbf{i}} \end{bmatrix}^T \begin{pmatrix} \delta \mathbf{C}^e_{\mathbf{i}} \\ \delta \mathbf{C}^e_{\mathbf{j}} \\ \delta \mathbf{C}^e_{\mathbf{k}} \\ \delta \mathbf{C}^e_{\mathbf{l}} \end{pmatrix}
\end{aligned}
$$

The outer product of the center of mass deviation reads

$$
\begin{aligned}
\delta \mathbf{c}_m \delta \mathbf{c}_m^T = \frac{1}{\bar{V}^2} \sum_{e,f=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\|\mathbf{l}|,|\mathbf{m}|,|\mathbf{p}|,=n\\|\mathbf{q}|,|\mathbf{r}|}} \gamma_{\mathbf{ijkl}} \gamma_{\mathbf{mpqr}} \begin{bmatrix} \bar{\mathbf{C}}^{eT}_{\mathbf{j}} \mathbf{v}^e_{\mathbf{kl}} I_3 \\ \mathbf{v}^e_{\mathbf{kl}} \bar{\mathbf{C}}^{eT}_{\mathbf{i}} \\ \mathbf{v}^e_{\mathbf{lj}} \bar{\mathbf{C}}^{eT}_{\mathbf{i}} \\ \mathbf{v}^e_{\mathbf{jk}} \bar{\mathbf{C}}^{eT}_{\mathbf{i}} \end{bmatrix}^T \\
\cdot \begin{bmatrix} \delta \mathbf{C}^e_{\mathbf{i}} \delta \mathbf{C}^{fT}_{\mathbf{m}} & \delta \mathbf{C}^e_{\mathbf{i}} \delta \mathbf{C}^{fT}_{\mathbf{p}} & \delta \mathbf{C}^e_{\mathbf{i}} \delta \mathbf{C}^{fT}_{\mathbf{q}} & \delta \mathbf{C}^e_{\mathbf{i}} \delta \mathbf{C}^{fT}_{\mathbf{r}} \\ \delta \mathbf{C}^e_{\mathbf{j}} \delta \mathbf{C}^{fT}_{\mathbf{m}} & \delta \mathbf{C}^e_{\mathbf{j}} \delta \mathbf{C}^{fT}_{\mathbf{p}} & \delta \mathbf{C}^e_{\mathbf{j}} \delta \mathbf{C}^{fT}_{\mathbf{q}} & \delta \mathbf{C}^e_{\mathbf{j}} \delta \mathbf{C}^{fT}_{\mathbf{r}} \\ \delta \mathbf{C}^e_{\mathbf{k}} \delta \mathbf{C}^{fT}_{\mathbf{m}} & \delta \mathbf{C}^e_{\mathbf{k}} \delta \mathbf{C}^{fT}_{\mathbf{p}} & \delta \mathbf{C}^e_{\mathbf{k}} \delta \mathbf{C}^{fT}_{\mathbf{q}} & \delta \mathbf{C}^e_{\mathbf{k}} \delta \mathbf{C}^{fT}_{\mathbf{r}} \\ \delta \mathbf{C}^e_{\mathbf{l}} \delta \mathbf{C}^{fT}_{\mathbf{m}} & \delta \mathbf{C}^e_{\mathbf{l}} \delta \mathbf{C}^{fT}_{\mathbf{p}} & \delta \mathbf{C}^e_{\mathbf{l}} \delta \mathbf{C}^{fT}_{\mathbf{q}} & \delta \mathbf{C}^e_{\mathbf{l}} \delta \mathbf{C}^{fT}_{\mathbf{r}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{C}}^{fT}_{\mathbf{p}} \mathbf{v}^f_{\mathbf{qr}} I_3 \\ \mathbf{v}^f_{\mathbf{qr}} \bar{\mathbf{C}}^{fT}_{\mathbf{m}} \\ \mathbf{v}^f_{\mathbf{rp}} \bar{\mathbf{C}}^{fT}_{\mathbf{m}} \\ \mathbf{v}^f_{\mathbf{pq}} \bar{\mathbf{C}}^{fT}_{\mathbf{m}} \end{bmatrix} \quad (6.31)
\end{aligned}
$$

The covariance in the center of mass position is finally given by

$$P_{\mathbf{c}_m} = \frac{1}{\bar{V}_t^2} \sum_{e,f=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|, \\ |\mathbf{l}|,|\mathbf{m}|,|\mathbf{p}|,=n \\ |\mathbf{q}|,|\mathbf{r}|}} \gamma_{\mathbf{ijkl}} \gamma_{\mathbf{mpqr}}$$
$$\cdot \begin{bmatrix} \bar{\mathbf{C}}_{\mathbf{j}}^{eT} \mathbf{v}_{\mathbf{kl}}^e I_3 \\ \mathbf{v}_{\mathbf{kl}}^e \bar{\mathbf{C}}_{\mathbf{i}}^{eT} \\ \mathbf{v}_{\mathbf{lj}}^e \bar{\mathbf{C}}_{\mathbf{i}}^{eT} \\ \mathbf{v}_{\mathbf{jk}}^e \bar{\mathbf{C}}_{\mathbf{i}}^{eT} \end{bmatrix}^T \begin{bmatrix} P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{m}}^f} & P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{q}}^f} & P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{r}}^f} \\ P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{m}}^f} & P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{q}}^f} & P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{r}}^f} \\ P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{m}}^f} & P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{q}}^f} & P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{r}}^f} \\ P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{m}}^f} & P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{q}}^f} & P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{r}}^f} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{C}}_{\mathbf{p}}^{fT} \mathbf{v}_{\mathbf{qr}}^f I_3 \\ \mathbf{v}_{\mathbf{qr}}^f \bar{\mathbf{C}}_{\mathbf{m}}^{fT} \\ \mathbf{v}_{\mathbf{rp}}^f \bar{\mathbf{C}}_{\mathbf{m}}^{fT} \\ \mathbf{v}_{\mathbf{pq}}^f \bar{\mathbf{C}}_{\mathbf{m}}^{fT} \end{bmatrix} \tag{6.32}$$

6.1.2.3 Inertia's second moments about the mean

Contrary to the first-order mapping readily providing the second moment about the mean of the volume and center of mass, respectively in the form of a standard-deviation and covariance matrix, the inertia tensor requires more caution. Instead of directly extracting the statistical moments of the inertia tensor, one should operate on its parametrization in terms of inertia moments and attitude parameters describing the orientation of the principal axes. Starting from $M = \rho V$, the inertia tensor with respect to the center of mass is given by the parallel-axis formula [68]:

$$[I]_{\mathbf{c}_m} = [I]_{\mathbf{O}} - M[\widetilde{\mathbf{c}_m}][\widetilde{\mathbf{c}_m}]^T$$

where

$$[I]_{\mathbf{O}} = \rho \sum_{e=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}| \\ |\mathbf{l}|,|\mathbf{m}|=n}} \kappa_{\mathbf{ijklm}} [\widetilde{\mathbf{C}_{\mathbf{i}}^e}][\widetilde{\mathbf{C}_{\mathbf{j}}^e}] \mathbf{C}_{\mathbf{k}}^{eT} \left(\mathbf{C}_{\mathbf{l}}^e \times \mathbf{C}_{\mathbf{m}}^e\right)$$

The control mesh about which the linearization is computed can have its coordinates expressed with respect to its barycenter, so that $\mathbf{c}_m = \mathbf{0}$. This way, one can directly operate on $[I]_{\mathbf{O}}$ without linearizing the contribution of the parallel axis theorem. This manuscript therefore uses $[I] \equiv [I]_{\mathbf{O}} = [I]_{\mathbf{c}_m}$ to represent the barycentered inertia tensor from now on. Following Dobrovolskis' notation [103], the principal dimensions of the equivalent ellipsoid are given by

$$a = \sqrt{\frac{5\left(B + C - A\right)}{2M}}$$
$$b = \sqrt{\frac{5\left(A + C - B\right)}{2M}}$$
$$c = \sqrt{\frac{5\left(A + B - C\right)}{2M}}$$

where the three moments of inertia A, B and C are the eigenvalues of the inertia tensor sorted in ascending order and M the total mass of the body. Defining the inertia tensor's parametrization as

$$\mathbf{I} = \begin{pmatrix} I_{xx} \\ I_{yy} \\ I_{zz} \\ I_{xy} \\ I_{xz} \\ I_{yz} \end{pmatrix} \tag{6.33}$$

this section is thus concerned with the expression of the first-order variations and second moment about the means of $\mathbf{I}$, the principal dimensions, moments and parametrization of the principal frame attitude relative to the current body frame.

Principal dimensions, moments, and inertia parametrization

First variations

The first variation in the ellipsoid axes dimensions is given by

$$\delta a = \frac{5}{4aM} \begin{pmatrix} -1 \\ 1 \\ 1 \\ -\frac{B+C-A}{M} \end{pmatrix}^T \begin{pmatrix} \delta A \\ \delta B \\ \delta C \\ \delta M \end{pmatrix} = \frac{\partial a}{\partial \mathbf{M}} \delta \mathbf{M} \tag{6.34}$$

$$\delta b = \frac{5}{4bM} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -\frac{A+C-B}{M} \end{pmatrix}^T \begin{pmatrix} \delta A \\ \delta B \\ \delta C \\ \delta M \end{pmatrix} = \frac{\partial b}{\partial \mathbf{M}} \delta \mathbf{M} \tag{6.35}$$

$$\delta c = \frac{5}{4cM} \begin{pmatrix} 1 \\ 1 \\ -1 \\ -\frac{A+B-C}{M} \end{pmatrix}^T \begin{pmatrix} \delta A \\ \delta B \\ \delta C \\ \delta M \end{pmatrix} = \frac{\partial c}{\partial \mathbf{M}} \delta \mathbf{M} \tag{6.36}$$

which can be conveniently expressed as

$$\mathbf{d} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \ \mathbf{M} = \begin{pmatrix} \mathbf{D} \\ M \end{pmatrix}, \ \mathbf{D} = \begin{pmatrix} A \\ B \\ C \end{pmatrix} \tag{6.37}$$

and

The first variations in A, B, C and M can be expressed through

$$\begin{aligned} \delta \mathbf{d} &= \frac{\partial \mathbf{d}}{\partial \mathbf{M}} \delta \mathbf{M} \\ &= \frac{\partial \mathbf{d}}{\partial \mathbf{M}} \begin{pmatrix} \delta \mathbf{D} \\ \delta M \end{pmatrix} \end{aligned}$$

The variation in the mass is simply given by

$$\delta M = \rho \delta V \tag{6.38}$$

The first variation in the moments can be obtained from the manipulation of the inertia tensor. From the definition of the principal axes $\{\hat{u}_i\}_{i=A,B,C}$ and moments $\{\lambda_i\}_{i=A,B,C}$

$$[I]\hat{u}_i = \lambda_i \hat{u}_i \tag{6.39}$$

Taking the first variation of this equation yields

$$[\delta I]\hat{u}_i + [I]\delta\hat{u}_i = \delta\lambda_i \hat{u}_i + \lambda_i \delta\hat{u}_i \tag{6.40}$$

Defining the matrix U_i through the relationship $[I]\hat{u}_i = U_i\mathbf{I}$ and taking the dot product of Equation (6.40) with $\hat{u}_i$ cancels out the first variation in the eigenvector to leave

$$\delta\lambda_i = \frac{1}{\hat{u}_i^T\hat{u}_i}\hat{u}_i^T U_i \delta\mathbf{I} \tag{6.41}$$

Leveraging the unit-norm of the eigenvectors and repeating the process for the other two eigenvalues leaves

$$\delta\mathbf{D} = \begin{bmatrix} \hat{u}_A^T U_A \\ \hat{u}_B^T U_B \\ \hat{u}_C^T U_C \end{bmatrix} \delta\mathbf{I} \tag{6.42}$$

so the sought-for partial is

$$\frac{\partial\mathbf{D}}{\partial\mathbf{I}} = \begin{bmatrix} \hat{u}_A^T U_A \\ \hat{u}_B^T U_B \\ \hat{u}_C^T U_C \end{bmatrix} \tag{6.43}$$

Since

$$I_{rq} = \hat{e}_r^T [I]_{\mathbf{c}_m} \hat{e}_q$$

with $r, q \in \{x, y, z\} \times \{x, y, z\}$ such that

$$\hat{e}_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \ \hat{e}_y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \hat{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

the first variation of one of the inertia tensor's individual components is given by

$$\delta I_{rq} = \rho \sum_{e=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|,|\mathbf{m}|=n}} \kappa_{\mathbf{ijklm}} \bigg(- \mathbf{C}_{\mathbf{k}}^T \left(\mathbf{C}_{\mathbf{l}} \times \mathbf{C}_{\mathbf{m}}\right) \cdot \left[\hat{e}_q^T [\tilde{\mathbf{C}}_{\mathbf{j}}][\tilde{\hat{e}}_r]\delta\mathbf{C}_{\mathbf{i}} + \hat{e}_r^T [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\hat{e}}_q]\delta\mathbf{C}_{\mathbf{j}}\right]$$

$$+ \hat{e}_r^T [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\mathbf{C}}_{\mathbf{j}}]\hat{e}_q \left[\left(\mathbf{C}_{\mathbf{l}} \times \mathbf{C}_{\mathbf{m}}\right)^T \delta\mathbf{C}_{\mathbf{k}} - \mathbf{C}_{\mathbf{k}}^T [\tilde{\mathbf{C}}_{\mathbf{m}}]\delta\mathbf{C}_{\mathbf{l}} + \mathbf{C}_{\mathbf{k}}^T [\tilde{\mathbf{C}}_{\mathbf{l}}]\delta\mathbf{C}_{\mathbf{m}}\right] \tag{6.44}$$

$$= \rho \sum_{e=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|,|\mathbf{m}|=n}} \kappa_{\mathbf{ijklm}} \begin{pmatrix} -\mathbf{C}_{\mathbf{k}}^T \mathbf{v}_{\mathbf{lm}} [\tilde{\hat{e}}_r][\tilde{\mathbf{C}}_{\mathbf{j}}]\hat{e}_q \\ -\mathbf{C}_{\mathbf{k}}^T \mathbf{v}_{\mathbf{lm}} [\tilde{\hat{e}}_q][\tilde{\mathbf{C}}_{\mathbf{i}}]\hat{e}_r \\ \hat{e}_r^T [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\mathbf{C}}_{\mathbf{j}}]\hat{e}_q \mathbf{v}_{\mathbf{lm}} \\ \hat{e}_r^T [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\mathbf{C}}_{\mathbf{j}}]\hat{e}_q \mathbf{v}_{\mathbf{mk}} \\ \hat{e}_r^T [\tilde{\mathbf{C}}_{\mathbf{i}}][\tilde{\mathbf{C}}_{\mathbf{j}}]\hat{e}_q \mathbf{v}_{\mathbf{kl}} \end{pmatrix}^T \begin{pmatrix} \delta\mathbf{C}_{\mathbf{i}} \\ \delta\mathbf{C}_{\mathbf{j}} \\ \delta\mathbf{C}_{\mathbf{k}} \\ \delta\mathbf{C}_{\mathbf{l}} \\ \delta\mathbf{C}_{\mathbf{m}} \end{pmatrix} \tag{6.45}$$

$$= \rho \sum_{e=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|,|\mathbf{m}|=n}} \kappa_{\mathbf{ijklm}} \mathbf{L}_{rq}^T \begin{pmatrix} \delta\mathbf{C}_{\mathbf{i}} \\ \delta\mathbf{C}_{\mathbf{j}} \\ \delta\mathbf{C}_{\mathbf{k}} \\ \delta\mathbf{C}_{\mathbf{l}} \\ \delta\mathbf{C}_{\mathbf{m}} \end{pmatrix} \tag{6.46}$$

Therefore

$$\delta\mathbf{I} = \rho \sum_{e=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|,|\mathbf{m}|=n}} \kappa_{\mathbf{ijklm}} \begin{bmatrix} \mathbf{L}_{xx}^T \\ \mathbf{L}_{yy}^T \\ \mathbf{L}_{zz}^T \\ \mathbf{L}_{xy}^T \\ \mathbf{L}_{xz}^T \\ \mathbf{L}_{yz}^T \end{bmatrix} \begin{pmatrix} \delta\mathbf{C}_{\mathbf{i}} \\ \delta\mathbf{C}_{\mathbf{j}} \\ \delta\mathbf{C}_{\mathbf{k}} \\ \delta\mathbf{C}_{\mathbf{l}} \\ \delta\mathbf{C}_{\mathbf{m}} \end{pmatrix} \tag{6.47}$$

Finally, the first variation in the mass is given by

$$\delta M = \rho\delta V = \rho \sum_{e=1}^{N_e} \sum_{|\mathbf{i}|,|\mathbf{j}|,\mathbf{k}|=n} \alpha_{\mathbf{ijk}} \begin{pmatrix} \mathbf{v}_{\mathbf{jk}}^e \\ \mathbf{v}_{\mathbf{ki}}^e \\ \mathbf{v}_{\mathbf{ij}}^e \end{pmatrix}^T \begin{pmatrix} \delta\mathbf{C}_{\mathbf{i}}^e \\ \delta\mathbf{C}_{\mathbf{j}}^e \\ \delta\mathbf{C}_{\mathbf{k}}^e \end{pmatrix} \tag{6.48}$$

Second moments about the mean

The sought-for second moment about the mean of the principal dimensions is given by

$$P_{\mathbf{d}} = \mathrm{E}\left(\delta \mathbf{d} \delta \mathbf{d}^T\right) = \left[\frac{\partial \mathbf{d}}{\partial \mathbf{M}}\right] P_{\mathbf{M}} \left[\frac{\partial \mathbf{d}}{\partial \mathbf{M}}\right]^T \tag{6.49}$$

with

$$P_{\mathbf{M}} = \mathrm{E}\left(\delta \mathbf{M} \delta \mathbf{M}^T\right) = \begin{bmatrix} P_{\mathbf{D}} & P_{M\mathbf{D}} \\ P_{M\mathbf{D}}^T & \sigma_M^2 \end{bmatrix} \tag{6.50}$$

$$P_{\mathbf{D}} = \mathrm{E}\left(\delta \mathbf{D} \delta \mathbf{D}^T\right) = \left[\frac{\partial \mathbf{D}}{\partial \mathbf{I}}\right] P_{\mathbf{I}} \left[\frac{\partial \mathbf{D}}{\partial \mathbf{I}}\right]^T \tag{6.51}$$

$$P_{M\mathbf{D}} = \mathrm{E}\left(\delta M \delta \mathbf{D}\right) = \left[\frac{\partial \mathbf{D}}{\partial \mathbf{I}}\right] P_{M\mathbf{I}} \tag{6.52}$$

$$P_M = \rho^2 \sigma_V^2 \tag{6.53}$$

$$\tag{6.54}$$

where $P_{\mathbf{M}}$ holds the covariance of the principal moments. The covariance $P_{\mathbf{I}}$ and correlation matrix $P_{M\mathbf{I}}$ are given by

$$P_{\mathbf{I}} = \rho^2 \sum_{e,f=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|,|\mathbf{m}|,|\mathbf{p}|\\ |\mathbf{q}|,|\mathbf{r}|,|\mathbf{s}|,\\ |\mathbf{t}|=n}} \kappa_{\mathbf{ijklm}} \kappa_{\mathbf{pqrst}}$$

$$\cdot \begin{bmatrix} \mathbf{L}_{xx}^T \\ \mathbf{L}_{yy}^T \\ \mathbf{L}_{zz}^T \\ \mathbf{L}_{xy}^T \\ \mathbf{L}_{xz}^T \\ \mathbf{L}_{yz}^T \end{bmatrix}_{e,\mathbf{ijklm}} \begin{pmatrix} P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{q}}^f} & P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{r}}^f} & P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{s}}^f} & P_{\mathbf{C}_{\mathbf{i}}^e \mathbf{C}_{\mathbf{t}}^f} \\ P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{q}}^f} & P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{r}}^f} & P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{s}}^f} & P_{\mathbf{C}_{\mathbf{j}}^e \mathbf{C}_{\mathbf{t}}^f} \\ P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{q}}^f} & P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{r}}^f} & P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{s}}^f} & P_{\mathbf{C}_{\mathbf{k}}^e \mathbf{C}_{\mathbf{t}}^f} \\ P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{q}}^f} & P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{r}}^f} & P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{s}}^f} & P_{\mathbf{C}_{\mathbf{l}}^e \mathbf{C}_{\mathbf{t}}^f} \\ P_{\mathbf{C}_{\mathbf{m}}^e \mathbf{C}_{\mathbf{p}}^f} & P_{\mathbf{C}_{\mathbf{m}}^e \mathbf{C}_{\mathbf{q}}^e} & P_{\mathbf{C}_{\mathbf{m}}^e \mathbf{C}_{\mathbf{r}}^f} & P_{\mathbf{C}_{\mathbf{m}}^e \mathbf{C}_{\mathbf{s}}^r} & P_{\mathbf{C}_{\mathbf{m}}^e \mathbf{C}_{\mathbf{t}}^f} \end{pmatrix} \begin{bmatrix} \mathbf{L}_{xx}^T \\ \mathbf{L}_{yy}^T \\ \mathbf{L}_{zz}^T \\ \mathbf{L}_{xy}^T \\ \mathbf{L}_{xz}^T \\ \mathbf{L}_{yz}^T \end{bmatrix}_{f,\mathbf{pqrst}}^T \tag{6.55}$$

and

$$P_{M\mathbf{I}} \equiv \rho^2 \sum_{e,f=1}^{N_e} \sum_{\substack{|\mathbf{i}|,|\mathbf{j}|,|\mathbf{k}|,\\ |\mathbf{l}|,|\mathbf{m}|,|\mathbf{p}|,\\ |\mathbf{q}|,|\mathbf{r}|=n}} \kappa_{\mathbf{ijklm}} \alpha_{\mathbf{pqr}}$$

$$\cdot \begin{bmatrix} \mathbf{L}_{xx}^T \\ \mathbf{L}_{yy}^T \\ \mathbf{L}_{zz}^T \\ \mathbf{L}_{xy}^T \\ \mathbf{L}_{xz}^T \\ \mathbf{L}_{yz}^T \end{bmatrix}_{e,\mathbf{ijklm}} \begin{bmatrix} P_{\mathbf{C}_\mathbf{i}^e \mathbf{C}_\mathbf{p}^f} & P_{\mathbf{C}_\mathbf{i}^e \mathbf{C}_\mathbf{q}^f} & P_{\mathbf{C}_\mathbf{i}^e \mathbf{C}_\mathbf{r}^f} \\ P_{\mathbf{C}_\mathbf{j}^e \mathbf{C}_\mathbf{p}^f} & P_{\mathbf{C}_\mathbf{j}^e \mathbf{C}_\mathbf{q}^f} & P_{\mathbf{C}_\mathbf{j}^e \mathbf{C}_\mathbf{r}^f} \\ P_{\mathbf{C}_\mathbf{k}^e \mathbf{C}_\mathbf{p}^f} & P_{\mathbf{C}_\mathbf{k}^e \mathbf{C}_\mathbf{q}^f} & P_{\mathbf{C}_\mathbf{k}^e \mathbf{C}_\mathbf{r}^f} \\ P_{\mathbf{C}_\mathbf{l}^e \mathbf{C}_\mathbf{p}^f} & P_{\mathbf{C}_\mathbf{l}^e \mathbf{C}_\mathbf{q}^f} & P_{\mathbf{C}_\mathbf{l}^e \mathbf{C}_\mathbf{r}^f} \\ P_{\mathbf{C}_\mathbf{m}^e \mathbf{C}_\mathbf{p}^f} & P_{\mathbf{C}_\mathbf{m}^e \mathbf{C}_\mathbf{q}^f} & P_{\mathbf{C}_\mathbf{m}^e \mathbf{C}_\mathbf{r}^f} \end{bmatrix} \begin{pmatrix} \mathbf{v}_{\mathbf{qr}}^f \\ \mathbf{v}_{\mathbf{rp}}^f \\ \mathbf{v}_{\mathbf{pq}}^f \end{pmatrix} \tag{6.56}$$

Principal axes **First variation** The three orthogonal unit-norm eigenvectors expressed in the body-frame ($\mathcal{B}$) $\hat{e}_A$, $\hat{e}_B$ and $\hat{e}_C$ associated with each of the principal moments A, B, C define the body-frame to principal-frame ($\mathcal{P}$) Direction Cosine Matrix (DCM) $[\mathcal{PB}]$ through

$$[\mathcal{PB}] = \begin{bmatrix} \hat{e}_A^T \\ \hat{e}_B^T \\ \hat{e}_C^T \end{bmatrix} \tag{6.57}$$

From the definition of the principal frame,

$$[I] = [\mathcal{BP}][D][\mathcal{PB}] \tag{6.58}$$

where

$$[D] = \begin{pmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{pmatrix} \tag{6.59}$$

Introducing a variation in the shape modifies the definition of the principal axes and moments. Letting primed quantities denote post-variation values and frames, Equation (6.58) become

$$[I] + [\delta I] = [\mathcal{BP}][\mathcal{PP}'] \left([D] + \delta[D]\right) [\mathcal{P}'\mathcal{P}][\mathcal{PB}] \tag{6.60}$$

Parametrize $[\mathcal{PB}]$ in terms of the Modified Rodrigues Parameter (MRP) set $\boldsymbol{\sigma}$ [68], the incremental DCM $[\mathcal{P}'\mathcal{P}]$ is given by

$$[\mathcal{P}'\mathcal{P}] \simeq I_3 - 4[\widetilde{\delta\boldsymbol{\sigma}}] \tag{6.61}$$

By expanding out Equation (6.60), discarding the zero-th and second order terms, and taking its left and right dot product with $\hat{e}_r$ and $\hat{e}_q$ with $r, q \in \{x, y, z\} \times \{x, y, z\}$, one can get the following scalar equation

$$\hat{e}_r^T[\widetilde{\delta\boldsymbol{\sigma}}][D]\hat{e}_q - \hat{e}_r^T[D][\widetilde{\delta\boldsymbol{\sigma}}]\hat{e}_q = \frac{1}{4}\left(\hat{e}_r^T[\mathcal{PB}][\delta I][\mathcal{BP}]\hat{e}_q - \hat{e}_r^T[\delta D]\hat{e}_q\right) \tag{6.62}$$

Use the anti-commutative property of the cross-product matrix and define $\hat{f}_r = [\mathcal{BP}]\hat{e}_r$ and $\hat{f}_q = [\mathcal{BP}]\hat{e}_q$ such that

$$\hat{e}_r^T\left([D][\widetilde{\hat{e}_q}] - [\widetilde{[D]\hat{e}_q}]\right)\delta\boldsymbol{\sigma} = \frac{1}{4}\left(\hat{f}_r^T[\delta I]\hat{f}_q - \hat{e}_r^T[\delta D]\hat{e}_q\right) \tag{6.63}$$

Finally, define $\mathbf{J}_r^q$ such that $\hat{f}_r^T[\delta I]\hat{f}_q = \mathbf{J}_r^{qT}\delta\mathbf{I}$ to obtain

$$H_i\delta\boldsymbol{\sigma} = \mathbf{V}_i^T\delta\mathbf{I} \tag{6.64}$$

with $H_i = \hat{e}_r^T\left([D][\widetilde{\hat{e}_q}] - [\widetilde{[D]\hat{e}_q}]\right)$ and $\mathbf{V}_i^T = \frac{1}{4}\left(\mathbf{J}_r^{qT} - \delta_r^q\hat{e}_r^T\frac{\partial\mathbf{D}}{\partial\mathbf{I}}\right)$

where δ_r^q is the Kronecker delta. Accounting for all nine combinations of $\hat{e}_r, \hat{e}_q$ and stacking up the nine consecutive equations,

$$\begin{pmatrix} \vdots \\ H_i \\ \vdots \end{pmatrix}\delta\boldsymbol{\sigma} = \begin{bmatrix} \vdots \\ \mathbf{V}_i^T \\ \vdots \end{bmatrix}\delta\mathbf{I} \tag{6.65}$$

Define

$$H = \begin{pmatrix} \vdots \\ H_i \\ \vdots \end{pmatrix} \tag{6.66}$$

$$V = \begin{bmatrix} \vdots \\ \mathbf{V}_i^T \\ \vdots \end{bmatrix} \tag{6.67}$$

$$G \equiv \left(H^T H\right)^{-1} H^T V \tag{6.68}$$

the first variation in the MRP set orienting the principal axes is finally given by

$$\delta\boldsymbol{\sigma} = G\delta\mathbf{I} \tag{6.69}$$

The least square solution presented in Equation 6.69 is not an approximation of $\delta\boldsymbol{\sigma}$, because six of the nine combinations of $\hat{e}_r, \hat{e}_q$ are linear combinations of the three combinations that actually yield linearly independent equations. The least-square approach to extracting $\delta\boldsymbol{\sigma}$ merely facilitates implementation. It must be noted that the matrix $H^T H$ becomes singular when two of the inertia moments of the reference shape become equal, which corresponds to an infinity of possible principal frames definitions.

Covariance The covariance in the MRP set parametrizing $[\mathcal{P}'\mathcal{P}]$ is thus given by

$$P_{\boldsymbol{\sigma}} = GP_{\mathbf{I}}G^T \tag{6.70}$$

6.2 Results

The methods developed in this chapter are demonstrated on a perturbed first-order Bezier - hence polyhedron - shape model, where each control point $\mathbf{C_i}$ was assigned with a self-covariance of the form

$$P_{\mathbf{C_i}\mathbf{C_i}} = \sigma^2 \left(\hat{n}_{\mathbf{i}}\hat{n}_{\mathbf{i}}^T + \epsilon\left[\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T\right]\right) \tag{6.71}$$

such that $\epsilon << 1$, $\hat{n}_{\mathbf{i}}$ being the outward surface normal at $\mathbf{C_i}$. This direction was obtained by simply averaging the normals at this point evaluated across the different patches owning this point. The surface tangents $\hat{e}_1$ and $\hat{e}_2$ were chosen such that $\hat{n}_{\mathbf{i}}$, $\hat{e}_1$ and $\hat{e}_2$ form an orthonormal basis. The uncertainty in the control point was thus prescribed to be mostly along their normal. In addition, the control points were correlated with each other through the correlation matrix

$$P_{\mathbf{C_i}\mathbf{C_j}} = \sigma^2 e^{-\frac{\|\mathbf{C_i}-\mathbf{C_j}\|^2}{l^2}} \hat{n}_{\mathbf{i}} \hat{n}_{\mathbf{j}}^T \tag{6.72}$$

This correlation matrix was automatically set to zero should $\|\mathbf{C_i} - \mathbf{C_j}\|$ become larger than $3l$. The correlation length l plays a similar role as in [63].

The inertia statistics computed with the analytical model were compared to the results of a Monte-Carlo simulation featuring 10,000 runs. In each run, the baseline shape model had its control mesh coordinates perturbed by a deviation sampled from the multivariate Gaussian distribution of zero mean and covariance given by

$$P_{\mathbf{CC}} = \begin{bmatrix} P_{\mathbf{C}_0\mathbf{C}_0} & P_{\mathbf{C}_0\mathbf{C}_1} & \dots & P_{\mathbf{C}_0\mathbf{C}_{N-1}} \\ P_{\mathbf{C}_1\mathbf{C}_0} & P_{\mathbf{C}_1\mathbf{C}_1} & \dots & P_{\mathbf{C}_1\mathbf{C}_{N-1}} \\ P_{\mathbf{C}_2\mathbf{C}_0} & P_{\mathbf{C}_2\mathbf{C}_1} & \dots & P_{\mathbf{C}_2\mathbf{C}_{N-1}} \\ & \dots & \dots & \\ P_{\mathbf{C}_{N-1}\mathbf{C}_0} & P_{\mathbf{C}_{N-1}\mathbf{C}_1} & \dots & P_{\mathbf{C}_{N-1}\mathbf{C}_{N-1}} \end{bmatrix} \tag{6.73}$$

where the 3x3 covariance matrices comprising $P_{\mathbf{CC}}$ are obtained from Equation (6.71) and (6.72) for each of the N control points. The deviation to be applied to the control mesh in a given Monte-Carlo run is thus

$$\delta\mathbf{C} = L\delta\mathbf{U} \tag{6.74}$$

where $\delta\mathbf{U} \in \mathbb{R}^{3N}$ is a random vector obeying a Gaussian distribution of zero-mean and unity covariance, and L being *a* square-root of $P_{\mathbf{CC}}$. When N gets larger that a few thousands, getting L from the Cholesky decomposition of $P_{\mathbf{CC}}$ may fail due to numerical instabilities. Instead, the spectral decomposition $P_{\mathbf{CC}} = HDH^T$ where D (resp H) is the diagonal matrix storing the positive eigenvalues (resp the orthogonal matrix of unity eigenvectors) was empirically found to be better behaved than the Cholesky decomposition in this case, with L then readily provided by $L = H\sqrt{D}H^T$.

Table 6.1 shows the resolution and mean inertia parameters of the polyhedron shape model of asteroid Itokawa [104] and the polyhedron shape model of comet 67P/ChuryumovGerasimenko [105] used in this study. Table 6.2 lists the input parameters of the different tests that were run in this study. Cases 1 and 2 consider a shape model of Asteroid Itokawa under different uncertainty levels while Case 3 focuses on Comet ChuryumovGerasimenko (67P)

A comparison of the Monte-Carlo and predicted statistical moments of the different considered inertia terms is shown on Tables 6.3, 6.4 and 6.5 for the three different cases. The agreement between the Monte-Carlo moments and the analytical ones appears very good overall. The large deviations than can be seen for some of the cross correlations are fairly inconsequential for the shape of the covariance ellipses themselves, as the off-diagonal correlations are negligible compared to the diagonal ones.

Figures 6.2, 6.3 and 6.4 show a subsample of the generated Monte-Carlo shapes for the three different cases. Furthermore, as said in the previous paragraph, the covariances computed from the analytical model are nearly indistinguishable from those computed from the Monte-Carlo samples, as shown in the distribution of the centers of mass (Figures 6.5,6.6 and 6.7), principal inertia moments (Figures 6.8, 6.9 and 6.10), principal dimensions (Figures 6.11, 6.12 and 6.13) and principal axes MRP (Figures 6.14, 6.15 and 6.16). The few outliers seen in Figure 6.15 are discussed hereunder.

Table 6.1: Small bodies mean inertia parameters

	Itokawa_8	67P/C-G
Vertices	386	2501
Elements	768	4998
Volume V (km^3)	0.0173	18.5506
Center of mass $\mathbf{c}_m$ (km)	$\mathbf{0}_3$	$\mathbf{0}_3$
Inertia $[I]_{\mathbf{c}_m}/\rho$ (km^5)	$\begin{bmatrix} 1.072\cdot 10^{-4} & -9.479\cdot 10^{-21} & 4.87\cdot 10^{-20} \\ \cdot & 3.535\cdot 10^{-4} & -1.036\cdot 10^{-20} \\ \cdot & \cdot & 3.723\cdot 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 1.759\cdot 10^{1} & -2.512\cdot 10^{-14} & -3.26\cdot 10^{-15} \\ \cdot & 3.243\cdot 10^{1} & 6.9\cdot 10^{-16} \\ \cdot & \cdot & 3.490\cdot 10^{1} \end{bmatrix}$
Principal moments $A/\rho, B/\rho, C/\rho$ (km^5)	$\begin{pmatrix} 1.072\cdot 10^{-4} \\ 3.535\cdot 10^{-4} \\ 3.723\cdot 10^{-4} \end{pmatrix}$	$\begin{pmatrix} 1.759\cdot 10^{1} \\ 3.243\cdot 10^{1} \\ 3.490\cdot 10^{1} \end{pmatrix}$
Principal dimensions a, b, c (km)	$\begin{pmatrix} 2.991\cdot 10^{-1} \\ 1.350\cdot 10^{-1} \\ 1.131\cdot 10^{-1} \end{pmatrix}$	$\begin{pmatrix} 2.589 \\ 1.644 \\ 1.428 \end{pmatrix}$
Principal axes MRP $\boldsymbol{\sigma}$	$\mathbf{0}_3$	$\mathbf{0}_3$

Table 6.2: Simulation parameters

	Case 1	Case 2	Case 3
Shape model	Itokawa_8	Itokawa_8	67P/C-G
Standard deviation σ (m)	5	10	75
Correlation length l (m)	50	100	300

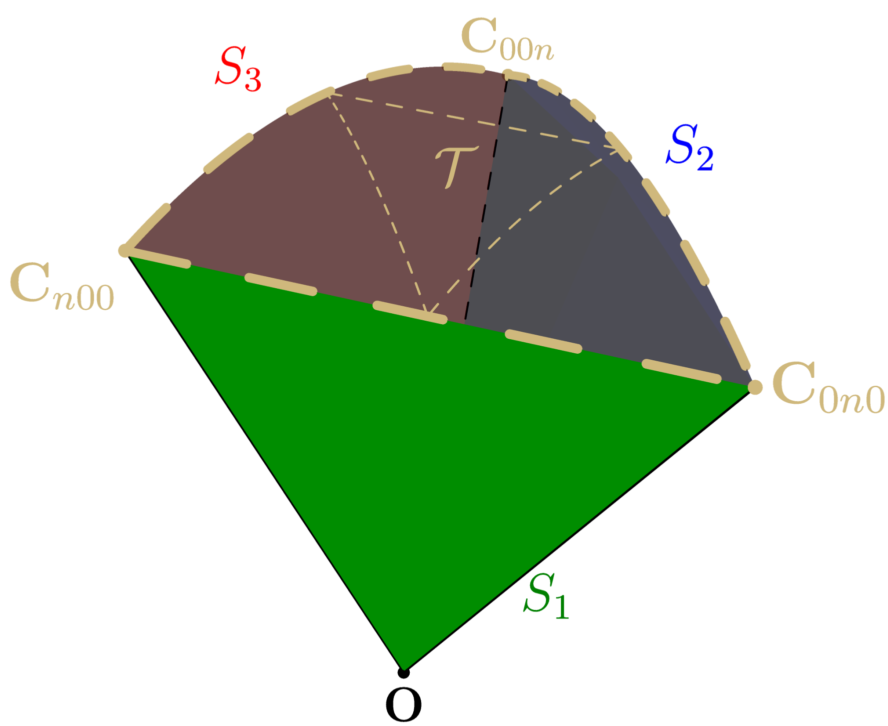


Figure 6.1: Illustration of a Bezier tetrahedron, generated from the triangular Bezier patch $\mathcal{T}$ subtending 3 sides S_1, S_2 and S_3 between its end control points $\mathbf{C}_{n00}$, $\mathbf{C}_{0n0}$ and $\mathbf{C}_{00n}$ and an arbitrary origin $\mathbf{O}$

.

Table 6.3: Volume, center of mass, inertia moments, principal dimensions and principal axes covariance error between analytical prediction and Monte-Carlo results, case 1

	Monte-Carlo	Modeled	Deviation (%)
σ_V (km^3)	0.000267724	0.000267141	0.218
$P_{\mathbf{c}_m}$ (m^2)	$\begin{bmatrix} 5.377 & -7.056 \cdot 10^{-2} & 2.502 \cdot 10^{-2} \\ \cdot & 1.359 & -1.743 \cdot 10^{-3} \\ \cdot & \cdot & 1.024 \end{bmatrix}$	$\begin{bmatrix} 5.253 & -1.358 \cdot 10^{-2} & 1.838 \cdot 10^{-2} \\ \cdot & 1.371 & -3.499 \cdot 10^{-4} \\ \cdot & \cdot & 1.032 \end{bmatrix}$	$\begin{bmatrix} -2.304 & -80.749 & -26.529 \\ \cdot & 0.935 & -79.921 \\ \cdot & \cdot & 0.780 \end{bmatrix}$
P_{ABC} (km^{10})	$\begin{bmatrix} 9.939 \cdot 10^{-12} & 1.756 \cdot 10^{-11} & 1.971 \cdot 10^{-11} \\ \cdot & 8.348 \cdot 10^{-11} & 8.342 \cdot 10^{-11} \\ \cdot & \cdot & 8.777 \cdot 10^{-11} \end{bmatrix}$	$\begin{bmatrix} 9.770 \cdot 10^{-12} & 1.746 \cdot 10^{-11} & 1.957 \cdot 10^{-11} \\ \cdot & 8.274 \cdot 10^{-11} & 8.291 \cdot 10^{-11} \\ \cdot & \cdot & 8.732 \cdot 10^{-11} \end{bmatrix}$	$\begin{bmatrix} -1.705 & -0.558 & -0.684 \\ \cdot & -0.891 & -0.614 \\ \cdot & \cdot & -0.515 \end{bmatrix}$
P_{abc} (km^2)	$\begin{bmatrix} 7.623 \cdot 10^{-6} & -9.780 \cdot 10^{-7} & -4.415 \cdot 10^{-7} \\ \cdot & 2.667 \cdot 10^{-6} & -4.643 \cdot 10^{-7} \\ \cdot & \cdot & 1.988 \cdot 10^{-6} \end{bmatrix}$	$\begin{bmatrix} 7.626 \cdot 10^{-6} & -9.182 \cdot 10^{-7} & -4.411 \cdot 10^{-7} \\ \cdot & 2.607 \cdot 10^{-6} & -4.341 \cdot 10^{-7} \\ \cdot & \cdot & 1.949 \cdot 10^{-6} \end{bmatrix}$	$\begin{bmatrix} 0.045 & -6.110 & -0.094 \\ \cdot & -2.244 & -6.507 \\ \cdot & \cdot & -1.922 \end{bmatrix}$
$P_{\boldsymbol{\sigma}}$	$\begin{bmatrix} 1.563 \cdot 10^{-4} & 1.070 \cdot 10^{-6} & -8.639 \cdot 10^{-7} \\ \cdot & 4.425 \cdot 10^{-6} & 3.914 \cdot 10^{-7} \\ \cdot & \cdot & 6.337 \cdot 10^{-6} \end{bmatrix}$	$\begin{bmatrix} 1.555 \cdot 10^{-4} & 8.457 \cdot 10^{-7} & -8.820 \cdot 10^{-7} \\ \cdot & 4.493 \cdot 10^{-6} & 3.565 \cdot 10^{-7} \\ \cdot & \cdot & 6.281 \cdot 10^{-6} \end{bmatrix}$	$\begin{bmatrix} -0.525 & -20.954 & 2.101 \\ \cdot & 1.550 & -8.918 \\ \cdot & \cdot & -0.885 \end{bmatrix}$

Table 6.4: Volume, center of mass, inertia moments, principal dimensions and principal axes covariance error between analytical prediction and Monte-Carlo results, case 2

	Monte-Carlo	Modeled	Deviation (%)
σ_V (km^3)	0.001101	0.001095	0.664
$P_{\mathbf{c}_m}$ (m^2)	$\begin{bmatrix} 8.171 \cdot 10^{1} & -1.000 & 4.385 \cdot 10^{-1} \\ \cdot & 1.792 \cdot 10^{1} & -1.467 \cdot 10^{-2} \\ \cdot & \cdot & 1.285 \cdot 10^{1} \end{bmatrix}$	$\begin{bmatrix} 8.046 \cdot 10^{1} & -2.037 \cdot 10^{-1} & 3.523 \cdot 10^{-1} \\ \cdot & 1.807 \cdot 10^{1} & 1.707 \cdot 10^{-3} \\ \cdot & \cdot & 1.292 \cdot 10^{1} \end{bmatrix}$	$\begin{bmatrix} -1.522 & -79.634 & -19.648 \\ \cdot & 0.856 & -111.640 \\ \cdot & \cdot & 0.558 \end{bmatrix}$
P_{ABC} (km^{10})	$\begin{bmatrix} 1.657 \cdot 10^{-10} & 3.193 \cdot 10^{-10} & 3.493 \cdot 10^{-10} \\ \cdot & 1.294 \cdot 10^{-9} & 1.311 \cdot 10^{-9} \\ \cdot & \cdot & 1.367 \cdot 10^{-9} \end{bmatrix}$	$\begin{bmatrix} 1.557 \cdot 10^{-10} & 3.043 \cdot 10^{-10} & 3.322 \cdot 10^{-10} \\ \cdot & 1.247 \cdot 10^{-9} & 1.264 \cdot 10^{-9} \\ \cdot & \cdot & 1.317 \cdot 10^{-9} \end{bmatrix}$	$\begin{bmatrix} -6.036 & -4.712 & -4.880 \\ \cdot & -3.663 & -3.632 \\ \cdot & \cdot & -3.620 \end{bmatrix}$
P_{abc} (km^2)	$\begin{bmatrix} 9.679 \cdot 10^{-5} & -1.127 \cdot 10^{-5} & -3.461 \cdot 10^{-6} \\ \cdot & 2.718 \cdot 10^{-5} & 2.161 \cdot 10^{-6} \\ \cdot & \cdot & 2.070 \cdot 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 9.766 \cdot 10^{-5} & -1.094 \cdot 10^{-5} & -3.582 \cdot 10^{-6} \\ \cdot & 2.748 \cdot 10^{-5} & 1.859 \cdot 10^{-6} \\ \cdot & \cdot & 2.107 \cdot 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 0.903 & -2.988 & 3.484 \\ \cdot & 1.098 & -13.998 \\ \cdot & \cdot & 1.779 \end{bmatrix}$
$P_{\boldsymbol{\sigma}}$	$\begin{bmatrix} 1.589 \cdot 10^{-3} & 1.167 \cdot 10^{-5} & -1.258 \cdot 10^{-5} \\ \cdot & 4.769 \cdot 10^{-5} & 5.408 \cdot 10^{-6} \\ \cdot & \cdot & 7.058 \cdot 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 1.219 \cdot 10^{-3} & 1.217 \cdot 10^{-5} & -1.197 \cdot 10^{-5} \\ \cdot & 4.822 \cdot 10^{-5} & 5.099 \cdot 10^{-6} \\ \cdot & \cdot & 6.930 \cdot 10^{-5} \end{bmatrix}$	$\begin{bmatrix} -23.287 & 4.288 & -4.803 \\ \cdot & 1.122 & -5.717 \\ \cdot & \cdot & -1.820 \end{bmatrix}$

Table 6.5: Volume, center of mass, inertia moments, principal dimensions and principal axes covariance error between analytical prediction and Monte-Carlo results, case 3

	Monte-Carlo	Modeled	Deviation (%)
σ_V (km^3)	0.268	0.266	0.879
$P_{\mathbf{c}_m}$ (m^2)	$\begin{bmatrix} 3.401\cdot 10^{2} & 4.565 & -4.438\cdot 10^{-1} \\ \cdot & 1.701\cdot 10^{2} & 1.111 \\ \cdot & \cdot & 1.358\cdot 10^{2} \end{bmatrix}$	$\begin{bmatrix} 3.460\cdot 10^{2} & 5.443 & -1.503 \\ \cdot & 1.661\cdot 10^{2} & 2.567 \\ \cdot & \cdot & 1.326\cdot 10^{2} \end{bmatrix}$	$\begin{bmatrix} 1.717 & 19.240 & 238.596 \\ \cdot & -2.354 & 131.107 \\ \cdot & \cdot & -2.332 \end{bmatrix}$
P_{ABC} (km^{10})	$\begin{bmatrix} 2.180\cdot 10^{-1} & 2.291\cdot 10^{-1} & 2.660\cdot 10^{-1} \\ \cdot & 5.933\cdot 10^{-1} & 5.780\cdot 10^{-1} \\ \cdot & \cdot & 6.631\cdot 10^{-1} \end{bmatrix}$	$\begin{bmatrix} 2.067\cdot 10^{-1} & 2.166\cdot 10^{-1} & 2.511\cdot 10^{-1} \\ \cdot & 5.742\cdot 10^{-1} & 5.555\cdot 10^{-1} \\ \cdot & \cdot & 6.347\cdot 10^{-1} \end{bmatrix}$	$\begin{bmatrix} -5.217 & -5.464 & -5.612 \\ \cdot & -3.218 & -3.892 \\ \cdot & \cdot & -4.289 \end{bmatrix}$
P_{abc} (km^2)	$\begin{bmatrix} 5.536\cdot 10^{-4} & -7.937\cdot 10^{-5} & -7.193\cdot 10^{-5} \\ \cdot & 3.432\cdot 10^{-4} & -4.536\cdot 10^{-5} \\ \cdot & \cdot & 2.881\cdot 10^{-4} \end{bmatrix}$	$\begin{bmatrix} 5.486\cdot 10^{-4} & -8.674\cdot 10^{-5} & -7.067\cdot 10^{-5} \\ \cdot & 3.416\cdot 10^{-4} & -4.888\cdot 10^{-5} \\ \cdot & \cdot & 2.888\cdot 10^{-4} \end{bmatrix}$	$\begin{bmatrix} -0.900 & 9.289 & -1.759 \\ \cdot & -0.444 & 7.749 \\ \cdot & \cdot & 0.256 \end{bmatrix}$
$P_{\boldsymbol{\sigma}}$	$\begin{bmatrix} 2.631\cdot 10^{-4} & 4.285\cdot 10^{-6} & -1.385\cdot 10^{-6} \\ \cdot & 1.170\cdot 10^{-5} & -6.113\cdot 10^{-7} \\ \cdot & \cdot & 1.987\cdot 10^{-5} \end{bmatrix}$	$\begin{bmatrix} 2.532\cdot 10^{-4} & 4.403\cdot 10^{-6} & -1.788\cdot 10^{-6} \\ \cdot & 1.155\cdot 10^{-5} & -3.663\cdot 10^{-7} \\ \cdot & \cdot & 1.931\cdot 10^{-5} \end{bmatrix}$	$\begin{bmatrix} -3.751 & 2.764 & 29.036 \\ \cdot & -1.332 & -40.073 \\ \cdot & \cdot & -2.815 \end{bmatrix}$

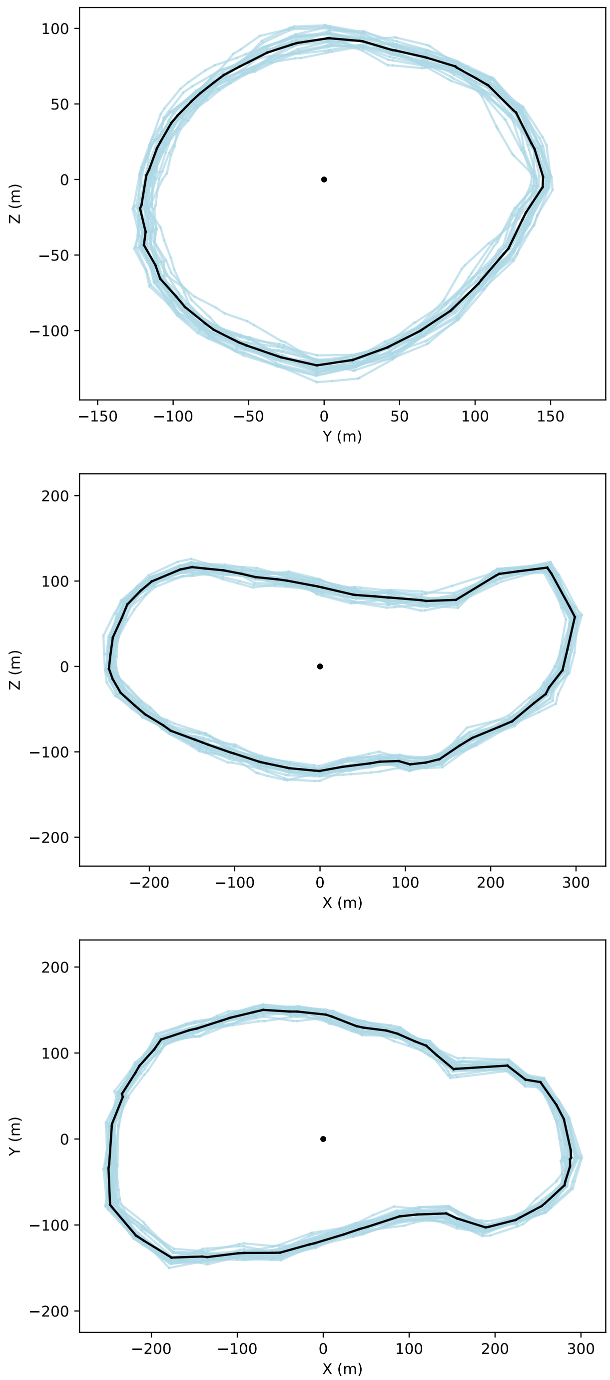


Figure 6.2: Mean (black) and MC shapes (blue), case 1. Only a fraction of the MC outcomes are shown

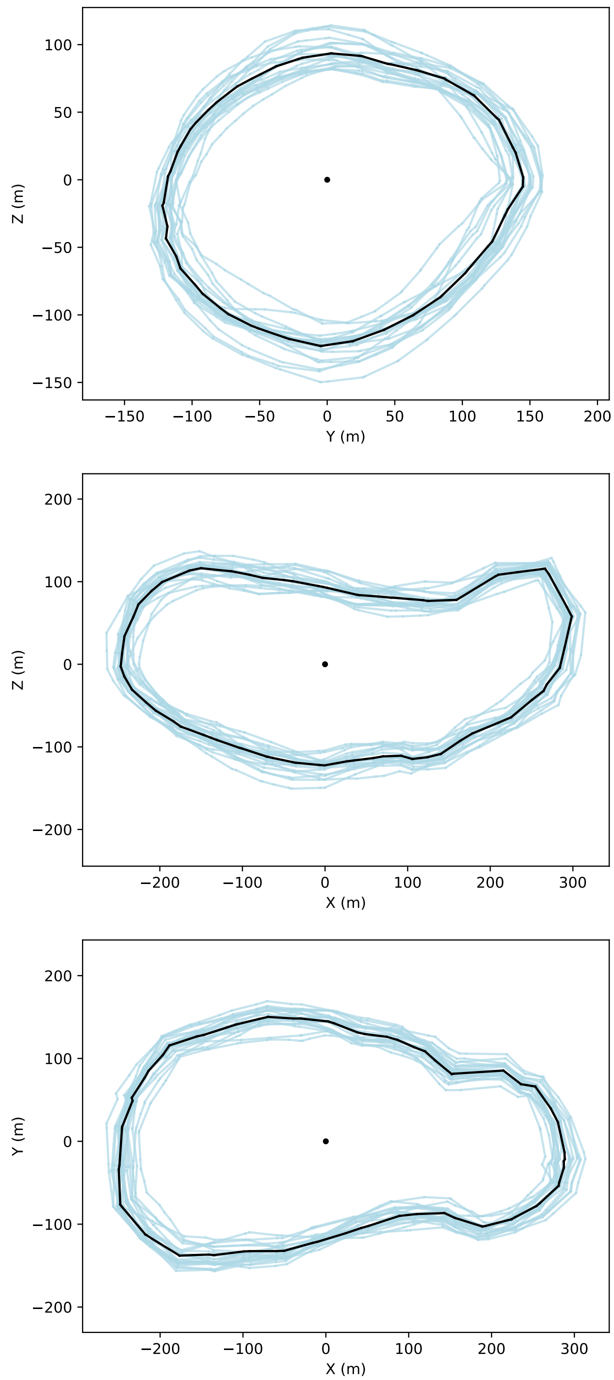


Figure 6.3: Mean (black) and MC shapes (blue), case 2. Only a fraction of the MC outcomes are shown

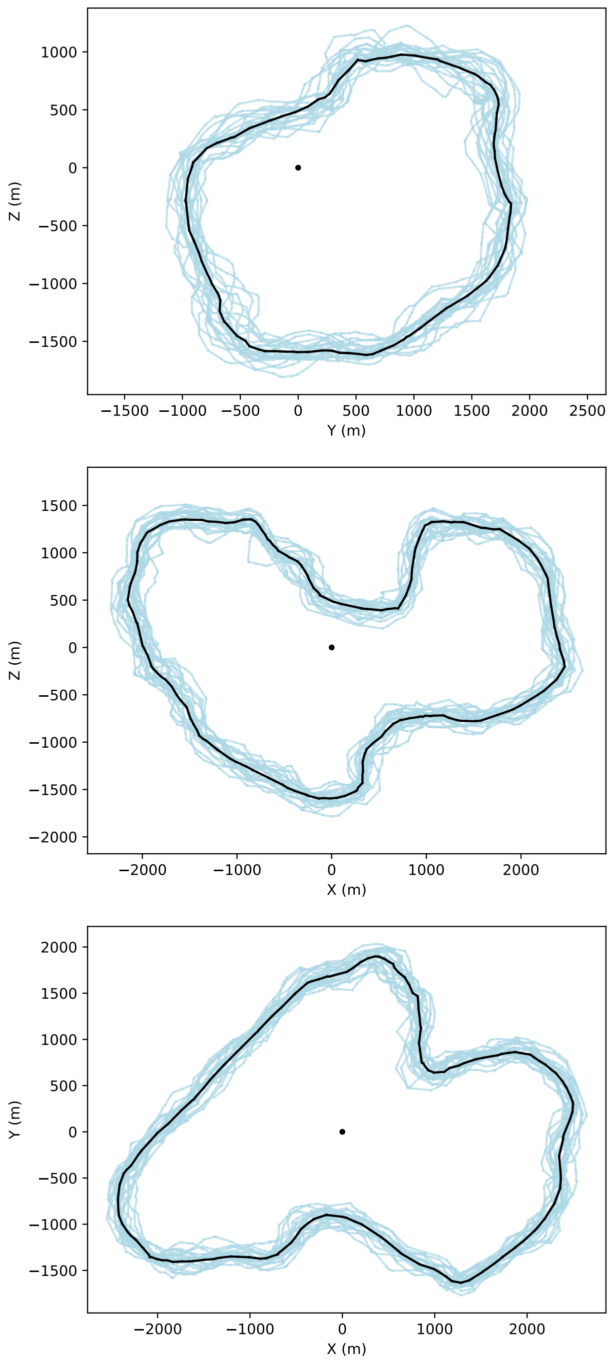


Figure 6.4: Mean (black) and MC shapes (blue), case 3. Only a fraction of the MC outcomes are shown

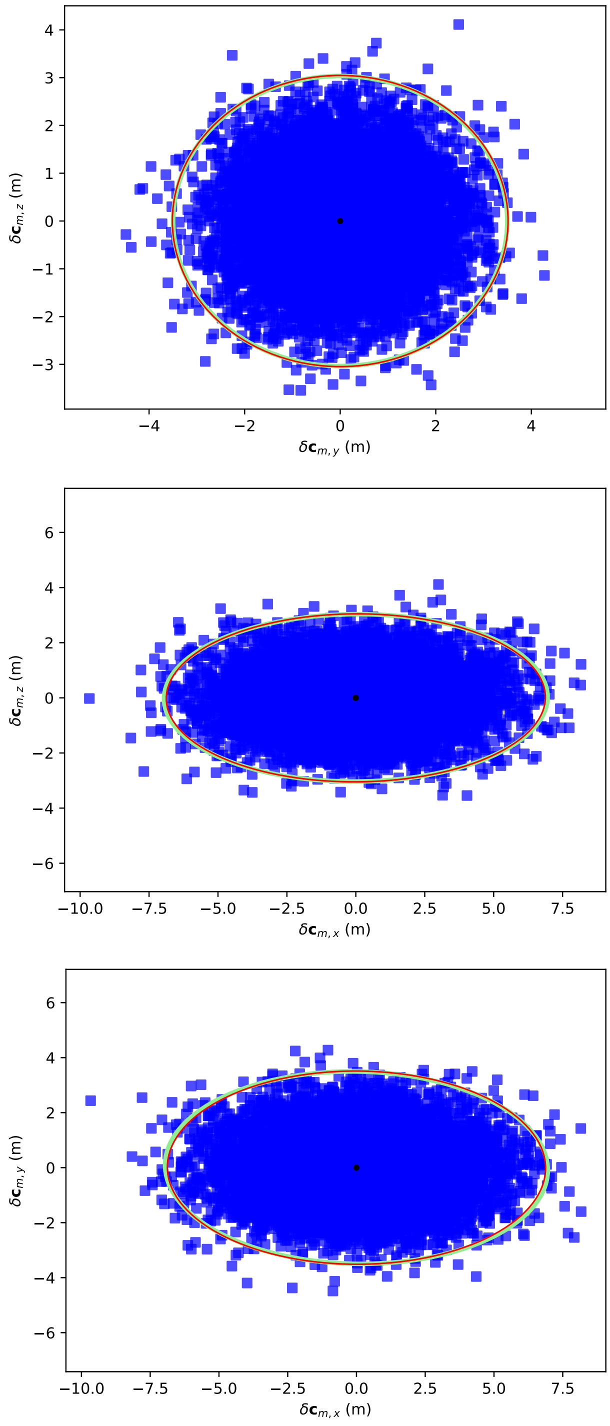


Figure 6.5: Dispersion in the center of mass coordinates (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 1

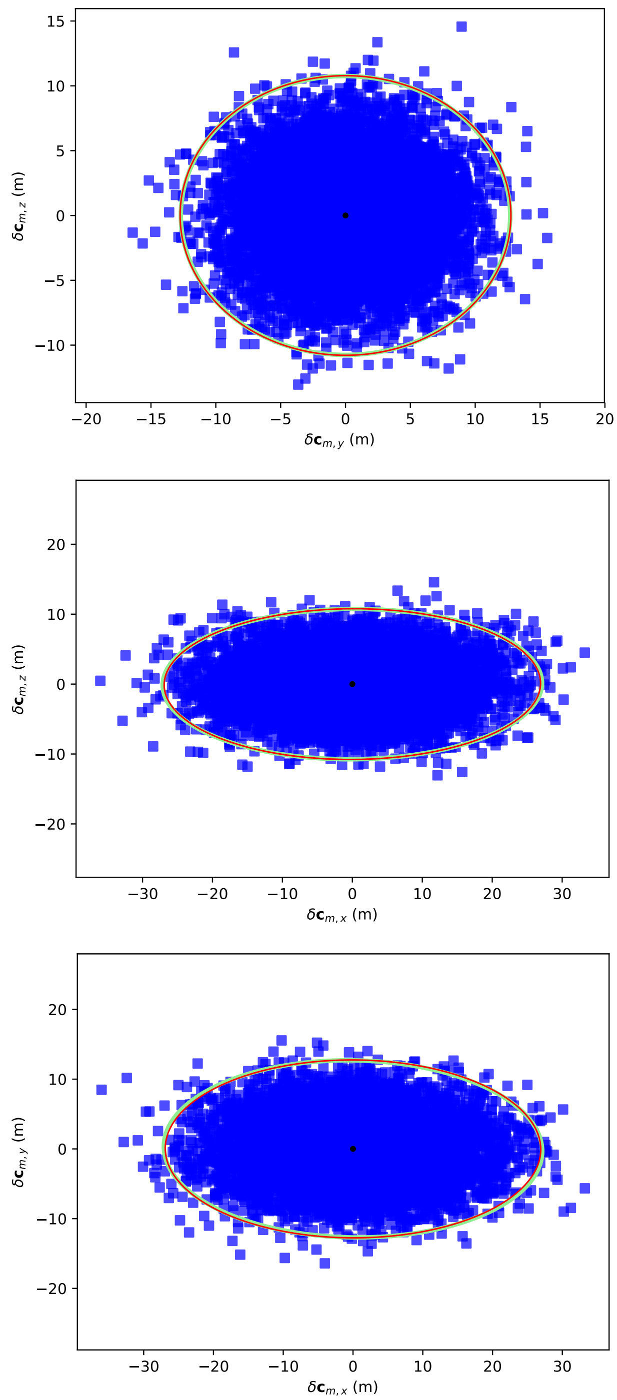


Figure 6.6: Dispersion in the center of mass coordinates (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 2

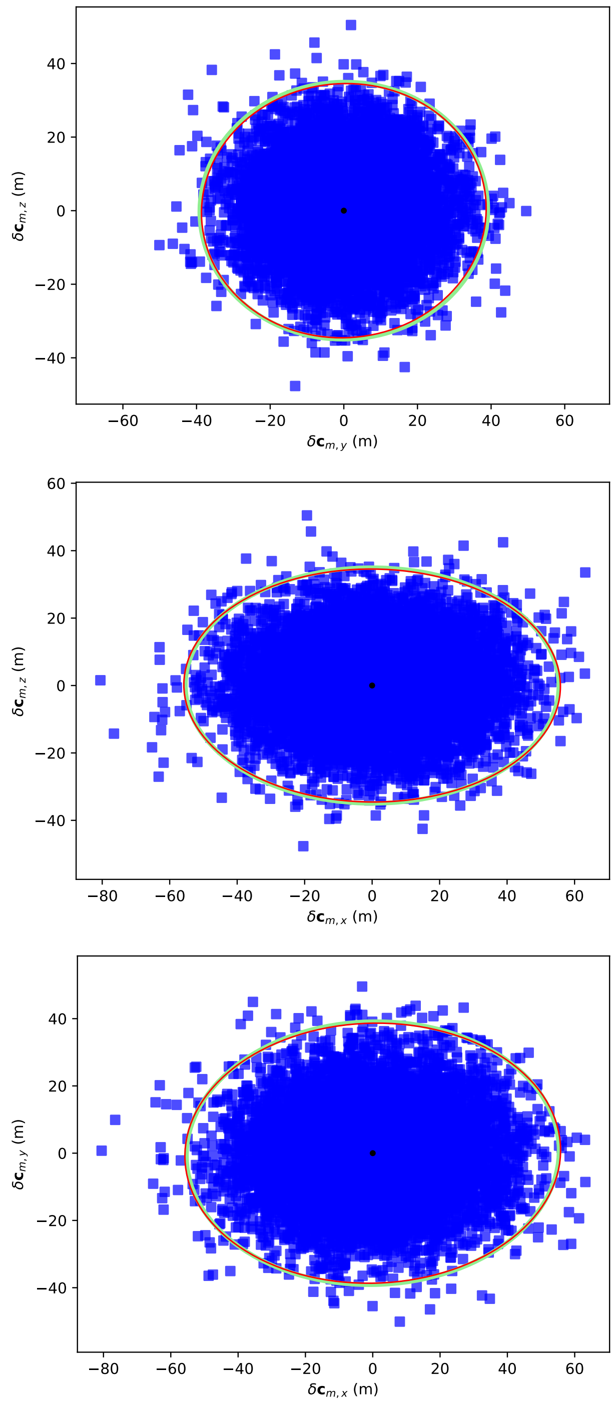


Figure 6.7: Dispersion in the center of mass coordinates (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 3

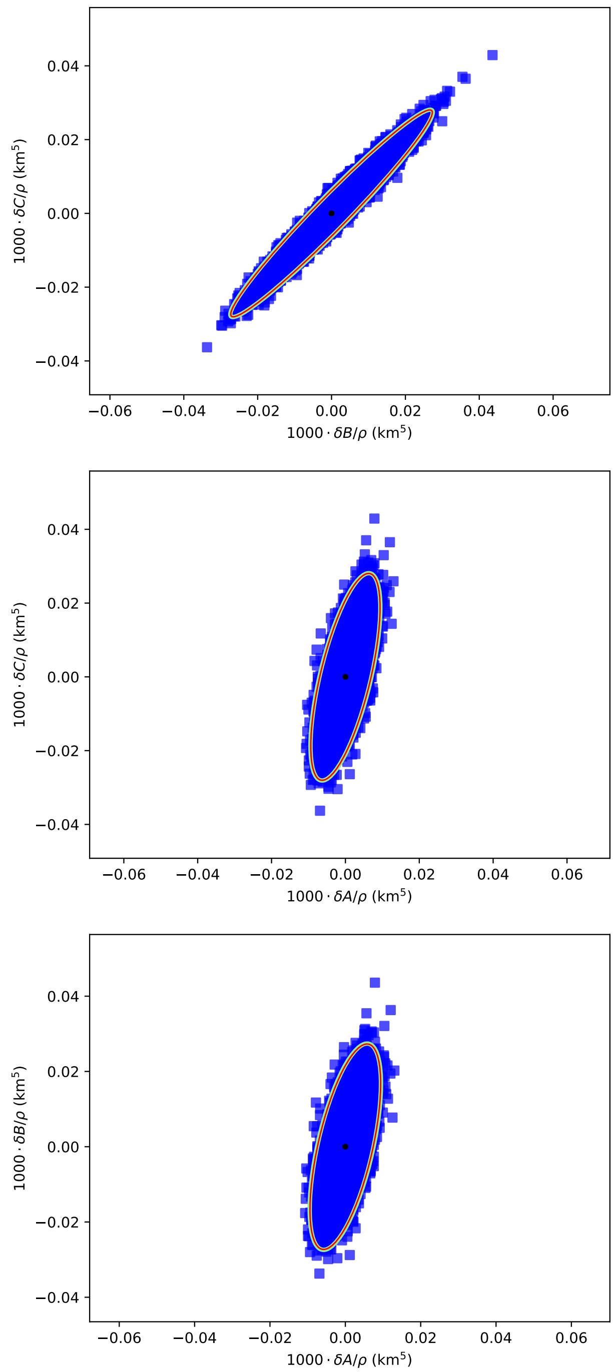


Figure 6.8: Dispersion in the three principal moments (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 1

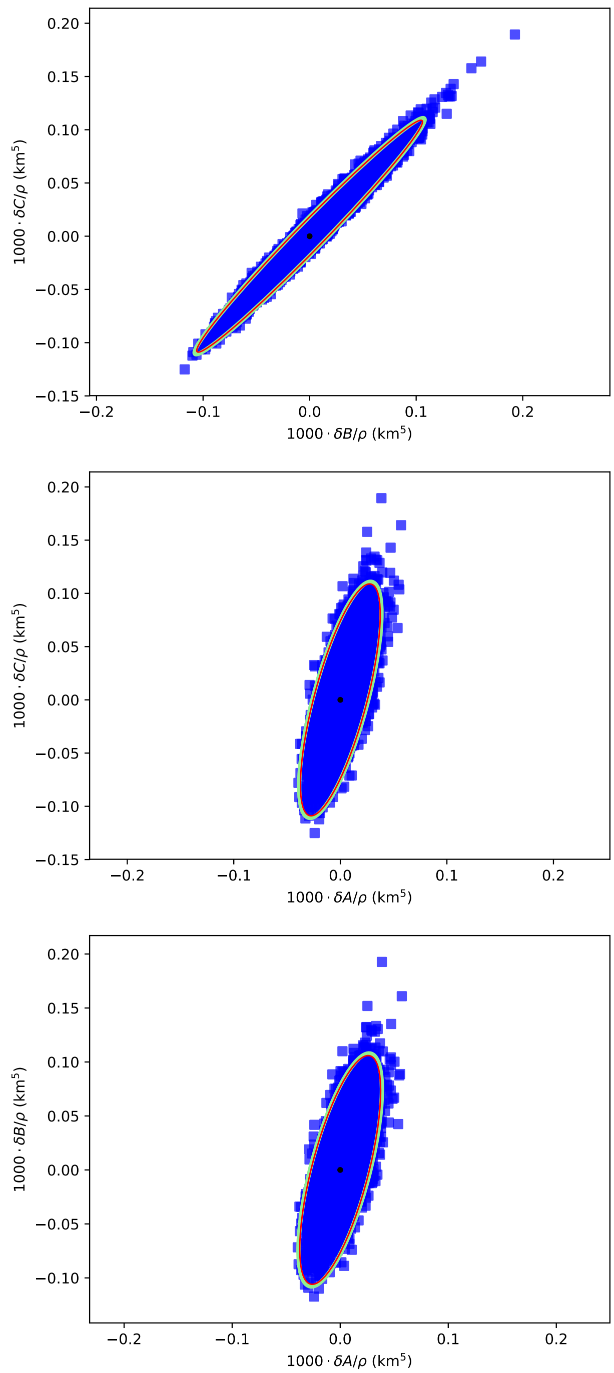


Figure 6.9: Dispersion in the three principal moments (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 2

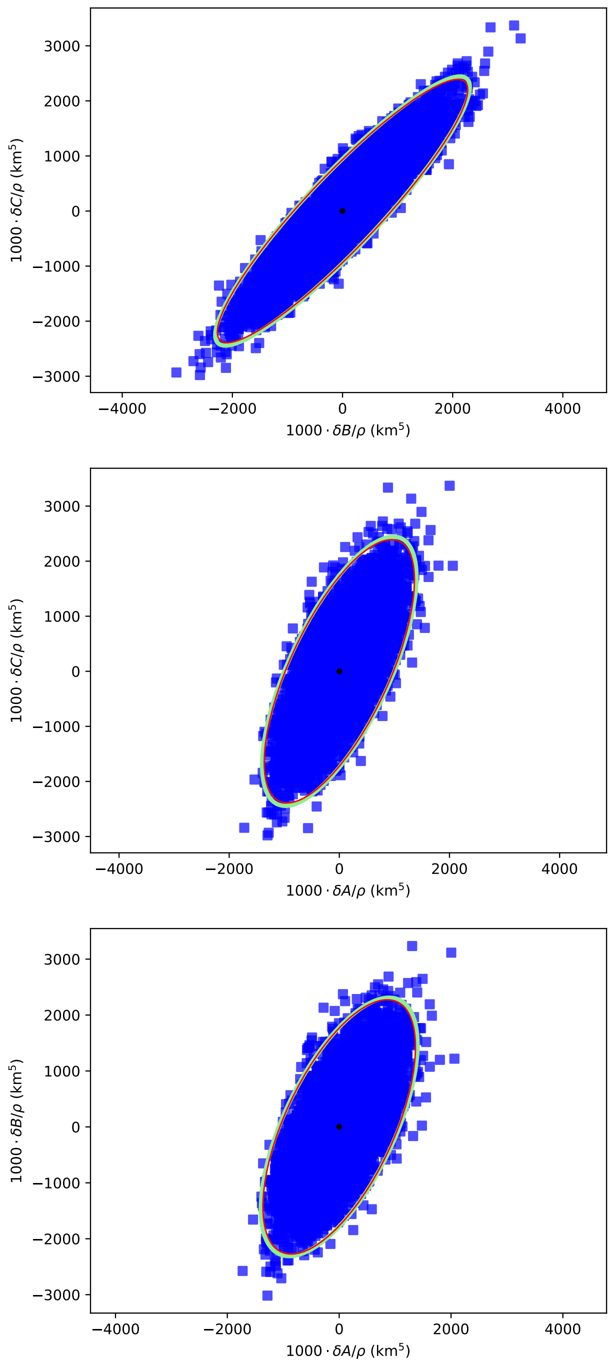


Figure 6.10: Dispersion in the three principal moments (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 3

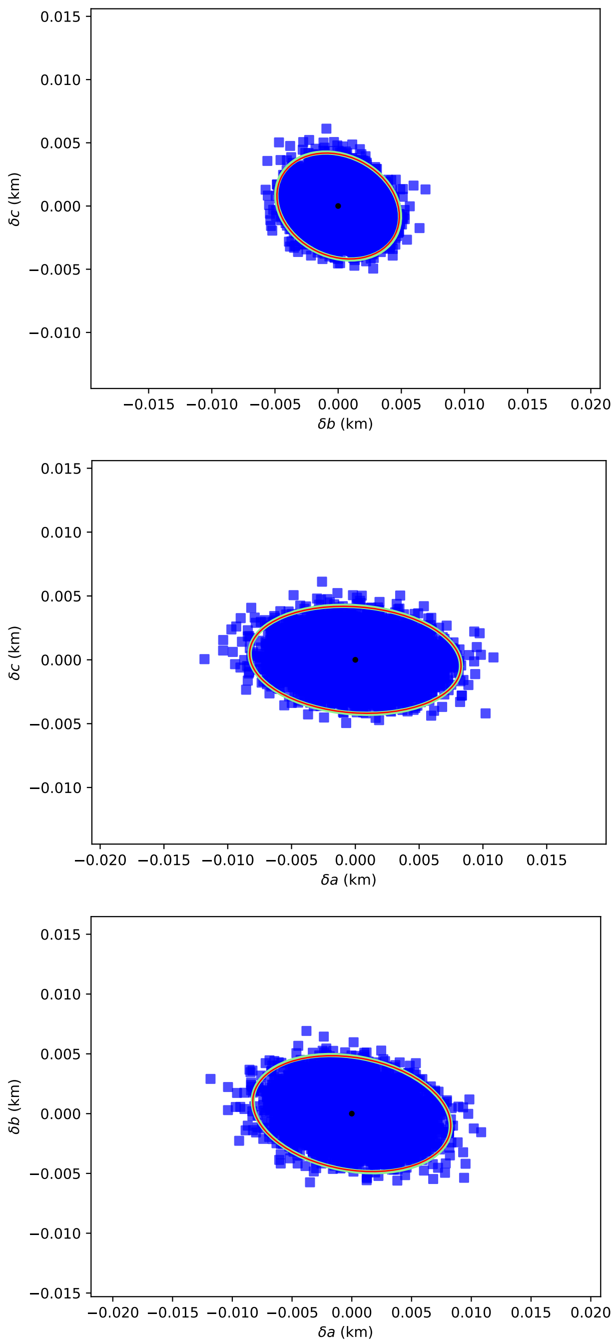


Figure 6.11: Dispersion in the three principal dimensions (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 1

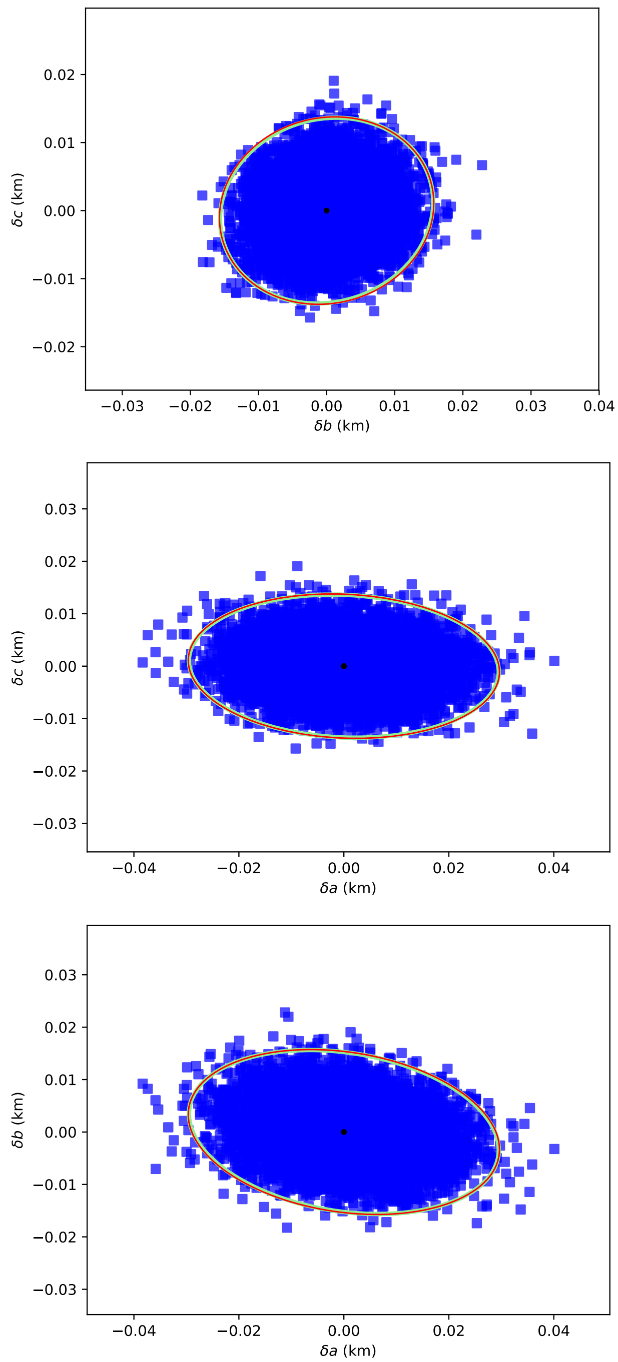


Figure 6.12: Dispersion in the three principal dimensions (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 2

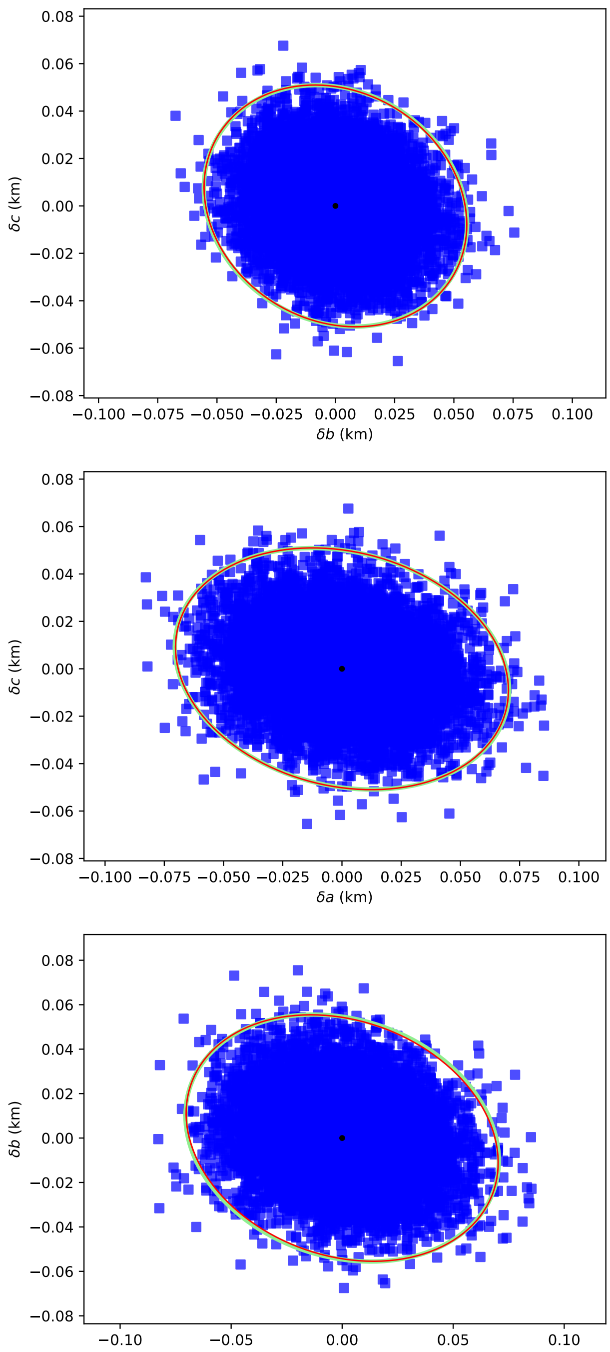


Figure 6.13: Dispersion in the three principal dimensions (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 3

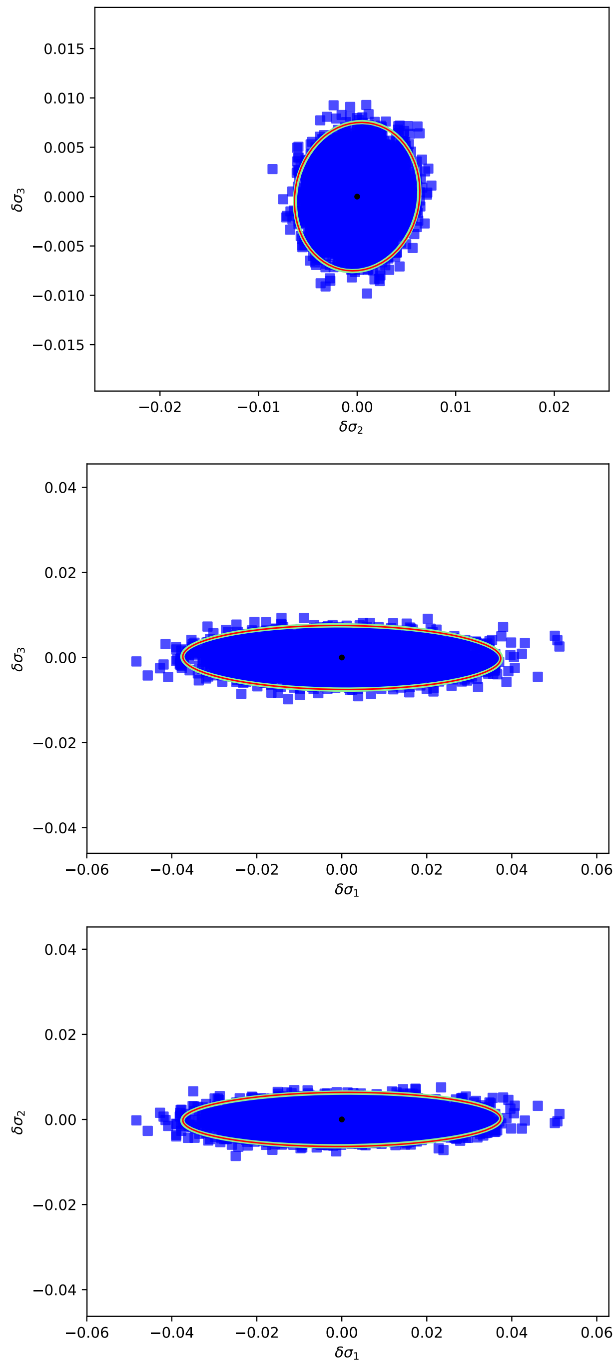


Figure 6.14: Dispersion in the principal axes MRP (blue) and 3 σellipses from the MC (green) and from the uncertainty model (red), case 1

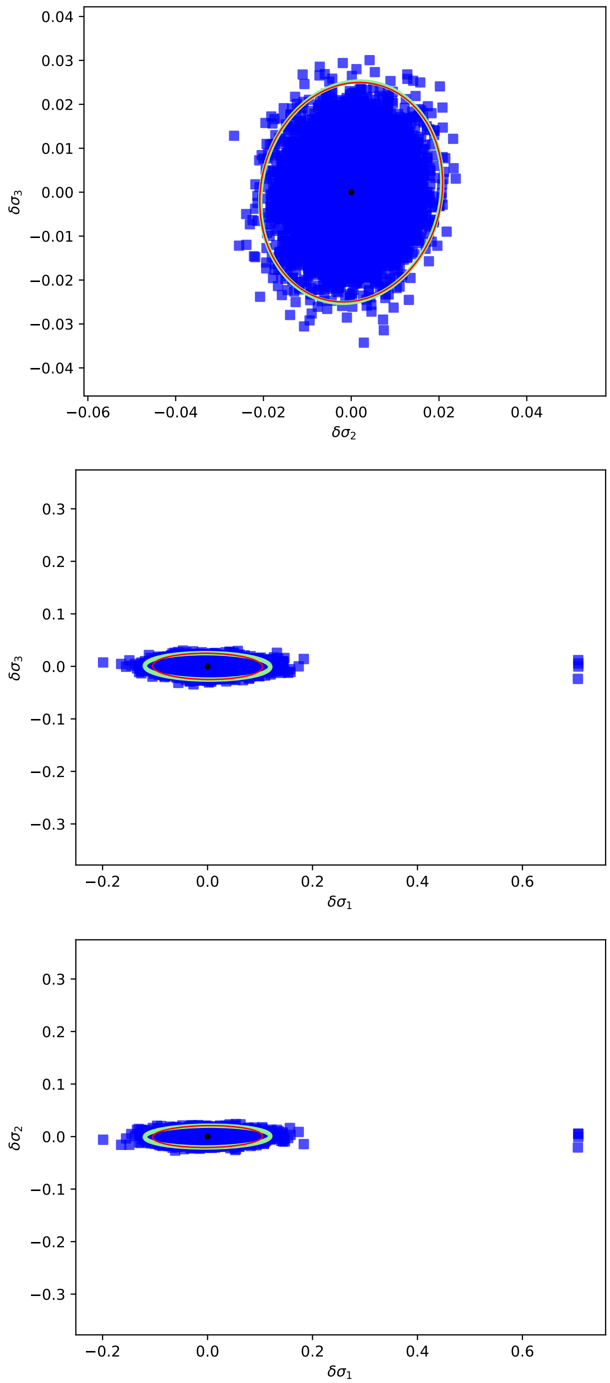


Figure 6.15: Dispersion in the principal axes MRP (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 2

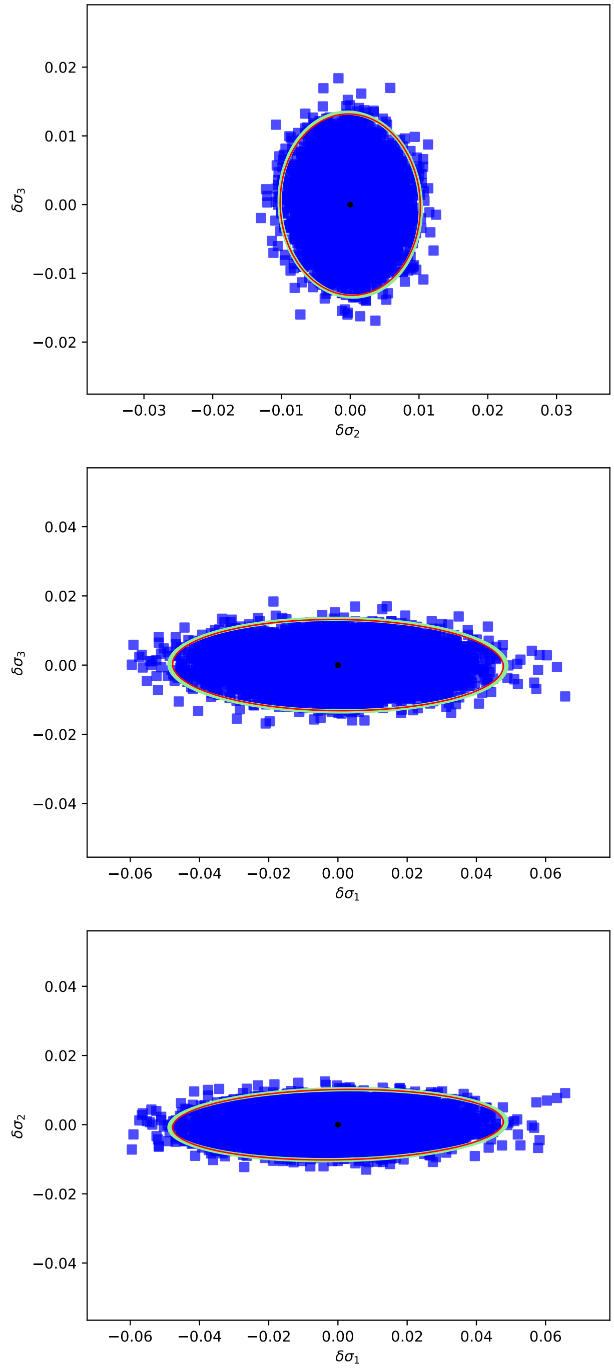


Figure 6.16: Dispersion in the principal axes MRP (blue) and 3 σ ellipses from the MC (green) and from the uncertainty model (red), case 3

6.3 Discussion

The results from the numerical simulations have confirmed the ability of the derived model to predict the variations in the inertia characteristics of a shape given surface uncertainties, under a linearized error assumption. The quality of the prediction is good at the considered error levels, but would degrade as these get bigger due to the violation of the linearized formulation. The few outliers that can be seen in Figure 6.15 showing a seemingly large deviation in the first component of the MRP are caused by the coalescence of the second and third inertia moments of Itokawa under the applied deviation, the former becoming larger than the third. This causes two principal axes to flip, effectively affecting the MRP governing their orientation. These outliers are thus the results of a mislabeling of the principal axes in the Monte-Carlo outcome evaluation and not a limitation of the linearized uncertainty model. If one were to run a MC simulation without visually checking the results, but only computing statistics, these types of errors can lead to erroneous results, which is yet another benefit of the proposed approach over pure numerical simulations.

Beyond providing an analytical insight into the behavior of the inertia parameters under shape uncertainties, the developed model could also provide an alternative to Monte-Carlo sampling of the shape uncertainties should the extraction of the square root of the shape covariance become intractable, as the model only operates on the subpartitions of the shape covariance. Although the density was not considered as varying in this work, it would be fairly straightforward to add it to the uncertain parameters as long as it is assumed as uniform, as the uniform density hypothesis would remove any correlations between the uncertainty in the mesh control points and that in the density. The only modification would be to rewrite Equation (6.48) as $\delta M = \rho\delta V + \delta\rho V$, and augment the input statistics with $E\left(\delta\rho^2\right) = \sigma_\rho^2$. Allowing the density distribution to become non-uniform and uncertain would drastically increase the computational burden associated with the inertia statistics. It must also be noted that the density would have to remain piecewise-constant across the Bezier tetrahedrons comprising the shape.

Since the uncertainty model derived in this chapter relies on a local description of the asteroid terrain, it is able to handle highly irregular, non-convex bodies that cannot be described in terms of spherical coordinates. This is an improvement over Muinonen's Gaussian sphere technique. Also, there is no limitation on the correlation distance between a given pair of points, as the only limiting factor is the norm of the uncertainty in the shape control points which should remain small relative to the shape's dimensions.

Although the methods were only demonstrated over a polyhedron, the provided expressions remain valid for Bezier shapes of arbitrary order n strictly greater than 1. Yet, one must be wary of the combinatorial explosion occurring when computing the statistical moments of the inertia properties. In particular, evaluating $P_{\mathbf{I}}$ over a Bezier shape of order $n = 2$ would require a maximum of $6^{10} = 60466176$ evaluations per facet pair. In practice, this number can be reduced by eliminating the $\kappa_{\mathbf{ijklm}}\kappa_{\mathbf{pqrst}}$ products evaluating to zero. The remaining computational burden can still be spread out over multiple agents due to its embarrassingly parallel nature.

Finally, it must be noted that this chapter does not address the computation of the covariances $P_{\mathbf{C_i C_j}}$, as these would normally be provided along with a shape model estimate reconstructed by means of remote observations (lightcurve, radar,...). [106] provide a detailed procedure explaining how one could come up with the corresponding covariances for a shape reconstructed by means of point cloud data acquired by a Lidar instrument. Other observations types like radar images or luminosity curves could be handled similarly. Since the gravity spherical harmonics less or equal than two in degree and order are a function of the inertia tensor parameters [107], analytical quantification of the uncertainty in the orbit dynamics about an unknown small body is thus captured by the linearized model up to the second degree and order.

Chapter 7

Uncertainties in Polyhedron Gravity Model Arising From An Uncertain Shape

7.1 The Polyhedron Gravity Model

Werner and Scheeres proposed closed form expressions of the potential created by a constant-density polyhedral shape comprised of triangular surface elements (dubbed "facets"), from which expressions of the gravity acceleration and gravity gradient matrix could be readily derived [5]. These expressions are known as that of the Polyhedron Gravity Model, or PGM. Denote the i-th shape vertex as $\mathbf{C}_i$. An edge indexed by e is formed by connecting two points $\mathbf{C}_{i^E_{e,0}}$ and $\mathbf{C}_{i^E_{e,1}}$. A facet indexed by f is formed by associating in a counter-clockwise fashion the points $\mathbf{C}_{i^F_{f,0}}$, $\mathbf{C}_{i^F_{f,1}}$ and $\mathbf{C}_{i^F_{f,2}}$. The potential, acceleration and gravity-gradient matrix arising from a polyhedron shape comprised of N_f facets and N_e edges of constant uniform density ρ takes the form

$$U\left(\mathbf{r}\right) = \frac{G\rho}{2}\left[\sum_{e=1}^{N_e}\mathbf{r}^T_{i^E_{e,0}}E_e\mathbf{r}_{i^E_{e,0}}L_e - \sum_{f=1}^{N_f}\mathbf{r}^T_{i^F_{f,0}}F_f\mathbf{r}_{i^F_{f,0}}\omega_f\right] \tag{7.1}$$

$$\mathbf{a}\left(\mathbf{r}\right) = G\rho\left[-\sum_{e=1}^{N_e}E_e\mathbf{r}_{i^E_{e,0}}L_e + \sum_{f=1}^{N_f}F_f\mathbf{r}_{i^F_{f,0}}\omega_f\right] \tag{7.2}$$

$$\left[\frac{\partial\mathbf{a}}{\partial\mathbf{r}}\right] = G\rho\left[\sum_{e=1}^{N_e}E_eL_e - \sum_{f=1}^{N_f}F_f\omega_f\right] \tag{7.3}$$

The edge potential L_e, performance factor ω_f and the other terms constituting these expressions are detailed below.

$$\mathbf{r}_i = \mathbf{C}_i - \mathbf{r} \tag{7.4}$$

$$r_i = \|\mathbf{r}_i\| \tag{7.5}$$

$$L_e = \ln\left(\frac{r_{i_{e,0}} + r_{i_{e,1}} + l_e}{r_{i_{e,0}} + r_{i_{e,1}} - l_e}\right) \tag{7.6}$$

$$l_e = \|\mathbf{C}_{i^E_{e,1}} - \mathbf{C}_{i^E_{e,0}}\| \tag{7.7}$$

$$\begin{aligned}\omega_f = 2 \cdot \operatorname{arctan2}\Big(&\mathbf{r}^T_{i^F_{f,0}}\left(\mathbf{r}_{i^F_{f,1}} \times \mathbf{r}_{i^F_{f,2}}\right)\\ &, r_{i^F_{f,0}} r_{i^F_{f,1}} r_{i^F_{f,2}} + r_{i^F_{f,0}} \mathbf{r}^T_{i^F_{f,1}} \mathbf{r}_{i^F_{f,2}} + r_{i^F_{f,1}} \mathbf{r}^T_{i^F_{f,2}} \mathbf{r}_{i^F_{f,0}} + r_{i^F_{f,2}} \mathbf{r}^T_{i^F_{f,0}} \mathbf{r}_{i^F_{f,1}}\Big)\end{aligned} \tag{7.8}$$

$$F_f = \hat{n}_f \hat{n}_f^T \tag{7.9}$$

The edge dyad E_e is defined as

$$E_e = \hat{n}_A \hat{n}^A_{12} + \hat{n}_B \hat{n}^B_{21} \tag{7.10}$$

following the notations of Figure 7.1.

The PGM is an exact representation of the gravity field of a constant-density polyhedron, as opposed to a truncated spherical harmonics expansion of its gravity field. The PGM expressions remain valid within the Briouillin sphere of the object, unlike exterior spherical harmonics expansions that diverge once evaluated within it [108]. The PGM in its provided form is valid everywhere except on the edges of the considered shape, although modified PGM expressions dealing with these singularities do exist [109].

7.2 First variation in the PGM expressions

The expressions of the first variations $\delta U(\mathbf{r})$ and $\delta\mathbf{a}(\mathbf{r})$ provide valuable insight into the evolution of these gravity terms under a change in the shape, in addition to providing the pathway towards linearized uncertainty quantification in these quantities.

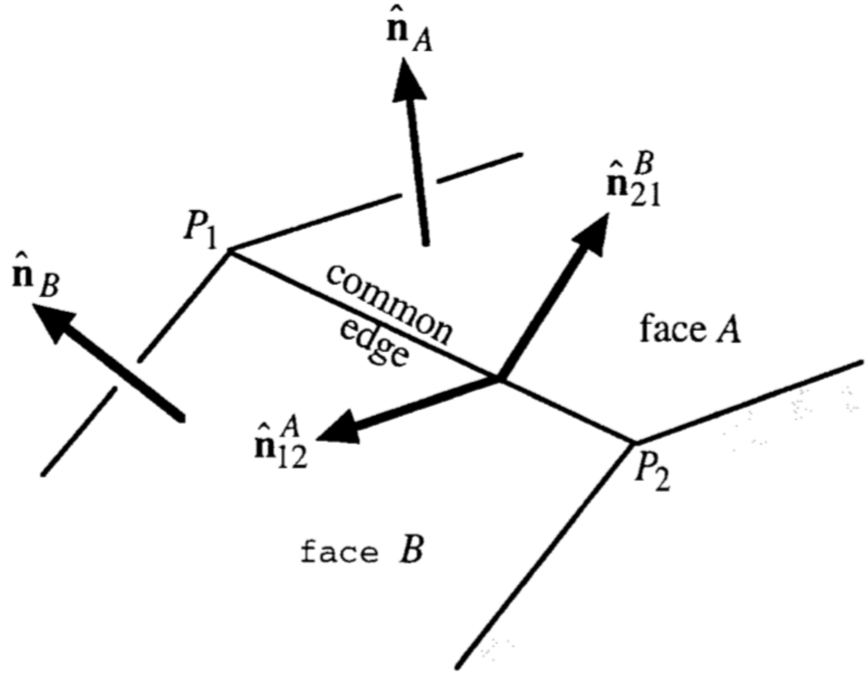


Figure 7.1: Facet-Edge geometry in definition of edge dyad [5]

7.2.1 First variation in the potential due to a change in the shape

Rewrite the potential as

$$U(\mathbf{r}) = \frac{G\rho}{2}\left[\sum_{e=1}^{N_e} U_f^F + \sum_{f=1}^{N_f} U_e^E\right] \tag{7.11}$$

with

$$U_e^E = \mathbf{r}_{i_{e,0}^E}^T E_e \mathbf{r}_{i_{e,0}^E} L_e \tag{7.12}$$

$$U_f^F = -\mathbf{r}_{i_{f,0}^F}^T F_f \mathbf{r}_{i_{f,0}^F} \omega_f \tag{7.13}$$

Taking the first variation yields

$$\delta U(\mathbf{r}) = \frac{G\rho}{2}\left[\sum_{e=1}^{N_e} \delta U_e^E + \sum_{f=1}^{N_f} \delta U_f^F\right] \tag{7.14}$$

with

$$\delta U_e^E = \begin{pmatrix} \mathbf{r}_{i_{e,0}^E}^T E_e \mathbf{r}_{i_{e,0}^E} \\ 2L_e E_e \mathbf{r}_{i_{e,0}^E} \\ L_e R_{i_{e,0}^E}^{ET} \mathbf{r}_{i_{e,0}^E} \end{pmatrix}^T \begin{pmatrix} \delta L_e \\ \delta \mathbf{r}_{i_{e,0}^E} \\ \delta \mathbf{E}_e \end{pmatrix} \tag{7.15}$$

where $\mathbf{E}_e$ and $R_{i_{e,0}^E}^E$ are defined implicitly as $E_e \mathbf{r}_{i_{e,0}^E} = R_{i_{e,0}^E}^E \mathbf{E}_e$. Similarly,

$$\delta U_f^F = \begin{pmatrix} \mathbf{r}_{i_{f,0}^F}^T F_f \mathbf{r}_{i_{f,0}^F} \\ 2\omega_f F_f \mathbf{r}_{i_{f,0}^F} \\ \omega_f R_{i_{f,0}^F}^{FT} \mathbf{r}_{i_{f,0}^F} \end{pmatrix}^T \begin{pmatrix} \delta \omega_f \\ \delta \mathbf{r}_{i_{f,0}^F} \\ \delta \mathbf{F}_f \end{pmatrix} \tag{7.16}$$

where again $F_f \mathbf{r}_{i_{f,0}^F} = R_{i_{f,0}^F}^F \mathbf{F}_f$. In so many words,

$$\delta U_e^E = \frac{\partial U_e^E}{\partial \mathbf{X}_e^E} \delta \mathbf{X}_e^E \tag{7.17}$$

$$\delta U_f^F = \frac{\partial U_f^F}{\partial \mathbf{X}_f^F} \delta \mathbf{X}_f^F \tag{7.18}$$

with

$$\mathbf{X}_e^E = \begin{pmatrix} L_e \\ \mathbf{r}_{i_{e,0}^E} \\ \mathbf{E}_e \end{pmatrix} \tag{7.19}$$

$$\mathbf{X}_f^F = \begin{pmatrix} \omega_f \\ \mathbf{r}_{i_{f,0}^F} \\ \mathbf{F}_f \end{pmatrix} \tag{7.20}$$

The following sections delve into the derivation of the first variations in the different partitions of $\mathbf{X}_e^E$ and $\mathbf{X}_f^F$ so as to relate them to the first variations in the underlying first variation in the vertices coordinates.

7.2.1.1 First variation in $\mathbf{X}_f^F$

The first variation in $\mathbf{X}_f^F$ can be written as

$$\delta\mathbf{X}_f^F = \begin{pmatrix} \delta\omega_f \\ \delta\mathbf{r}_{i_{f,0}^F} \\ \delta\mathbf{F}_f \end{pmatrix} \tag{7.21}$$

$$= \frac{\partial \mathbf{X}_f^F}{\partial \mathbf{T}_f}\delta\mathbf{T}_f \tag{7.22}$$

$$= \begin{bmatrix} \frac{\partial\omega_f}{\partial\mathbf{T}_f} \\ \frac{\partial\mathbf{r}_{i_{f,0}^F}}{\partial\mathbf{T}_f} \\ \frac{\partial\mathbf{F}_f}{\partial\mathbf{T}_f} \end{bmatrix} \delta\mathbf{T}_f \tag{7.23}$$

where the current triangular facet is formed by three control points whose coordinates are stacked in a vector $\mathbf{T}_f$

$$\mathbf{T}_f = \begin{pmatrix} \mathbf{C}_{i_{f,0}^F} \\ \mathbf{C}_{i_{f,1}^F} \\ \mathbf{C}_{i_{f,2}^F} \end{pmatrix} \tag{7.24}$$

The different partials are now defined $\frac{\partial\omega_f}{\partial\mathbf{T}_f}$**:**

One must be wary of the use of the arctan2 function when taking its derivative. Indeed, treating arctan2 like arctan and differentiating (7.8) will yield spurious results should the denominator and numerator be of different signs. Instead, define

$$\mathbf{Z}_f = \begin{pmatrix} 1 + \hat{r}^T_{i_{f,1}^F}\hat{r}_{i_{f,2}^F} + \hat{r}^T_{i_{f,2}^F}\hat{r}_{i_{f,0}^F} + \hat{r}^T_{i_{f,0}^F}\hat{r}_{i_{f,1}^F} \\ \hat{r}^T_{i_{f,0}^F}\left(\hat{r}_{i_{f,1}^F} \times \hat{r}_{i_{f,2}^F}\right) \end{pmatrix} \tag{7.25}$$

$$= \begin{pmatrix} \alpha_f \\ \gamma_f \end{pmatrix} \tag{7.26}$$

$$\hat{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{7.27}$$

$$\hat{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{7.28}$$

and use the following definition of arctan2:

$$\arctan2\left(\mathbf{Z}\right) = 2\arctan\left(\frac{\hat{e}_2^T\mathbf{Z}}{\|\mathbf{Z}\| + \hat{e}_1^T\mathbf{Z}}\right) \tag{7.29}$$

Then, the performance factor simply becomes

$$\omega_f = 4\arctan\left(\frac{\hat{e}_2^T\mathbf{Z}_f}{\|\mathbf{Z}_f\| + \hat{e}_1^T\mathbf{Z}_f}\right) \tag{7.30}$$

Defining $\hat{r}_f = \begin{pmatrix} \hat{r}_{i^F_{f,0}} \\ \hat{r}_{i^F_{f,1}} \\ \hat{r}_{i^F_{f,2}} \end{pmatrix}$ and $\mathbf{r}_f = \begin{pmatrix} \mathbf{r}_{i^F_{f,0}} \\ \mathbf{r}_{i^F_{f,1}} \\ \mathbf{r}_{i^F_{f,2}} \end{pmatrix}$, the first variation of the performance factor is then given by

$$\delta\omega_f = 4\frac{\left[\|\mathbf{Z}_f\| + \hat{e}_1^T\mathbf{Z}_f\right]\hat{e}_2^T - \hat{e}_2^T\mathbf{Z}_f\left[\hat{e}_1^T + \frac{\mathbf{Z}_f^T}{\|\mathbf{Z}\|}\right]}{\left(\|\mathbf{Z}_f\| + \hat{e}_1^T\mathbf{Z}_f\right)^2 + \left(\hat{e}_2^T\mathbf{Z}_f\right)^2}\frac{\partial\mathbf{Z}_f}{\partial\hat{r}_f}\frac{\partial\hat{r}_f}{\partial\mathbf{r}_f}\frac{\partial\mathbf{r}_f}{\partial\mathbf{T}_f} \tag{7.31}$$

And

$$\delta\mathbf{Z}_f = \begin{bmatrix} \hat{r}_{i^F_{f,2}} + \hat{r}_{i^F_{f,1}} & \widetilde{[\hat{r}_{i^F_{f,1}}]}\hat{r}_{i^F_{f,2}} \\ \hat{r}_{i^F_{f,0}} + \hat{r}_{i^F_{f,2}} & \widetilde{[\hat{r}_{i^F_{f,2}}]}\hat{r}_{i^F_{f,0}} \\ \hat{r}_{i^F_{f,0}} + \hat{r}_{i^F_{f,1}} & \widetilde{[\hat{r}_{i^F_{f,0}}]}\hat{r}_{i^F_{f,1}} \end{bmatrix}^T \begin{pmatrix} \delta\hat{r}_{i^F_{f,0}} \\ \delta\hat{r}_{i^F_{f,1}} \\ \delta\hat{r}_{i^F_{f,2}} \end{pmatrix} \tag{7.32}$$

The first variation in either of the $\hat{r}_{i^F_{f,j}} = \frac{\mathbf{r}_{i^F_{f,j}}}{\|\mathbf{r}_{i^F_{f,j}}\|}$ $(j \in \{0,1,2\})$ is given by

$$\delta\hat{r}_{i^F_{f,j}} = \left(\frac{I_{33}}{\|\mathbf{r}_{i^F_{f,j}}\|} - \frac{\mathbf{r}_{i^F_{f,j}}\mathbf{r}^T_{i^F_{f,j}}}{\|\mathbf{r}_{i^F_{f,j}}\|^3}\right)\delta\mathbf{r}_{i^F_{f,j}} \tag{7.33}$$

$$= \frac{\partial\hat{r}_{i^F_{f,j}}}{\partial\mathbf{r}_{i^F_{f,j}}}\delta\mathbf{r}_{i^F_{f,j}} \tag{7.34}$$

Finally, since $\mathbf{r}_{i^F_{f,j}} = \mathbf{C}_{i^F_{f,j}} - \mathbf{r}$, it is clear that

$$\begin{pmatrix} \delta\mathbf{r}_{i^F_{f,0}} \\ \delta\mathbf{r}_{i^F_{f,1}} \\ \delta\mathbf{r}_{i^F_{f,2}} \end{pmatrix} = \begin{bmatrix} I_{33} & 0_{33} & 0_{33} \\ 0_{33} & I_{33} & 0_{33} \\ 0_{33} & 0_{33} & I_{33} \end{bmatrix} \begin{pmatrix} \delta\mathbf{C}_{i^F_{f,0}} \\ \delta\mathbf{C}_{i^F_{f,1}} \\ \delta\mathbf{C}_{i^F_{f,2}} \end{pmatrix} \tag{7.35}$$

$$= \begin{pmatrix} \delta\mathbf{C}_{i^F_{f,0}} \\ \delta\mathbf{C}_{i^F_{f,1}} \\ \delta\mathbf{C}_{i^F_{f,2}} \end{pmatrix} \tag{7.36}$$

$$= \delta\mathbf{T}_f \tag{7.37}$$

$\frac{\partial \mathbf{r}_{i^F_{f,0}}}{\partial \mathbf{T}_f}$**:** Since

$$\mathbf{r}_{i^F_{f,0}} = \mathbf{C}_{i^F_{f,0}} - \mathbf{r} \tag{7.38}$$

The sought-for partial simply reads

$$\frac{\partial \mathbf{r}_{i^F_{f,0}}}{\partial \mathbf{T}_f} = \begin{bmatrix} I_{33} & 0_{33} & 0_{33} \end{bmatrix} \tag{7.39}$$

$\frac{\partial \mathbf{F}_f}{\partial \mathbf{T}_f}$**:**

The facet dyad $F_f = \hat{n}_f \hat{n}_f^T$ is formed from the outer-product of the normalized surface normal with itself. It is obviously symmetric, so that it can be parametrized by

$$F_f = \begin{bmatrix} F_f(0,0) & F_f(0,1) & F_f(0,2) \\ . & F_f(1,1) & F_f(1,2) \\ . & . & F_f(2,2) \end{bmatrix} \tag{7.40}$$

Again,

$$F_f(i,j) = \hat{e}_q^T F_f \hat{e}_r \tag{7.41}$$

So the vector-form parametrization of F_f

$$\mathbf{F}_f = \begin{pmatrix} F_f(0,0) \\ F_f(1,1) \\ F_f(2,2) \\ F_f(0,1) \\ F_f(0,2) \\ F_f(1,2) \end{pmatrix} \tag{7.42}$$

can actually be rewritten as

$$\mathbf{F}_f = \begin{pmatrix} \hat{e}_0^T F_f \hat{e}_0 \\ \hat{e}_1^T F_f \hat{e}_1 \\ \hat{e}_2^T F_f \hat{e}_2 \\ \hat{e}_0^T F_f \hat{e}_1 \\ \hat{e}_0^T F_f \hat{e}_2 \\ \hat{e}_1^T F_f \hat{e}_2 \end{pmatrix} = \begin{pmatrix} \hat{e}_0^T \hat{n}_f \hat{n}_f^T \hat{e}_0 \\ \hat{e}_1^T \hat{n}_f \hat{n}_f^T \hat{e}_1 \\ \hat{e}_2^T \hat{n}_f \hat{n}_f^T \hat{e}_2 \\ \hat{e}_0^T \hat{n}_f \hat{n}_f^T \hat{e}_1 \\ \hat{e}_0^T \hat{n}_f \hat{n}_f^T \hat{e}_2 \\ \hat{e}_1^T \hat{n}_f \hat{n}_f^T \hat{e}_2 \end{pmatrix} = \begin{pmatrix} \hat{n}_f^T \hat{e}_0 \hat{e}_0^T \hat{n}_f \\ \hat{n}_f^T \hat{e}_1 \hat{e}_1^T \hat{n}_f \\ \hat{n}_f^T \hat{e}_2 \hat{e}_2^T \hat{n}_f \\ \hat{n}_f^T \hat{e}_0 \hat{e}_1^T \hat{n}_f \\ \hat{n}_f^T \hat{e}_0 \hat{e}_2^T \hat{n}_f \\ \hat{n}_f^T \hat{e}_1 \hat{e}_2^T \hat{n}_f \end{pmatrix} \tag{7.43}$$

Therefore,

$$\delta \mathbf{F}_f = \begin{pmatrix} 2\hat{n}_f^T \hat{e}_0 \hat{e}_0^T \\ 2\hat{n}_f^T \hat{e}_1 \hat{e}_1^T \\ 2\hat{n}_f^T \hat{e}_2 \hat{e}_2^T \\ \hat{n}_f^T \left(\hat{e}_0 \hat{e}_1^T + \hat{e}_1 \hat{e}_0^T \right) \\ \hat{n}_f^T \left(\hat{e}_0 \hat{e}_2^T + \hat{e}_2 \hat{e}_0^T \right) \\ \hat{n}_f^T \left(\hat{e}_1 \hat{e}_2^T + \hat{e}_2 \hat{e}_1^T \right) \end{pmatrix} \delta \hat{n}_f \tag{7.44}$$

$$= \frac{\partial \mathbf{F}_f}{\partial \hat{n}_f} \delta \hat{n}_f \tag{7.45}$$

Introducing the non-normalized surface normal $\mathbf{N}_f$,

$$\hat{n}_f = \frac{\mathbf{N}_f}{\|\mathbf{N}_f\|} \tag{7.46}$$

with

$$\mathbf{N}_f = \left(\mathbf{C}_{i^F_{f,1}} - \mathbf{C}_{i^F_{f,0}}\right) \times \left(\mathbf{C}_{i^F_{f,2}} - \mathbf{C}_{i^F_{f,0}}\right) \tag{7.47}$$

Hence

$$\delta\hat{n}_f = \left(\frac{I_{33}}{\|\mathbf{N}_f\|} - \frac{\mathbf{N}_f\mathbf{N}_f^T}{\|\mathbf{N}_f\|^3}\right)\delta\mathbf{N}_f \tag{7.48}$$

$$= \frac{\partial\hat{n}}{\partial\mathbf{N}_f}\delta\mathbf{N}_f \tag{7.49}$$

and

$$\delta\mathbf{N}_f = \left[[\widetilde{\mathbf{C}_{i^F_{f,2}} - \mathbf{C}_{i^F_{f,1}}}] \quad [\widetilde{\mathbf{C}_{i^F_{f,0}} - \mathbf{C}_{i^F_{f,2}}}] \quad [\widetilde{\mathbf{C}_{i^F_{f,1}} - \mathbf{C}_{i^F_{f,0}}}]\right] \begin{pmatrix} \delta\mathbf{C}_{i^F_{f,0}} \\ \delta\mathbf{C}_{i^F_{f,1}} \\ \delta\mathbf{C}_{i^F_{f,2}} \end{pmatrix} \tag{7.50}$$

$$= \frac{\partial\mathbf{N}_f}{\partial\mathbf{T}_f}\delta\mathbf{T}_f \tag{7.51}$$

Therefore, the sought-for partial is given by

$$\frac{\partial\mathbf{F}_f}{\partial\mathbf{T}_f} = \frac{\partial\mathbf{F}_f}{\partial\hat{n}_f}\frac{\partial\hat{n}_f}{\partial\mathbf{N}_f}\frac{\partial\mathbf{N}_f}{\partial\mathbf{T}_f} \tag{7.52}$$

7.2.1.2 First variation in $\mathbf{X}^E_e$

The first variation in $\mathbf{X}^E_e$ can be written as

$$\delta\mathbf{X}^E_e = \begin{pmatrix} \delta L_e \\ \delta\mathbf{r}_{i^E_{e,0}} \\ \delta\mathbf{E}_e \end{pmatrix} \tag{7.53}$$

$$= \begin{bmatrix} \frac{\partial L_e}{\partial\mathbf{A}_e} & \mathbf{0}_3^T & \mathbf{0}_3^T \\ \frac{\partial\mathbf{r}_{i^E_{e,0}}}{\partial\mathbf{A}_e} & 0_{33} & 0_{33} \\ \frac{\partial\mathbf{E}_e}{\partial\mathbf{A}_e} & \frac{\partial\mathbf{E}_e}{\partial\mathbf{T}_{i^{EF}_{e,0}}} & \frac{\partial\mathbf{E}_e}{\partial\mathbf{T}_{i^{EF}_{e,1}}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{A}_e \\ \delta\mathbf{T}_{i^{EF}_{e,0}} \\ \delta\mathbf{T}_{i^{EF}_{e,1}} \end{bmatrix} \tag{7.54}$$

where the current edge is formed by two control points whose coordinates are stacked in a vector $\mathbf{A}_e$

$$\mathbf{A}_e = \begin{pmatrix} \mathbf{C}_{i^E_{e,0}} \\ \mathbf{C}_{i^E_{e,1}} \end{pmatrix} \tag{7.55}$$

and $\mathbf{T}_{i^{EF}_{e,0}}$, $\mathbf{T}_{i^{EF}_{e,1}}$ hold the coordinates of the control points of the two facets adjacent to edge e.

$$\mathbf{B}_e = \begin{pmatrix} \mathbf{A}_e \\ \mathbf{T}_{i^{EF}_{e,0}} \\ \mathbf{T}_{i^{EF}_{e,1}} \end{pmatrix} \tag{7.56}$$

The different partials are now defined:

$\frac{\partial \mathbf{r}_{i^E_{e,0}}}{\partial \mathbf{A}_e}$**:** Since

$$\mathbf{r}_{i^E_{e,0}} = \mathbf{C}_{i^E_{e,0}} - \mathbf{r} \tag{7.57}$$

The sought-for partial simply reads

$$\frac{\partial \mathbf{r}_{i^E_{e,0}}}{\partial \mathbf{A}_e} = \begin{bmatrix} I_{33} & 0_{33} \end{bmatrix} \tag{7.58}$$

$\frac{\partial \mathbf{E}_e}{\partial \mathbf{A}_e}$**:** Following Werner and Scheeres' notation, the edge dyad E_e is formed from

$$E_e = \hat{n}_A \hat{n}^A_{12} + \hat{n}_B \hat{n}^B_{21} \tag{7.59}$$

Transforming this expression so as to make it consistent with our own notations,

$$E_e = \frac{1}{l_e} \left(\hat{n}_{i^{EF}_{e,1}} \hat{n}^T_{i^{EF}_{e,1}} - \hat{n}_{i^{EF}_{e,0}} \hat{n}^T_{i^{EF}_{e,0}} \right) [\widetilde{\mathbf{C}_{i^E_{e,1}} - \mathbf{C}_{i^E_{e,0}}}] \tag{7.60}$$

Here, the index $i^{EF}_{e,j}$ with $j \in \{0, 1\}$ refers to the index of the j-th facet associated with the e-th edge.

It is remarkable that E_e is a symmetric tensor [5]. It can thus be parametrized by

$$E_e = \begin{bmatrix} E_e(0,0) & E_e(0,1) & E_e(0,2) \\ . & E_e(1,1) & E_e(1,2) \\ . & . & E_e(2,2) \end{bmatrix} \tag{7.61}$$

It must be noted that for $(q, r) \in \{0, 1, 2\} \times \{0, 1, 2\}$,

$$E_e(q, r) = \hat{e}^T_q E_e \hat{e}_r \tag{7.62}$$

So the vector-form parametrization of E_e

$$\mathbf{E}_e = \begin{pmatrix} E_e(0,0) \\ E_e(1,1) \\ E_e(2,2) \\ E_e(0,1) \\ E_e(0,2) \\ E_e(1,2) \end{pmatrix} \tag{7.63}$$

can actually be rewritten as

$$\mathbf{E}_e = \begin{pmatrix} \hat{e}_0^T E_e \hat{e}_0 \\ \hat{e}_1^T E_e \hat{e}_1 \\ \hat{e}_2^T E_e \hat{e}_2 \\ \hat{e}_0^T E_e \hat{e}_1 \\ \hat{e}_0^T E_e \hat{e}_2 \\ \hat{e}_1^T E_e \hat{e}_2 \end{pmatrix} \tag{7.64}$$

Given an arbitrary component of the edge dyad $E_e(q,r) = \hat{e}_q^T E_e \hat{e}_r$ can be written

$$E_e(q,r) = \frac{1}{l_e} \hat{e}_q^T \left(\hat{n}_{i^{EF}_{e,1}} \hat{n}^T_{i^{EF}_{e,1}} - \hat{n}_{i^{EF}_{e,0}} \hat{n}^T_{i^{EF}_{e,0}} \right) [\widetilde{\mathbf{C}_{i^E_{e,1}} - \mathbf{C}_{i^E_{e,0}}}] \hat{e}_r \tag{7.65}$$

The first variation of $E_e(q,r)$ can be compactly expressed as

$$\delta E_e(q,r) = \frac{1}{l_e} \begin{bmatrix} -E_e(q,r) \\ M^T [\widetilde{\hat{e}_r}] \left[\left(\hat{e}_q^T \hat{n}_{i^{EF}_{e,1}} \right) \hat{n}_{i^{EF}_{e,1}} - \left(\hat{e}_q^T \hat{n}_{i^{EF}_{e,0}} \right) \hat{n}_{i^{EF}_{e,0}} \right] \\ - \left[\hat{e}_q \hat{n}^T_{i^{EF}_{e,0}} + \left(\hat{e}_q^T \hat{n}_{i^{EF}_{e,0}} \right) I_{33} \right] \mathbf{V}_r \\ \left[\hat{e}_q \hat{n}^T_{i^{EF}_{e,1}} + \left(\hat{e}_q^T \hat{n}_{i^{EF}_{e,1}} \right) I_{33} \right] \mathbf{V}_r \end{bmatrix}^T \begin{pmatrix} \delta l_e \\ \delta \mathbf{A}_e \\ \delta \hat{n}_{i^{EF}_{e,0}} \\ \delta \hat{n}_{i^{EF}_{e,1}} \end{pmatrix} \tag{7.66}$$

with $M = \begin{bmatrix} -I_{33} & I_{33} \end{bmatrix}$ and $\mathbf{V}_r = [\widetilde{\mathbf{C}_{i^E_{e,1}} - \mathbf{C}_{i^E_{e,0}}}] \hat{e}_r$.

Also,

$$\delta l_e = \frac{\left(\mathbf{C}_{i^E_{e,0}} - \mathbf{C}_{i^E_{e,1}} \right)^T \left(\delta \mathbf{C}_{i^E_{e,0}} - \delta \mathbf{C}_{i^E_{e,1}} \right)}{l_e} \tag{7.67}$$

$$= \frac{\left(\mathbf{C}_{i^E_{e,0}} - \mathbf{C}_{i^E_{e,1}} \right)^T}{l_e} \begin{bmatrix} I_{33} & -I_{33} \end{bmatrix} \delta \mathbf{A}_e \tag{7.68}$$

$$= \frac{\partial l_e}{\partial \mathbf{A}_e} \delta \mathbf{A}_e \tag{7.69}$$

and since $\hat{n}_{i^{EF}_{e,0}} = \frac{\mathbf{N}_{i^{EF}_{e,0}}}{\|\mathbf{N}_{i^{EF}_{e,0}}\|}$,

$$\delta\hat{n}_{i^{EF}_{e,j}} = \left(\frac{I_{33}}{\|\mathbf{N}_{i^{EF}_{e,j}}\|} - \frac{\mathbf{N}_{i^{EF}_{e,j}}\mathbf{N}^T_{i^{EF}_{e,j}}}{\|\mathbf{N}_{i^{EF}_{e,j}}\|^3} \right) \delta\mathbf{N}_{i^{EF}_{e,j}} \tag{7.70}$$

$$= \frac{\partial\hat{n}_{i^{EF}_{e,j}}}{\mathbf{N}_{i^{EF}_{e,j}}}\delta\mathbf{N}_{i^{EF}_{e,j}} \tag{7.71}$$

with

$$\delta\mathbf{N}_{i^{EF}_{e,j}} = \begin{bmatrix} \widetilde{[\mathbf{C}_{i^F_{i^{EF}_{e,j},2}} - \mathbf{C}_{i^F_{i^{EF}_{e,j},1}}]} & \widetilde{[\mathbf{C}_{i^F_{i^{EF}_{e,j},0}} - \mathbf{C}_{i^F_{i^{EF}_{e,j},2}}]} & \widetilde{[\mathbf{C}_{i^F_{i^{EF}_{e,j},1}} - \mathbf{C}_{i^F_{i^{EF}_{e,j},0}}]} \end{bmatrix} \begin{pmatrix} \delta\mathbf{C}_{i^F_{i^{EF}_{e,j},0}} \\ \delta\mathbf{C}_{i^F_{i^{EF}_{e,j},1}} \\ \delta\mathbf{C}_{i^F_{i^{EF}_{e,j},2}} \end{pmatrix} \tag{7.72}$$

$$= \frac{\partial\mathbf{N}_{i^{EF}_{e,j}}}{\partial\mathbf{T}_{i^{EF}_{e,j}}}\delta\mathbf{T}_{i^{EF}_{e,j}} \tag{7.73}$$

So the first variation in the parametrization of E_e is given by

$$\delta\mathbf{E}_e = \begin{pmatrix} \frac{\partial E(0,0)}{\partial\mathbf{B}_e} \\ \frac{\partial E(1,1)}{\partial\mathbf{B}_e} \\ \frac{\partial E(2,2)}{\partial\mathbf{B}_e} \\ \frac{\partial E(0,1)}{\partial\mathbf{B}_e} \\ \frac{\partial E(0,2)}{\partial\mathbf{B}_e} \\ \frac{\partial E(1,2)}{\partial\mathbf{B}_e} \end{pmatrix} \delta\mathbf{B}_e \tag{7.74}$$

$$= \frac{\partial\mathbf{E}_e}{\partial\mathbf{B}_e}\delta\mathbf{B}_e \tag{7.75}$$

$\frac{\partial L_e}{\partial\mathbf{A}_e}$**:**

From

$$L_e = \ln\left(\frac{r_{i_{e,0}} + r_{i_{e,1}} + l_e}{r_{i_{e,0}} + r_{i_{e,1}} - l_e}\right) \tag{7.76}$$

$$\delta L_e = \frac{\delta r_{i_{e,0}} + \delta r_{i_{e,1}} + \delta l_e}{r_{i_{e,0}} + r_{i_{e,1}} + l_e} - \frac{\delta r_{i_{e,0}} + \delta r_{i_{e,1}} - \delta l_e}{r_{i_{e,0}} + r_{i_{e,1}} - l_e} \tag{7.77}$$

Writing

$$\beta_e^+ = r_{i_{e,0}} + r_{i_{e,1}} + l_e \tag{7.78}$$

$$\beta_e^- = r_{i_{e,0}} + r_{i_{e,1}} - l_e \tag{7.79}$$

This first variation becomes

$$\delta L_e = \frac{\delta r_{i_{e,0}} + \delta r_{i_{e,1}} + \delta l_e}{\beta_e^+} - \frac{\delta r_{i_{e,0}} + \delta r_{i_{e,1}} - \delta l_e}{\beta_e^-} \tag{7.80}$$

$$= \begin{pmatrix} \frac{1}{\beta_e^+} + \frac{1}{\beta_e^-} \\ \frac{1}{\beta_e^+} - \frac{1}{\beta_e^-} \\ \frac{1}{\beta_e^+} - \frac{1}{\beta_e^-} \end{pmatrix}^T \begin{pmatrix} \delta l_e \\ \delta r_{i_{e,0}} \\ \delta r_{i_{e,1}} \end{pmatrix} \tag{7.81}$$

Since

$$\delta l_e = \frac{\partial l_e}{\partial \mathbf{A}_e} \delta \mathbf{A}_e \tag{7.82}$$

$$\delta r_{i_{e,j}} = \frac{\mathbf{r}_{i_{e,j}}^T}{r_{i_{e,j}}} \delta \mathbf{r}_{i_{e,j}} \tag{7.83}$$

$$= \frac{\partial r_{i_{e,j}}}{\partial \mathbf{r}_{i_{e,j}}} \delta \mathbf{r}_{i_{e,j}} \tag{7.84}$$

and

$$\frac{\partial \mathbf{r}_{i^E_{e,0}}}{\partial \mathbf{A}_e} = \begin{bmatrix} I_{33} & 0_{33} \end{bmatrix} \tag{7.85}$$

$$\frac{\partial \mathbf{r}_{i^E_{e,1}}}{\partial \mathbf{A}_e} = \begin{bmatrix} 0_{33} & I_{33} \end{bmatrix} \tag{7.86}$$

The first variation in L_e finally becomes

$$\delta L_e = \begin{pmatrix} \frac{1}{\beta_e^+} + \frac{1}{\beta_e^-} \\ \frac{1}{\beta_e^+} - \frac{1}{\beta_e^-} \\ \frac{1}{\beta_e^+} - \frac{1}{\beta_e^-} \end{pmatrix}^T \begin{bmatrix} \frac{\partial l_e}{\partial \mathbf{A}_e} \\ \frac{\partial r_{i_{e,0}}}{\partial \mathbf{r}_{i_{e,0}}} \frac{\partial \mathbf{r}_{i^E_{e,0}}}{\partial \mathbf{A}_e} \\ \frac{\partial r_{i_{e,1}}}{\partial \mathbf{r}_{i_{e,1}}} \frac{\partial \mathbf{r}_{i^E_{e,1}}}{\partial \mathbf{A}_e} \end{bmatrix} \delta \mathbf{A}_e \tag{7.87}$$

$$= \frac{\partial L_e}{\partial \mathbf{A}_e} \delta \mathbf{A}_e \tag{7.88}$$

7.2.2 First variation in the acceleration due to a change in the shape

From

$$\mathbf{a}(\mathbf{r}) = G\rho \left[\sum_{e=1}^{N_e} \mathbf{a}_e^E + \sum_{f=1}^{N_f} \mathbf{a}_f^F \right] \tag{7.89}$$

with

$$\mathbf{a}_e^E = -E_e \mathbf{r}_{i^E_{e,0}} L_e \tag{7.90}$$

$$\mathbf{a}_f^F = F_f \mathbf{r}_{i^F_{f,0}} \omega_f \tag{7.91}$$

The first variation in the acceleration caused by a change in the shape is given by

$$\delta\mathbf{a}\left(\mathbf{r}\right) = G\rho\left[\sum_{e=1}^{N_e}\delta\mathbf{a}_e^E + \sum_{f=1}^{N_f}\delta\mathbf{a}_f^F\right] \tag{7.92}$$

$$= G\rho\left[\sum_{e=1}^{N_e}\left[\frac{\partial\mathbf{a}_e^E}{\partial\mathbf{C}}\right] + \sum_{f=1}^{N_f}\left[\frac{\partial\mathbf{a}_f^F}{\partial\mathbf{C}}\right]\right]\delta\mathbf{C} \tag{7.93}$$

$$= \left[\frac{\partial\mathbf{a}}{\partial\mathbf{C}}\right]\delta\mathbf{C} \tag{7.94}$$

where

$$\delta\mathbf{a}_e^E = \left[E_e\mathbf{r}_{i_{e,0}^E} \quad L_eE_e \quad L_eR_{i_{e,0}^E}^E\right]\begin{pmatrix}\delta L_e\\ \delta\mathbf{r}_{i_{e,0}^E}\\ \delta\mathbf{E}_e\end{pmatrix} \tag{7.95}$$

$$= \left[\frac{\partial\mathbf{a}_e^E}{\partial\mathbf{X}_e^E}\right]\delta\mathbf{X}_e^E \tag{7.96}$$

$$= \left[\frac{\partial\mathbf{a}_e^E}{\partial\mathbf{X}_e^E}\right]\left[\frac{\partial\mathbf{X}_e^E}{\partial\mathbf{B}_e^E}\right]\left[\frac{\partial\mathbf{B}_e^E}{\partial\mathbf{C}}\right]\delta\mathbf{C} \tag{7.97}$$

and

$$\delta\mathbf{a}_f^F = \left[F_f\mathbf{r}_{i_{f,0}^F} \quad \omega_fF_f \quad \omega_fR_{i_{f,0}^F}^F\right]\begin{pmatrix}\delta\omega_f\\ \delta\mathbf{r}_{i_{f,0}^F}\\ \delta\mathbf{F}_f\end{pmatrix} \tag{7.98}$$

$$= \left[\frac{\partial\mathbf{a}_f^F}{\partial\mathbf{X}_f^F}\right]\delta\mathbf{X}_f^F \tag{7.99}$$

$$= \left[\frac{\partial\mathbf{a}_f^F}{\partial\mathbf{X}_f^F}\right]\left[\frac{\partial\mathbf{X}_f^F}{\partial\mathbf{T}_f^F}\right]\left[\frac{\partial\mathbf{T}_f^F}{\partial\mathbf{C}}\right]\delta\mathbf{C} \tag{7.100}$$

7.3 Second moment about the mean of the PGM quantities

7.3.1 Variance in potential of polyhedron gravity model

The variance in the potential arising from the stochastic polyhedron gravity model can be obtained in a straightforward manner. Recall

$$U\left(\mathbf{r}\right) = \frac{G\rho}{2}\left[\sum_{e=1}^{N_e}U_e^E + \sum_{f=1}^{N_f}U_f^F\right] \tag{7.101}$$

with

$$U_e^E = \mathbf{r}_{i_{e,0}^E}^T E_e \mathbf{r}_{i_{e,0}^E} L_e \tag{7.102}$$

$$U_f^F = -\mathbf{r}_{i_{f,0}^F}^T F_f \mathbf{r}_{i_{f,0}^F} \omega_f \tag{7.103}$$

Since the first variation in the potential caused by a change in the shape has been found to be

$$\delta U(\mathbf{r}) = \frac{G\rho}{2}\left[\sum_{e=1}^{N_e}\left(\frac{\partial U_e^E}{\partial \mathbf{C}}\right) + \sum_{f=1}^{N_f}\left(\frac{\partial U_f^F}{\partial \mathbf{C}}\right)\right]\delta\mathbf{C} = \left(\frac{\partial U}{\partial \mathbf{C}}\right)\delta\mathbf{C} \tag{7.104}$$

where

$$\begin{aligned}
\delta U_e^E &= \begin{pmatrix} \mathbf{r}_{i_{e,0}^E}^T E_e \mathbf{r}_{i_{e,0}^E} \\ 2L_e E_e \mathbf{r}_{i_{e,0}^E} \\ L_e R_{i_{e,0}^E}^{ET} \mathbf{r}_{i_{e,0}^E} \end{pmatrix}^T \begin{pmatrix} \delta L_e \\ \delta \mathbf{r}_{i_{e,0}^E} \\ \delta \mathbf{E}_e \end{pmatrix} && (7.105)\\
&= \left(\frac{\partial U_e^E}{\partial \mathbf{X}_e^E}\right)\delta\mathbf{X}_e^E && (7.106)\\
&= \left(\frac{\partial U_e^E}{\partial \mathbf{X}_e^E}\right)\left[\frac{\partial \mathbf{X}_e^E}{\partial \mathbf{B}_e^E}\right]\left[\frac{\partial \mathbf{B}_e^E}{\partial \mathbf{C}}\right]\delta\mathbf{C} && (7.107)\\
&= \left(\frac{\partial U_e^E}{\partial \mathbf{C}}\right)\delta\mathbf{C} && (7.108)
\end{aligned}$$

and

$$\begin{aligned}
\delta U_f^F &= \begin{pmatrix} \mathbf{r}_{i_{f,0}^F}^T F_f \mathbf{r}_{i_{f,0}^F} \\ 2\omega_f F_f \mathbf{r}_{i_{f,0}^F} \\ \omega_f R_{i_{f,0}^F}^{FT} \mathbf{r}_{i_{f,0}^F} \end{pmatrix}^T \begin{pmatrix} \delta \omega_f \\ \delta \mathbf{r}_{i_{f,0}^F} \\ \delta \mathbf{F}_f \end{pmatrix} && (7.109)\\
&= \left(\frac{\partial U_f^F}{\partial \mathbf{X}_f^F}\right)\delta\mathbf{X}_f^F && (7.110)\\
&= \left(\frac{\partial U_f^F}{\partial \mathbf{X}_f^F}\right)\left[\frac{\partial \mathbf{X}_f^F}{\partial \mathbf{T}_f^F}\right]\left[\frac{\partial \mathbf{T}_f^F}{\partial \mathbf{C}}\right]\delta\mathbf{C} && (7.111)\\
&= \left(\frac{\partial U_f^F}{\partial \mathbf{C}}\right)\delta\mathbf{C} && (7.112)
\end{aligned}$$

the variance in the gravitational potential at a fixed point in space $\mathbf{r}$ is given by

$$\sigma_{UU}^2(\mathbf{r}) = \left(\frac{\partial U}{\partial \mathbf{C}}\right) P_{\mathbf{CC}} \left(\frac{\partial U}{\partial \mathbf{C}}\right)^T \tag{7.113}$$

7.3.2 Covariance in the acceleration

The covariance in the acceleration at a given point in space is given by

$$P_{\mathbf{aa}}(\mathbf{r}) = \left[\frac{\partial \mathbf{a}}{\partial \mathbf{C}}\right] P_{\mathbf{CC}} \left[\frac{\partial \mathbf{a}}{\partial \mathbf{C}}\right]^T \tag{7.114}$$

7.4 Gravitational slopes

The gravitational slope at the center of facet f is defined as

$$s_f = \arccos\left(-\hat{b}_f^T \hat{n}_f\right) \tag{7.115}$$

$$= \arccos(-u) \tag{7.116}$$

s_f is equal to 0 if the body-fixed acceleration direction $\hat{b}_f = \mathbf{b}_f / \|\mathbf{b}_f\|$ evaluated at the center of the facet $\mathbf{P}_f$ and the facet normal direction $\hat{n}_f$ are equal. The body-fixed acceleration at the center of the f-th facet is given by

$$\mathbf{b}_f = \mathbf{a}_f - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\mathbf{P}_f - \mathbf{G})) \tag{7.117}$$

where $\mathbf{G}$ denotes the barycenter of the small body. Gravitational slopes are of utmost interest to small body science and engineering since they may be indicative of areas where material can settle on the small body surface [110] [58] [57]. The next section offers a short summary of the quantities of interest when investigating the variation in the slopes caused by uncertainties in the shape vertices coordinates and the small body rotation period. The actual expression of the first variation in the shape is then derived.

7.4.1 Inertia quantities of interest

The gravitational slope at the center of facet s_f is obviously tied to the local geometry through the facet normal $\hat{n}_f$, but also to the position of the center-of-mass of the body since the latter is undergoing rotation about an axis that goes through this point. This remark essentially ties together three important inertia quantities: the volume of the small body, its center-of-mass and its inertia tensor. Expressions of these quantities are thus provided to the reader in this section before moving on. Computing the volume, center-of-mass and inertia tensor of a constant density polyhedron boils down to accumulating the contribution of each of the N_f facets - instead, of each tetrahedron subtended by the facet - in an orderly fashion [103], in a similar manner to what was achieved in Chapter 6 of this thesis for higher-order surface elements. The expressions in this section are nonetheless specialized to triangular, planar, first-order surface elements.

7.4.1.1 Volume

The total volume of the polyhedron is given by

$$V = \sum_{f=1}^{N_f} \Delta V_f \tag{7.118}$$

where $\Delta V_f = \frac{1}{6} \begin{vmatrix} \mathbf{C}_{i^F_{f,0}} & \mathbf{C}_{i^F_{f,1}} & \mathbf{C}_{i^F_{f,2}} \end{vmatrix}$ designates the signed volume of the considered tetrahedron [103].

7.4.1.2 Center-of-mass

The coordinates of the constant-density polyhedron center-of-mass are given by

$$\mathbf{G} = \frac{1}{V} \sum_{f=1}^{N_f} \Delta V_f \Delta \mathbf{G}_f \tag{7.119}$$

where $\Delta \mathbf{G}_f$ stands for the barycenter of the tetrahedron subtended by the f-th facet [103]

$$\begin{aligned} \Delta \mathbf{G}_f &= \frac{1}{4} \left(\mathbf{C}_{i^F_{f,0}} + \mathbf{C}_{i^F_{f,1}} + \mathbf{C}_{i^F_{f,2}} \right) && (7.120) \\ &= \frac{1}{4} A \mathbf{T}_f && (7.121) \end{aligned}$$

with $A = \begin{bmatrix} I_{33} & I_{33} & I_{33} \end{bmatrix}$.

7.4.1.3 Inertia tensor

The unit-density inertia tensor of the whole shape about $(0,0,0)^T$ is given by [111]

$$[I]_{\mathbf{O}} = \sum_{f=1}^{N_f} [\Delta I]_{\mathbf{O},f} \tag{7.122}$$

where the contribution to the inertia tensor of every tetrahedron can be written as

$$[\Delta I]_{\mathbf{O}} = \frac{[\Delta I]_{\mathbf{O},f}}{\Delta V_f} \Delta V_f \tag{7.123}$$

and

$$\frac{[\Delta I]_{\mathbf{O},f}}{\Delta V_f} = -\frac{1}{20}\left(\widetilde{[\mathbf{C}_{f,0} + \mathbf{C}_{f,1} + \mathbf{C}_{f,2}]}^2 + \widetilde{[\mathbf{C}_{f,0}]}^2 + \widetilde{[\mathbf{C}_{f,1}]}^2 + \widetilde{[\mathbf{C}_{f,2}]}^2\right) \tag{7.124}$$

$$= -\frac{1}{20}\left(\widetilde{[A\mathbf{T}_f]}^2 + \widetilde{[A_0\mathbf{T}_f]}^2 + \widetilde{[A_1\mathbf{T}_f]}^2 + \widetilde{[A_2\mathbf{T}_f]}^2\right) \tag{7.125}$$

with

$$A_0 = \begin{bmatrix} I_{33} & 0_{33} & 0_{33} \end{bmatrix} \tag{7.126}$$

$$A_1 = \begin{bmatrix} 0_{33} & I_{33} & 0_{33} \end{bmatrix} \tag{7.127}$$

$$A_2 = \begin{bmatrix} 0_{33} & 0_{33} & I_{33} \end{bmatrix} \tag{7.128}$$

Just like in Chapter 6, the parametrization of the inertia tensor is denoted $\mathbf{I}$:

$$\mathbf{I} = \begin{pmatrix} [I]_{\mathbf{O}}(0,0) \\ [I]_{\mathbf{O}}(1,1) \\ [I]_{\mathbf{O}}(2,2) \\ [I]_{\mathbf{O}}(0,1) \\ [I]_{\mathbf{O}}(0,2) \\ [I]_{\mathbf{O}}(1,2) \end{pmatrix} \tag{7.129}$$

$$= \sum_{f=1}^{N_f} \begin{pmatrix} [\Delta I]_{\mathbf{O},f}(0,0) \\ [\Delta I]_{\mathbf{O},f}(1,1) \\ [\Delta I]_{\mathbf{O},f}(2,2) \\ [\Delta I]_{\mathbf{O},f}(0,1) \\ [\Delta I]_{\mathbf{O},f}(0,2) \\ [\Delta I]_{\mathbf{O},f}(1,2) \end{pmatrix} \tag{7.130}$$

with $[I]_{\mathbf{O}}(q,r) = \hat{e}_q^T [I]_{\mathbf{O}} \hat{e}_r$ for $(q,r) \in \{0,1,2\} \times \{0,1,2\}$

7.4.2 Partial derivative of the inertia quantities of interest

The expressions of the partial derivatives of the quantities of interest, specialized to triangular, planar, first-order surface elements are given hereunder

7.4.2.1 Volume

The partial derivative in the total volume relative to the shape is given by

$$\left(\frac{\partial V}{\partial \mathbf{C}}\right) = \left[\sum_{f=1}^{N_f}\left(\frac{\partial \Delta V_f}{\partial \mathbf{C}}\right)\right] \tag{7.131}$$

where

$$\left(\frac{\partial \Delta V_f}{\partial \mathbf{C}}\right) = \frac{1}{6}\left(\left[\mathbf{C}_{i^F_{f,1}} \times \mathbf{C}_{i^F_{f,2}}\right]^T \quad -\mathbf{C}_{i^F_{f,0}}\widetilde{[\mathbf{C}_{i^F_{f,2}}]} \quad \mathbf{C}_{i^F_{f,0}}\widetilde{[\mathbf{C}_{i^F_{f,1}}]}\right)\left[\frac{\partial \mathbf{T}_f}{\partial \mathbf{C}}\right] \tag{7.132}$$

7.4.2.2 Center of mass

The partial derivative in the barycenter with respect to the vertices coordinates is readily given by

$$\left[\frac{\partial \mathbf{G}}{\partial \mathbf{C}}\right] = \frac{1}{V}\sum_{f=1}^{N_f}\left[(\Delta \mathbf{G}_f - \mathbf{G})\left(\frac{\partial \Delta V_f}{\partial \mathbf{T}_f}\right) + \Delta V_f\left[\frac{\partial \Delta \mathbf{G}_f}{\partial \mathbf{T}_f}\right]\right]\left[\frac{\partial \mathbf{T}_f}{\partial \mathbf{C}}\right] \tag{7.133}$$

7.4.2.3 Inertia tensor parametrization

The first variation of the inertia tensor parametrization is written as

$$\delta \mathbf{I} = \begin{pmatrix} \delta\,[I]_{\mathbf{O}}\,(0,0) \\ \delta\,[I]_{\mathbf{O}}\,(1,1) \\ \delta\,[I]_{\mathbf{O}}\,(2,2) \\ \delta\,[I]_{\mathbf{O}}\,(0,1) \\ \delta\,[I]_{\mathbf{O}}\,(0,2) \\ \delta\,[I]_{\mathbf{O}}\,(1,2) \end{pmatrix} \tag{7.134}$$

$$= \sum_{f=1}^{N_f}\begin{pmatrix} \delta\,[\Delta I]_{\mathbf{O},f}\,(0,0) \\ \delta\,[\Delta I]_{\mathbf{O},f}\,(1,1) \\ \delta\,[\Delta I]_{\mathbf{O},f}\,(2,2) \\ \delta\,[\Delta I]_{\mathbf{O},f}\,(0,1) \\ \delta\,[\Delta I]_{\mathbf{O},f}\,(0,2) \\ \delta\,[\Delta I]_{\mathbf{O},f}\,(1,2) \end{pmatrix} \tag{7.135}$$

where

$$\delta\,[\Delta I]_{\mathbf{O},f}\,(q,r) = \delta\left(\Delta V_f \hat{e}_q^T \frac{[\Delta I]_{\mathbf{O},f}}{\Delta V_f}\hat{e}_r\right) \tag{7.136}$$

$$= \left[\hat{e}_q^T \frac{[\Delta I]_{\mathbf{O},f}}{\Delta V_f}\hat{e}_r\left[\frac{\partial \Delta V_f}{\partial \mathbf{T}_f}\right] + \Delta V_f\left[\frac{\partial\left(\hat{e}_q^T \frac{[\Delta I]_{\mathbf{O},f}}{\Delta V_f}\hat{e}_r\right)}{\partial \mathbf{T}_f}\right]\right]\left[\frac{\partial \mathbf{T}_f}{\partial \mathbf{C}}\right]\delta \mathbf{C} \tag{7.137}$$

and

$$\left[\frac{\partial\left(\hat{e}_q^T\frac{[\Delta I]_{\mathbf{O},f}}{\Delta V_f}\hat{e}_r\right)}{\partial\mathbf{T}_f}\right]=-\frac{1}{20}\mathbf{T}_f^T\Big[A^T\left(\widetilde{[\hat{e}_q]}\widetilde{[\hat{e}_r]}+\widetilde{[\hat{e}_r]}\widetilde{[\hat{e}_q]}\right)A$$

$$+A_0^T\left(\widetilde{[\hat{e}_q]}\widetilde{[\hat{e}_r]}+\widetilde{[\hat{e}_r]}\widetilde{[\hat{e}_q]}\right)A_0+A_1^T\left(\widetilde{[\hat{e}_q]}\widetilde{[\hat{e}_r]}+\widetilde{[\hat{e}_r]}\widetilde{[\hat{e}_q]}\right)A_1+A_2^T\left(\widetilde{[\hat{e}_q]}\widetilde{[\hat{e}_r]}+\widetilde{[\hat{e}_r]}\widetilde{[\hat{e}_q]}\right)A_2\Big] \quad (7.138)$$

7.4.3 Partial derivative of the gravitation slope relative to the shape coordinates and attitude

Taking the first variation of Equation (7.115),

$$\delta s_f=\frac{\delta u}{\sqrt{1-u^2}} \quad (7.139)$$

and

$$\delta u=\hat{n}_f^T\delta\hat{b}_f+\hat{b}_f^T\delta\hat{n}_f \quad (7.140)$$

$$=\begin{pmatrix}\hat{n}_f\\ \hat{b}_f\end{pmatrix}^T\begin{pmatrix}\delta\hat{b}_f\\ \delta\hat{n}_f\end{pmatrix} \quad (7.141)$$

$$=\begin{pmatrix}\hat{n}_f\\ \hat{b}_f\end{pmatrix}^T\begin{bmatrix}\frac{\partial\hat{b}_f}{\partial\boldsymbol{\omega}} & \frac{\partial\hat{b}_f}{\partial\mathbf{C}}\\ 0_{33} & \frac{\partial\hat{n}_f}{\partial\mathbf{C}}\end{bmatrix}\begin{pmatrix}\delta\boldsymbol{\omega}\\ \delta\mathbf{C}\end{pmatrix} \quad (7.142)$$

$\frac{\partial\hat{n}_f}{\partial\mathbf{C}}$ has already been found in 7.2.1.2, in addition to $\frac{\partial\hat{b}_f}{\partial\mathbf{b}_f}$, so only $\frac{\partial\mathbf{b}_f}{\partial\mathbf{C}}$ and $\frac{\partial\mathbf{b}_f}{\partial\boldsymbol{\omega}}$ need to be investigated. The first variation of (7.117) yields

$$\delta\mathbf{b}_f=\delta\mathbf{a}_f-\widetilde{[\boldsymbol{\omega}]}^2(\delta\mathbf{P}_f-\delta\mathbf{G})+\left([\boldsymbol{\omega}\times\widetilde{(\mathbf{P}_f-\mathbf{G})}]+\widetilde{[\boldsymbol{\omega}]}[\widetilde{\mathbf{P}_f-\mathbf{G}}]\right)\delta\boldsymbol{\omega} \quad (7.143)$$

$$=\left[[\boldsymbol{\omega}\times\widetilde{(\mathbf{P}_f-\mathbf{G})}]+\widetilde{[\boldsymbol{\omega}]}[\widetilde{\mathbf{P}_f-\mathbf{G}}]\quad\left[\frac{\partial\mathbf{a}_f}{\partial\mathbf{C}}\right]+\left[\frac{\partial\mathbf{a}_f}{\partial\mathbf{P}_f}\right]\left[\frac{\partial\mathbf{P}_f}{\partial\mathbf{T}_f}\right]\left[\frac{\partial\mathbf{T}_f}{\partial\mathbf{C}}\right]+\widetilde{[\boldsymbol{\omega}]}^2\left(\left[\frac{\partial\mathbf{G}}{\partial\mathbf{C}}\right]-\left[\frac{\partial\mathbf{P}_f}{\partial\mathbf{C}}\right]\right)\right]\begin{pmatrix}\delta\boldsymbol{\omega}\\ \delta\mathbf{C}\end{pmatrix} \quad (7.144)$$

where $\left[\frac{\partial\mathbf{a}_f}{\partial\mathbf{P}_f}\right]$ denote the gravity gradient matrix of the polyhedron gravity model evaluated at the reference facet center. $\left[\frac{\partial\mathbf{a}_f}{\partial\mathbf{C}}\right]$ has already been found. Also,

$$\left[\frac{\partial\mathbf{P}_f}{\partial\mathbf{C}}\right]=\frac{1}{3}\begin{bmatrix}I_{33} & I_{33} & I_{33}\end{bmatrix}\left[\frac{\partial\mathbf{T}_f}{\partial\mathbf{C}}\right] \quad (7.145)$$

Making the assumption that the body is rotating about an axis $\hat{e}$,

$$\boldsymbol{\omega}=\omega\hat{e} \quad (7.146)$$

If $\hat{e}$ is taken as the largest inertia axis, the small body is undergoing principal rotation and one can write

$$\boldsymbol{\omega} = \omega[\mathcal{BP}]\hat{e}_3 \tag{7.147}$$

where $\mathcal{P}$ and $\mathcal{B}$ respectively stand for the principal and current body-fixed frames, and $\hat{e}_3 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T$. Letting $[\mathcal{PB}]$ be parametrized by a set of Modified Rodrigues Parameters $\boldsymbol{\sigma}$ such that $[\mathcal{PB}] = [\mathcal{PB}]\,(\boldsymbol{\sigma})$, linearizing a perturbed DCM $[\mathcal{P}'\mathcal{B}]$ about a reference $[\mathcal{PB}]$ yields

$$[\mathcal{P}'\mathcal{P}] = I_{33} - 4[\widetilde{\delta\boldsymbol{\sigma}}] \tag{7.148}$$

As a result, the first variation in the angular velocity caused by a change in the spin rate and principal axes direction is given by

$$\delta\boldsymbol{\omega} = \begin{bmatrix} [\mathcal{BP}]\hat{e}_3 & -4\omega[\mathcal{BP}][\widetilde{\hat{e}_3}] \end{bmatrix} \begin{pmatrix} \delta\omega \\ \delta\boldsymbol{\sigma} \end{pmatrix} \tag{7.149}$$

$$= \left[\left(\frac{\partial\boldsymbol{\omega}}{\partial\omega}\right) \quad \left[\frac{\partial\boldsymbol{\omega}}{\partial\boldsymbol{\sigma}}\right]\right] \begin{pmatrix} \delta\omega \\ \delta\boldsymbol{\sigma} \end{pmatrix} \tag{7.150}$$

$$= \left[\left(\frac{\partial\boldsymbol{\omega}}{\partial\omega}\right) \quad \left[\frac{\partial\boldsymbol{\omega}}{\partial\boldsymbol{\sigma}}\right]\right] \begin{bmatrix} 1 & \mathbf{0}_{3N_C}^T \\ \mathbf{0}_3 & \left[\frac{\partial\boldsymbol{\sigma}}{\partial\mathbf{C}}\right] \end{bmatrix} \begin{pmatrix} \delta\omega \\ \delta\mathbf{C} \end{pmatrix} \tag{7.151}$$

$$= \left[\left(\frac{\partial\boldsymbol{\omega}}{\partial\omega}\right) \quad \left[\frac{\partial\boldsymbol{\omega}}{\partial\mathbf{C}}\right]\right] \begin{pmatrix} \delta\omega \\ \delta\mathbf{C} \end{pmatrix} \tag{7.152}$$

7.5 Results

7.5.1 25143 Itokawa

The methods developed in the previous chapter are demonstrated over asteroid Itokawa, subjected to significant deviations in its vertices coordinates. The model used to generate the vertices covariance in similar as in Chapter 6. The inputs used in the successive simulations are listed on Table 7.1. The shapes drawn from the Monte-Carlo and overlaid with the reference, unperturbed Itokawa-8 shape outline can be found on Figure 7.2 and Figure 7.3. The decrease on the correlation distance and its effect on the seemingly more erratic behavior of otherwise neighboring vertices is clear.

Table 7.1: Input parameters for the investigation of the uncertainty in Itokawa's polyhedron gravity

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Correlation distance (m)	100	100	100	200	200	200
Standard deviation in normal error (m)	10	10	10	10	10	10
Monte-Carlo samples	300	1000	3000	300	1000	3000

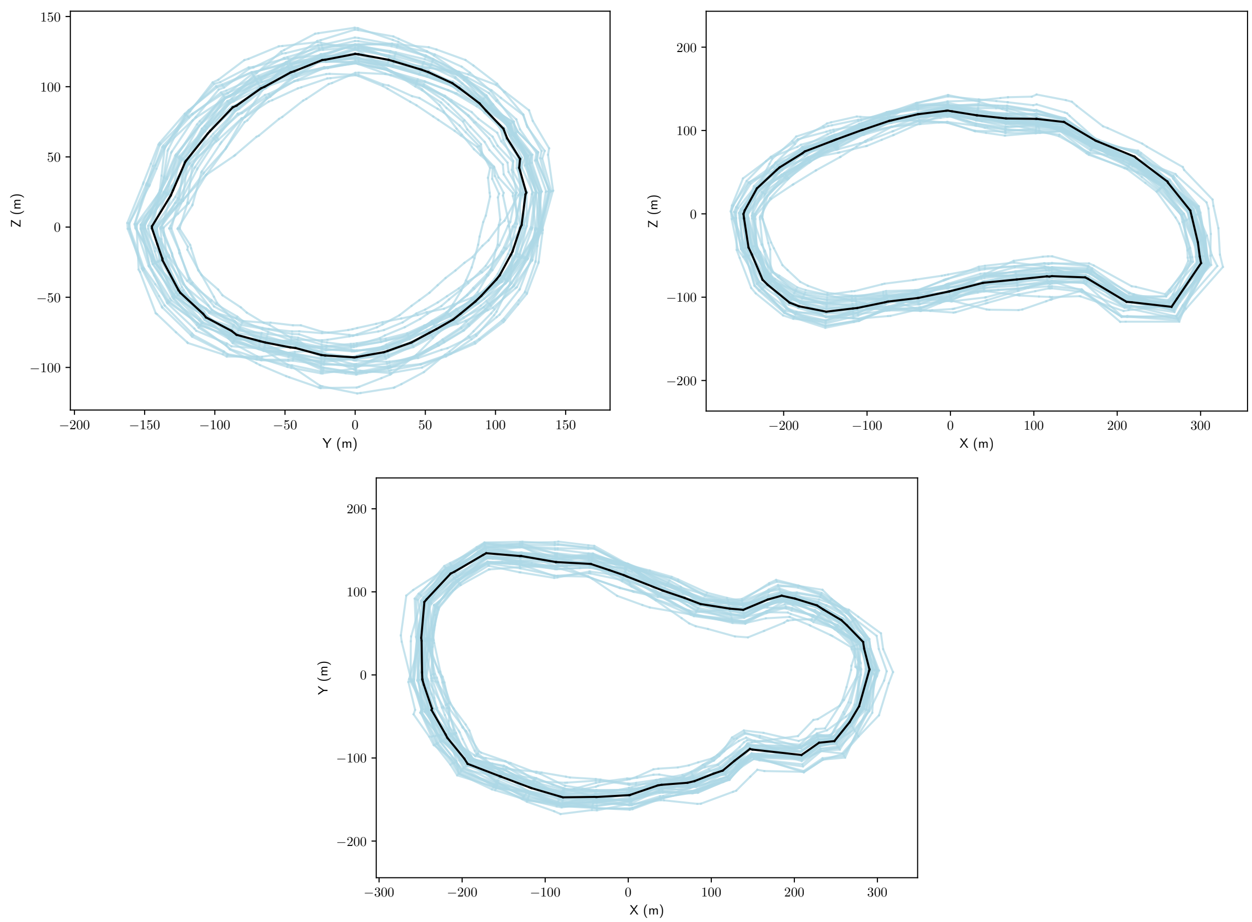


Figure 7.2: 30 outcomes from the Monte-Carlo sampling of the shape deviations (lightblue), overlaid with the reference Itokawa-8 shape model (black), (Case 1, 2 and 3)

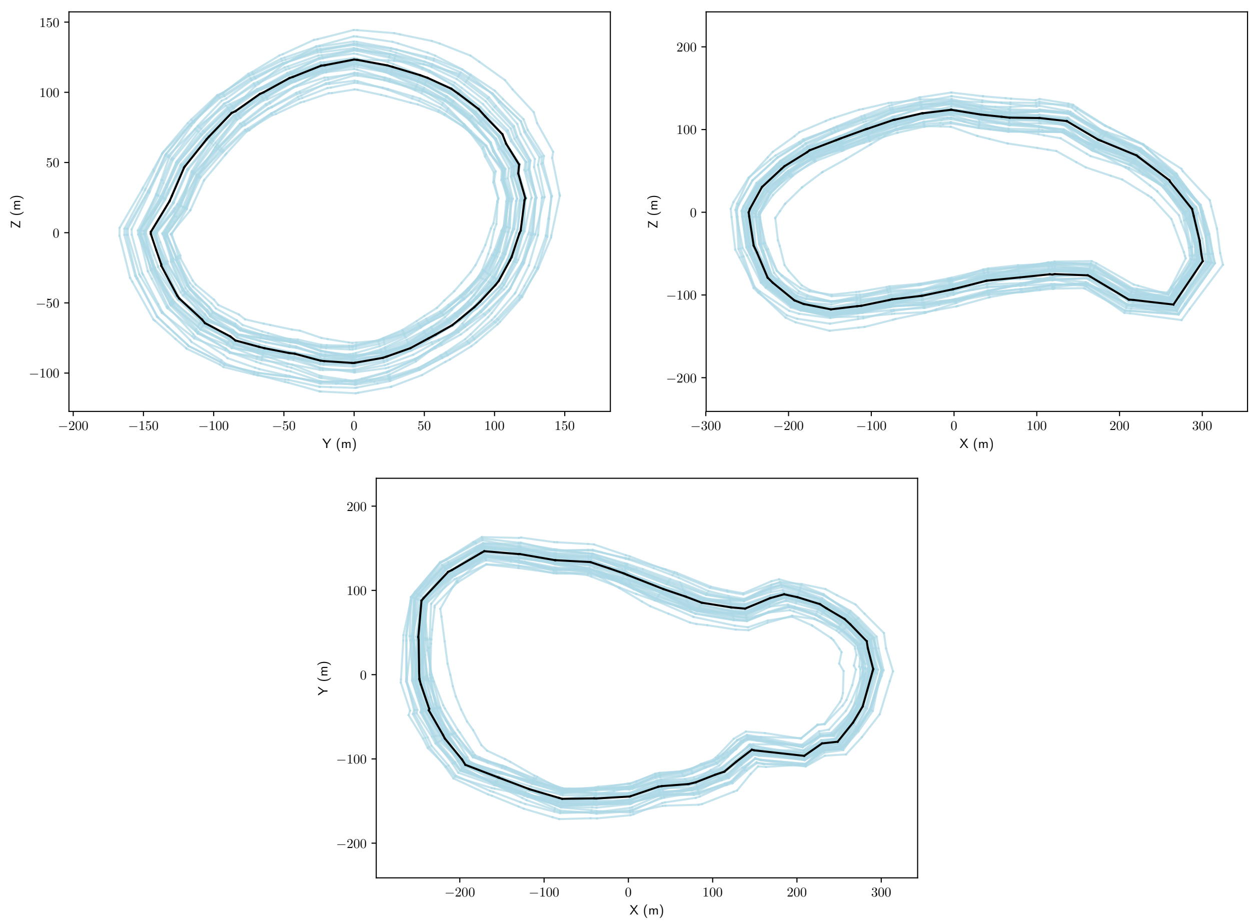


Figure 7.3: 30 outcomes from the Monte-Carlo sampling of the shape deviations (lightblue), overlaid with the reference Itokawa-8 shape model (black), (Case 4, 5 and 6)

The proposed uncertainty model enables the evaluation of the variance of the potential $\sigma_V^2(\mathbf{r})$ and the covariance of the acceleration $P_{\mathbf{a}}(\mathbf{r})$ at any point in space $\mathbf{r}$. The present section focuses on the acceleration. The model was first validated by comparing the predicted uncertainty levels to these obtained from a Monte-Carlo simulations where shape outcomes were randomly sampled and used to evaluate acceleration outcomes at selected positions. The acceleration covariances at the selected points were compared to the predicted ones so as to assess the accuracy of the uncertainty model relative to the Monte-Carlo runs, as in percentages of $\frac{\|P_{\mathbf{a}}(\mathbf{r})-P_{\mathbf{a}_{\mathrm{MC}}}(\mathbf{r})\|_2}{\mathrm{trace}\left(P_{\mathbf{a}_{\mathrm{MC}}}\right)}$ where $P_{\mathbf{a},\mathbf{MC}}(\mathbf{r})$ denotes the covariance in the Monte-Carlo acceleration outcomes and $\|.\|_2$ the usual L2-norm.

The agreement between the predicted covariance and the Monte-Carlo one measured by the L2 criterion is shown on Figures 7.4 through 7.9 . Outliers showing in red denote points for which the relative accuracy error was above the maximum value found in 90% of the other sampled points points, and were thus colored in a different color so as to avoid compressing the color scale. The agreement between the two statistical moments appears to be best away from the shape, with a rapid decay in the relative accuracy error as the query point moves away from the reference surface. Outliers appear to be systematically close to the reference surface, which appears to be consistent with the general trend of having higher relative accuracy errors in the acceleration uncertainty near the surface. The mean L2 errors over the selected positions are listed on Table 7.2, 7.3 and 7.4. It can be seen that the least error at each point is reached for either Case 3 and Case 6, both corresponding to the highest number of Monte-Carlo samples. This is an indication that the Monte-Carlo may have needed more samples to effectively converge. The worst outliers that were found in each of the cutting plane are the closest to the shape. The polyhedron gravity model as derived by Werner & Scheeres is singular along the edges of the shape of interest, which could explain why some, but not all of the points that were evaluated close to the surface present large prediction errors. The Kullback–Leibler (KL) divergence can also be used to obtain a measure of similarity between the two probability distributions $p_0(\mathbf{x}_0)$ and $p_1(\mathbf{x}_1)$, and is defined as [112]

$$D_{\mathrm{KL}}(p_0\|p_1) \equiv \mathrm{E}_{p_0}\left(\ln\left(\frac{\mathbf{x}_0}{\mathbf{x}_1}\right)\right) \tag{7.153}$$

Specialized to the case where the two considered probability density functions are n-dimensional Gaussians as in $p_0(\mathbf{x}_0) = \mathcal{N}_{\mathbf{x}_0}(\mathbf{m}_0, P_0)$ and $p_1(\mathbf{x}_1) = \mathcal{N}_{\mathbf{x}_1}(\mathbf{m}_1, P_1)$, the KL divergence becomes

$$D_{\mathrm{KL}}(\mathbf{m}_0, P_0 \| \mathbf{m}_1, P_1) = \frac{1}{2}\Big(\operatorname{trace}\left(P_1^{-1} P_0\right) + (\mathbf{m}_0 - \mathbf{m}_1)^T P_1^{-1} (\mathbf{m}_0 - \mathbf{m}_1) - n + \ln \frac{\det P_1}{\det P_0}\Big) \tag{7.154}$$

The agreement between the predicted covariance and the Monte-Carlo one measured by the L2 criterion is shown on Figures 7.4 through 7.9 . Outliers showing in red denote points for which the relative accuracy error was above the maximum value found in 90% of the other sampled points points, and were thus colored in a different color so as to avoid compressing the color scale. The agreement between the two statistical moments appears to be best away from the shape, with a rapid decay in the relative accuracy error as the query point moves away from the reference surface. Outliers appear to be systematically close to the reference surface, which appears to be consistent with the general trend of having higher relative accuracy errors in the acceleration uncertainty near the surface. The mean L2 errors over the selected positions are listed on Table 7.2, 7.3 and 7.4. It can be seen that the least error at each point is reached for either Case 3 and Case 6, both corresponding to the highest number of Monte-Carlo samples. This is an indication that the Monte-Carlo may have needed more samples to effectively converge. The worst outliers that were found in each of the cutting plane are the closest to the shape. The polyhedron gravity model as derived by Werner & Scheeres is singular along the edges of the shape of interest, which could explain why some, but not all of the points that were evaluated close to the surface present large prediction errors.

The KL divergence does not convey as much physical sense as the L2 accuracy error norm, but accounts for errors in both the mean of the distributions as well as their covariance. In a similar fashion to L2 case, the most extreme outliers are located close to the reference shape. Some selected points that are flagged as outliers are actually not quite as extreme as other much closer to the shape. In any case, the need for more Monte-Carlo samples is demonstrated on Tables 7.5, 7.6 and 7.7, since nearly every single point features a decreasing KL divergence as more samples are drawn. The proposed analytical uncertainty model, although linearized, is thus more efficient than the Monte-Carlo PGM evaluations to capture the uncertainty in the underlying gravity field.

Table 7.2: L2 error norm for selected points in the Y—Z plane

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$(0,-600,0)$	2.88	8.21	3.88	1.47	9.03	4.54
$(0,-500,0)$	3.87	8.10	3.92	1.64	8.99	4.57
$(0,-424,-424)$	2.70	7.80	2.58	1.57	8.70	3.70
$(0,-424,424)$	1.83	8.92	5.11	2.36	9.47	5.32
$(0,-400,0)$	5.14	7.97	3.94	2.60	8.96	4.60
$(0,-353,-353)$	3.79	7.58	2.49	1.44	8.59	3.61
$(0,-353,353)$	2.03	8.97	5.32	2.38	9.53	5.48
$(0,-300,0)$	6.28	7.78	3.91	4.04	8.95	4.60
$(0,-282,-282)$	5.26	7.26	2.39	2.36	8.42	3.48
$(0,-282,282)$	2.39	9.08	5.57	2.52	9.63	5.68
$(0,-212,-212)$	6.85	6.78	2.29	3.84	8.15	3.30
$(0,-212,212)$	3.09	9.36	5.83	2.87	9.85	5.94
$(0,0,-600)$	3.31	8.03	2.07	3.43	8.71	3.30
$(0,0,-500)$	3.47	7.91	2.03	3.41	8.60	3.18
$(0,0,-400)$	3.76	7.74	2.07	3.41	8.45	3.07
$(0,0,-300)$	4.31	7.44	2.24	3.49	8.20	2.96
$(0,0,300)$	3.04	9.19	5.82	3.16	9.89	6.00
$(0,0,400)$	2.87	9.21	5.66	3.18	9.76	5.77
$(0,0,500)$	2.82	9.21	5.45	3.23	9.69	5.57
$(0,0,600)$	2.82	9.19	5.25	3.27	9.63	5.41
$(0,212,-212)$	7.37	6.60	2.03	5.88	8.10	2.95
$(0,212,212)$	2.70	7.07	3.34	3.44	8.88	4.55
$(0,282,-282)$	6.99	7.48	2.03	5.62	8.52	3.13
$(0,282,282)$	3.26	8.03	3.82	3.89	9.22	4.72
$(0,300,0)$	5.53	5.90	1.69	5.32	8.14	3.30
$(0,353,-353)$	6.56	7.92	2.08	5.38	8.74	3.27
$(0,353,353)$	3.68	8.50	4.01	4.11	9.35	4.76
$(0,400,0)$	5.90	7.21	2.24	5.40	8.70	3.63
$(0,424,-424)$	6.15	8.16	2.16	5.17	8.86	3.39
$(0,424,424)$	3.92	8.73	4.08	4.20	9.41	4.75
$(0,500,0)$	5.96	7.89	2.57	5.35	8.96	3.81
$(0,600,0)$	5.85	8.25	2.77	5.24	9.10	3.91

Table 7.3: L2 error norm for selected points in the X—Z plane

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$(-600, 0, 0)$	3.24	8.75	4.79	2.20	9.63	5.11
$(-500, 0, 0)$	4.35	8.26	4.86	2.62	9.33	5.18
$(-424, 0, -424)$	2.76	9.52	3.70	2.37	9.95	4.38
$(-424, 0, 424)$	1.35	8.67	5.28	1.59	9.64	5.42
$(-400, 0, 0)$	6.43	7.32	4.81	3.63	8.68	5.22
$(-353, 0, -353)$	3.15	9.52	3.72	2.49	9.99	4.39
$(-353, 0, 353)$	1.44	8.22	5.39	1.43	9.44	5.51
$(-300, 0, 0)$	9.44	5.49	4.47	6.12	7.24	5.13
$(-282, 0, -282)$	3.75	9.49	3.74	2.81	10.03	4.40
$(-282, 0, 282)$	1.79	7.42	5.43	1.34	9.04	5.58
$(-212, 0, -212)$	4.51	9.59	3.91	3.48	10.21	4.47
$(-212, 0, 212)$	2.56	6.15	5.26	1.47	8.30	5.57
$(0, 0, -600)$	3.31	8.03	2.07	3.43	8.71	3.30
$(0, 0, -500)$	3.47	7.91	2.03	3.41	8.60	3.18
$(0, 0, -400)$	3.76	7.74	2.07	3.41	8.45	3.07
$(0, 0, -300)$	4.31	7.44	2.24	3.49	8.20	2.96
$(0, 0, 300)$	3.04	9.19	5.82	3.16	9.89	6.00
$(0, 0, 400)$	2.87	9.21	5.66	3.18	9.76	5.77
$(0, 0, 500)$	2.82	9.21	5.45	3.23	9.69	5.57
$(0, 0, 600)$	2.82	9.19	5.25	3.27	9.63	5.41
$(212, 0, -212)$	2.58	2.63	1.14	4.03	2.83	0.96
$(212, 0, 212)$	8.58	7.43	6.13	7.57	7.80	5.93
$(282, 0, -282)$	3.51	1.80	0.83	5.03	3.83	1.26
$(282, 0, 282)$	7.39	7.52	5.18	6.88	7.80	5.25
$(300, 0, 0)$	32.53	30.48	30.66	27.76	29.29	30.93
$(353, 0, -353)$	4.16	2.95	0.42	5.26	5.04	1.77
$(353, 0, 353)$	6.65	7.59	4.64	6.35	7.93	4.91
$(400, 0, 0)$	7.45	4.42	1.39	7.90	4.03	2.23
$(424, 0, -424)$	4.40	4.13	0.23	5.23	5.92	2.19
$(424, 0, 424)$	6.08	7.68	4.30	5.94	8.08	4.72
$(500, 0, 0)$	6.70	4.15	1.35	7.28	4.80	2.56
$(600, 0, 0)$	6.31	4.65	1.50	6.80	5.60	2.85

Table 7.4: L2 error norm for selected points in the X—Y plane

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$(-600, 0, 0)$	3.24	8.75	4.79	2.20	9.63	5.11
$(-500, 0, 0)$	4.35	8.26	4.86	2.62	9.33	5.18
$(-424, -424, 0)$	3.75	8.54	4.58	1.63	9.57	5.00
$(-424, 424, 0)$	5.07	9.02	4.09	4.37	9.77	4.68
$(-400, 0, 0)$	6.43	7.32	4.81	3.63	8.68	5.22
$(-353, -353, 0)$	5.08	8.10	4.64	2.82	9.39	5.05
$(-353, 353, 0)$	5.81	8.74	4.10	4.90	9.63	4.70
$(-300, 0, 0)$	9.44	5.49	4.47	6.12	7.24	5.13
$(-282, -282, 0)$	6.79	7.37	4.75	4.61	9.03	5.11
$(-282, 282, 0)$	7.03	8.27	3.98	5.90	9.35	4.71
$(-212, -212, 0)$	8.32	6.44	5.31	7.34	8.35	5.22
$(-212, 212, 0)$	9.03	7.77	3.55	8.17	8.93	4.70
$(0, -600, 0)$	2.88	8.21	3.88	1.47	9.03	4.54
$(0, -500, 0)$	3.87	8.10	3.92	1.64	8.99	4.57
$(0, -400, 0)$	5.14	7.97	3.94	2.60	8.96	4.60
$(0, -300, 0)$	6.28	7.78	3.91	4.04	8.95	4.60
$(0, 300, 0)$	5.53	5.90	1.69	5.32	8.14	3.30
$(0, 400, 0)$	5.90	7.21	2.24	5.40	8.70	3.63
$(0, 500, 0)$	5.96	7.89	2.57	5.35	8.96	3.81
$(0, 600, 0)$	5.85	8.25	2.77	5.24	9.10	3.91
$(212, -212, 0)$	4.76	5.19	2.35	5.78	5.76	3.41
$(212, 212, 0)$	12.19	6.60	3.30	8.42	6.28	3.49
$(282, -282, 0)$	4.05	5.06	2.31	5.30	6.12	3.47
$(282, 282, 0)$	10.22	6.29	2.45	7.81	6.56	3.22
$(300, 0, 0)$	32.53	30.48	30.66	27.76	29.29	30.93
$(353, -353, 0)$	3.67	5.46	2.39	4.92	6.60	3.56
$(353, 353, 0)$	8.90	6.40	2.18	7.25	6.94	3.23
$(400, 0, 0)$	7.45	4.42	1.39	7.90	4.03	2.23
$(424, -424, 0)$	3.44	5.91	2.49	4.65	7.01	3.65
$(424, 424, 0)$	8.00	6.64	2.14	6.80	7.29	3.32
$(500, 0, 0)$	6.70	4.15	1.35	7.28	4.80	2.56
$(600, 0, 0)$	6.31	4.65	1.50	6.80	5.60	2.85

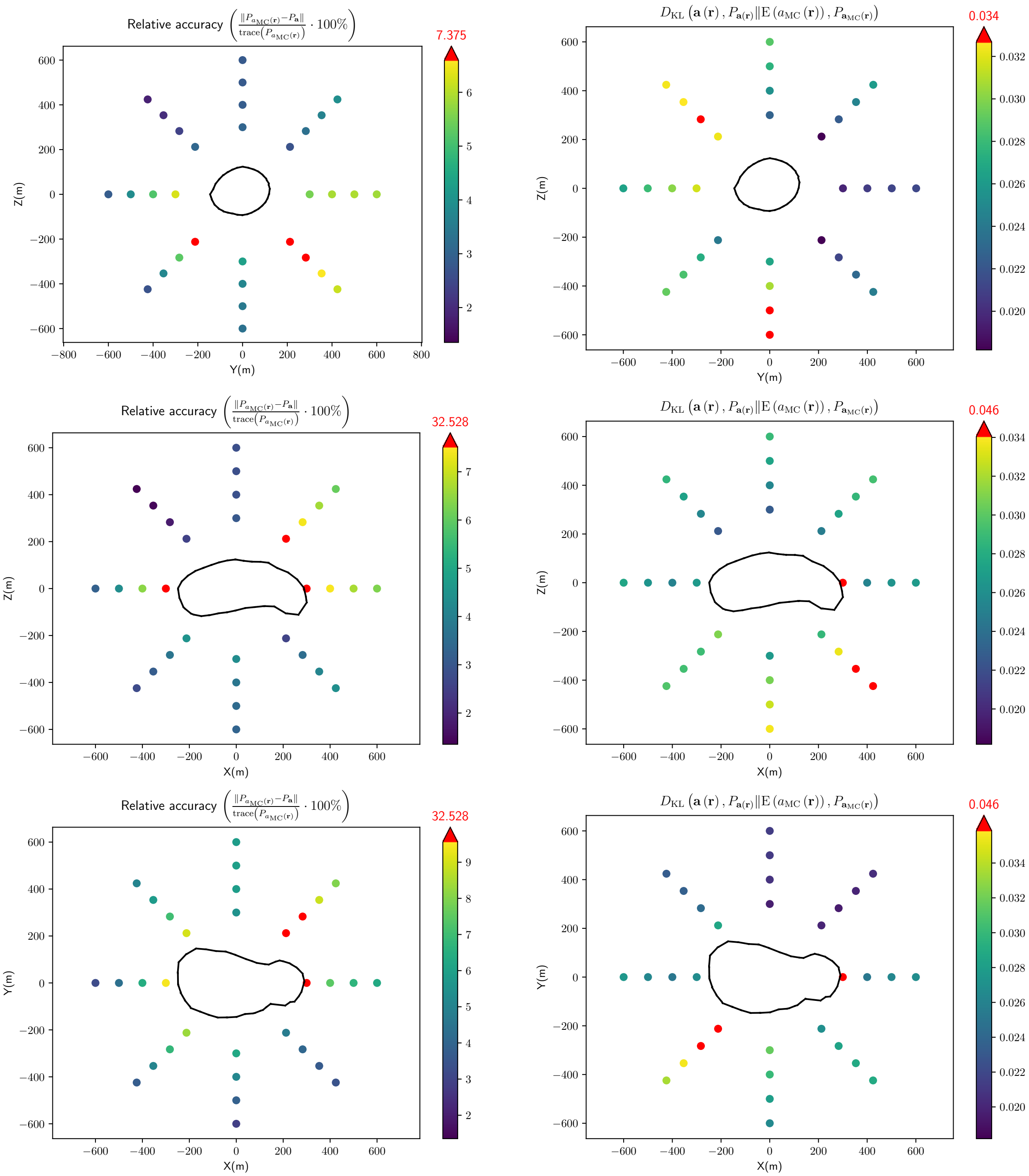


Figure 7.4: Accuracy of predicted uncertainty in acceleration relative to Monte-Carlo covariance, Case 1

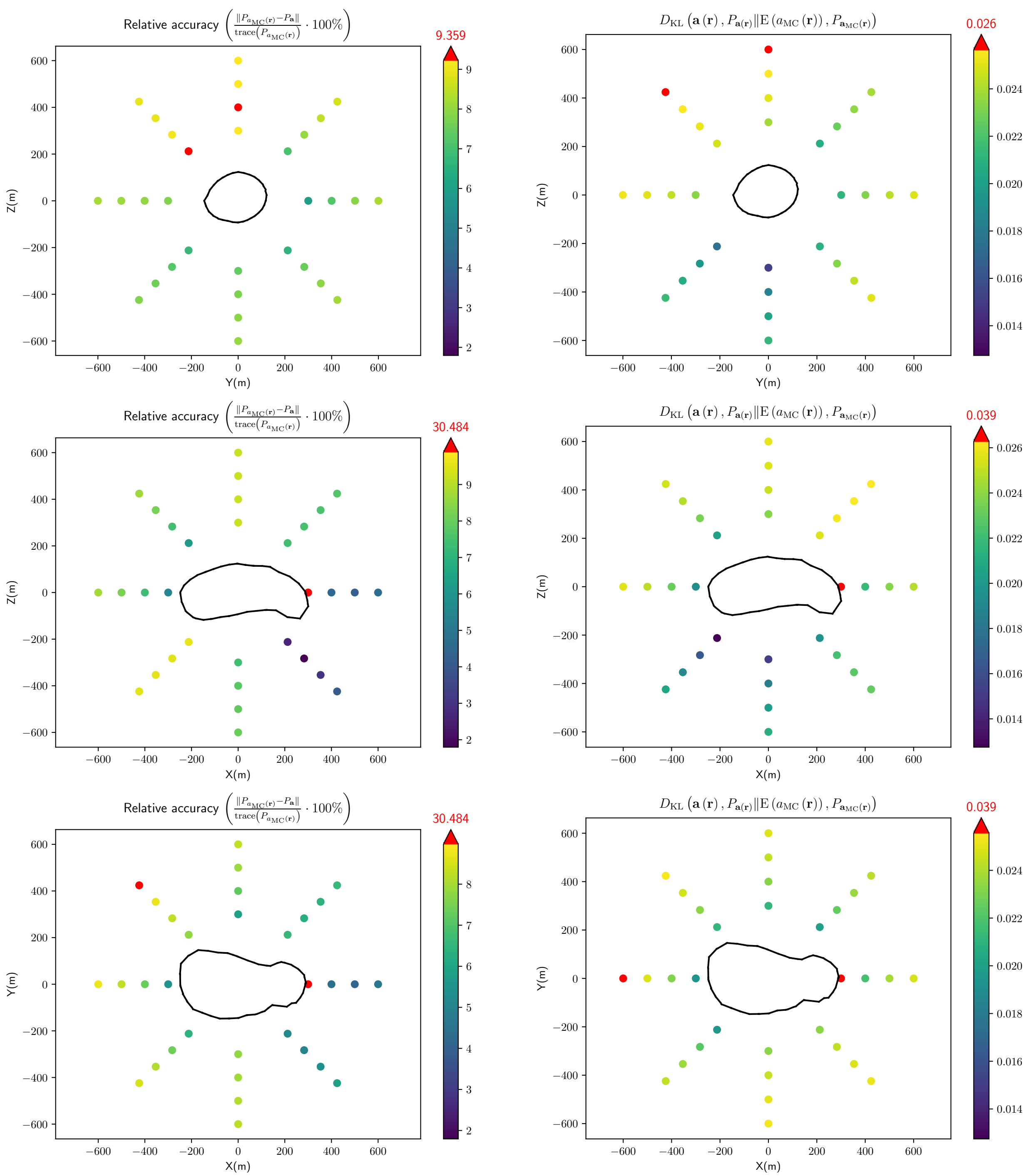


Figure 7.5: Accuracy of predicted uncertainty in acceleration relative to Monte-Carlo covariance, Case 2

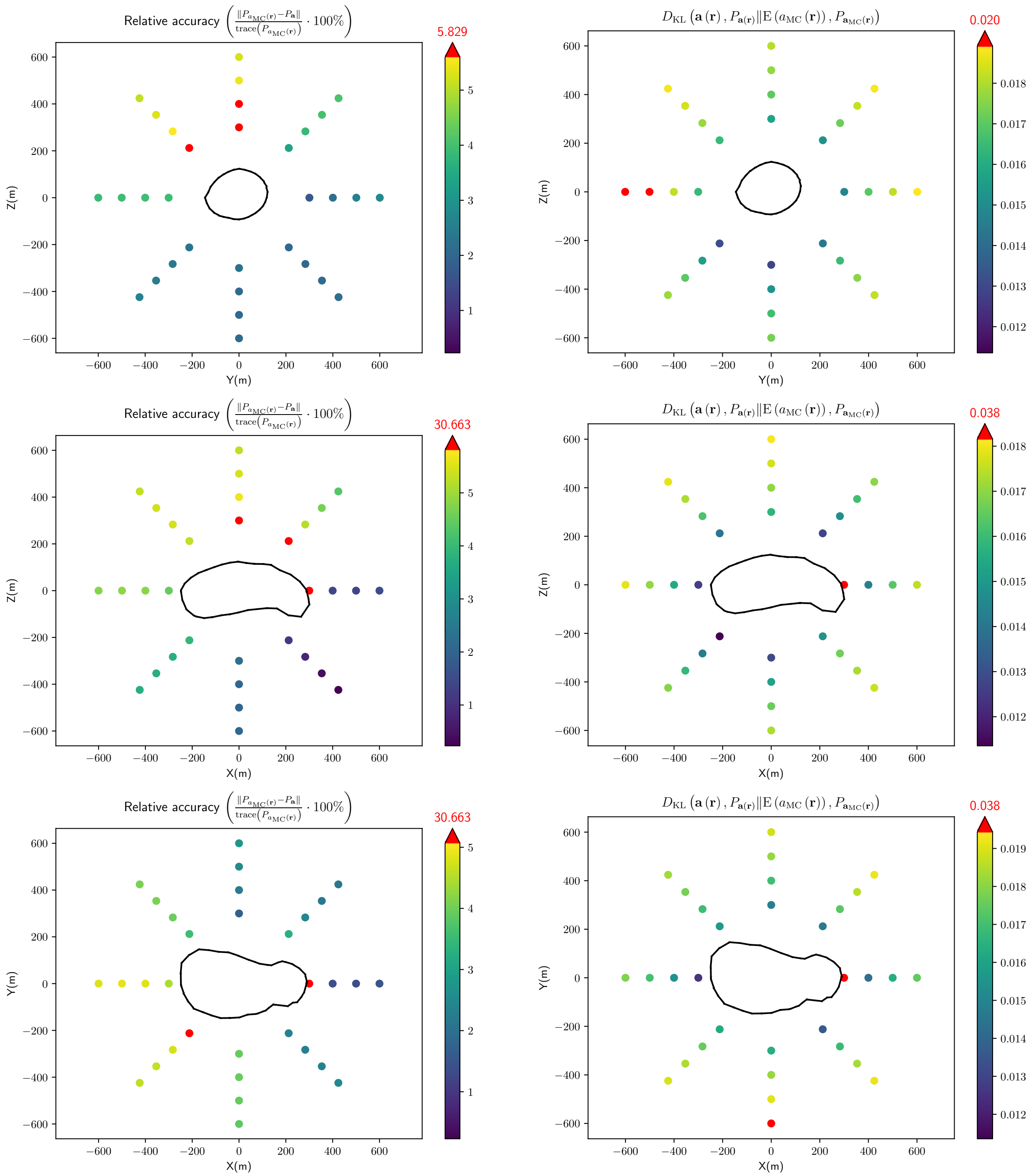


Figure 7.6: Accuracy of predicted uncertainty in acceleration relative to Monte-Carlo covariance, Case 3

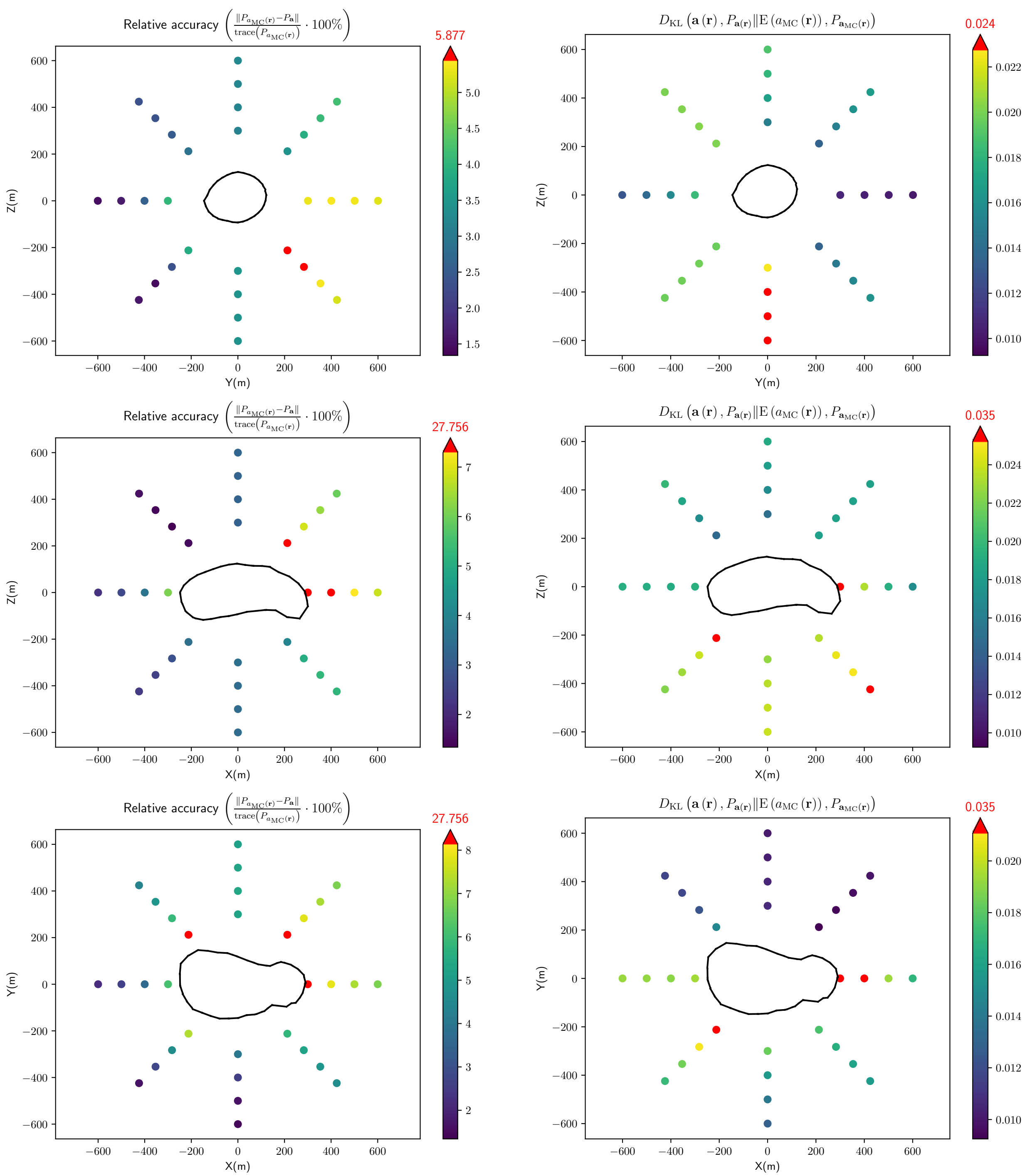


Figure 7.7: Accuracy of predicted uncertainty in acceleration relative to Monte-Carlo covariance, Case 4

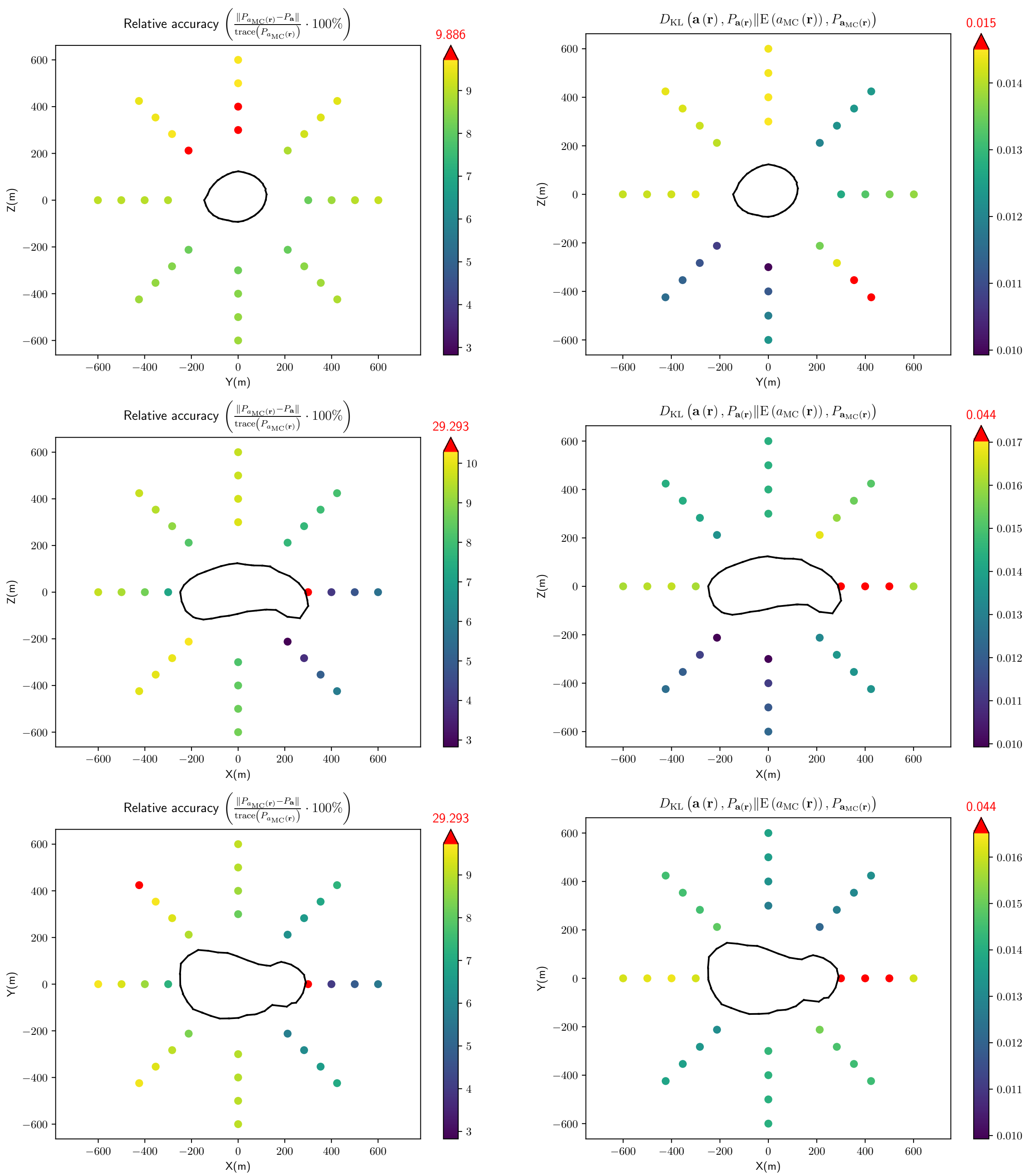


Figure 7.8: Accuracy of predicted uncertainty in acceleration relative to Monte-Carlo covariance, Case 5

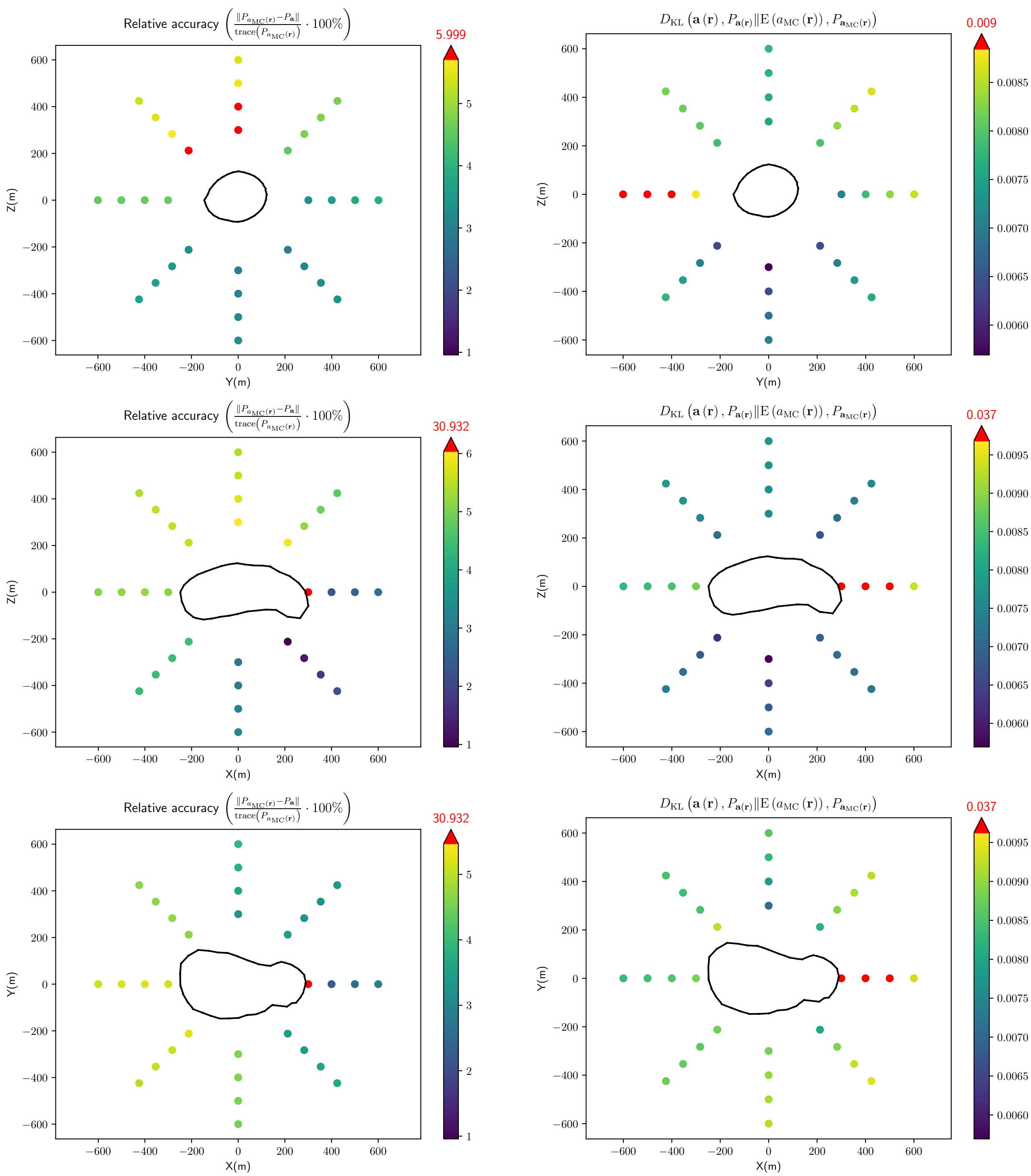


Figure 7.9: Accuracy of predicted uncertainty in acceleration relative to Monte-Carlo covariance, Case 6

Table 7.5: KL divergences for selected points in the Y—Z plane

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$(0, -600, 0)$	0.027	0.025	0.020	0.013	0.014	0.009
$(0, -500, 0)$	0.028	0.025	0.019	0.014	0.014	0.009
$(0, -424, -424)$	0.029	0.021	0.018	0.020	0.012	0.008
$(0, -424, 424)$	0.032	0.026	0.019	0.020	0.014	0.008
$(0, -400, 0)$	0.030	0.024	0.018	0.016	0.014	0.009
$(0, -353, -353)$	0.029	0.021	0.017	0.020	0.011	0.007
$(0, -353, 353)$	0.033	0.026	0.018	0.020	0.014	0.008
$(0, -300, 0)$	0.032	0.023	0.016	0.018	0.014	0.009
$(0, -282, -282)$	0.027	0.020	0.016	0.020	0.011	0.007
$(0, -282, 282)$	0.033	0.025	0.018	0.020	0.014	0.008
$(0, -212, -212)$	0.024	0.017	0.013	0.020	0.011	0.006
$(0, -212, 212)$	0.032	0.025	0.016	0.020	0.014	0.008
$(0, 0, -600)$	0.034	0.021	0.017	0.024	0.012	0.007
$(0, 0, -500)$	0.033	0.020	0.017	0.024	0.012	0.007
$(0, 0, -400)$	0.031	0.018	0.015	0.023	0.011	0.006
$(0, 0, -300)$	0.027	0.015	0.013	0.023	0.010	0.006
$(0, 0, 300)$	0.023	0.024	0.016	0.015	0.014	0.008
$(0, 0, 400)$	0.025	0.025	0.017	0.017	0.014	0.008
$(0, 0, 500)$	0.027	0.026	0.018	0.018	0.014	0.008
$(0, 0, 600)$	0.029	0.026	0.018	0.019	0.014	0.008
$(0, 212, -212)$	0.018	0.021	0.014	0.014	0.014	0.006
$(0, 212, 212)$	0.018	0.021	0.015	0.013	0.012	0.008
$(0, 282, -282)$	0.021	0.023	0.016	0.015	0.014	0.007
$(0, 282, 282)$	0.022	0.023	0.017	0.015	0.012	0.008
$(0, 300, 0)$	0.020	0.021	0.015	0.011	0.013	0.007
$(0, 353, -353)$	0.023	0.024	0.018	0.016	0.015	0.007
$(0, 353, 353)$	0.025	0.024	0.018	0.016	0.012	0.009
$(0, 400, 0)$	0.021	0.023	0.017	0.010	0.013	0.008
$(0, 424, -424)$	0.024	0.025	0.018	0.016	0.015	0.008
$(0, 424, 424)$	0.026	0.024	0.019	0.017	0.012	0.009
$(0, 500, 0)$	0.021	0.024	0.018	0.010	0.014	0.008
$(0, 600, 0)$	0.021	0.025	0.019	0.010	0.014	0.009

Table 7.6: KL divergences for selected points in the X—Z plane

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$(-600, 0, 0)$	0.028	0.026	0.018	0.019	0.016	0.008
$(-500, 0, 0)$	0.026	0.025	0.017	0.019	0.016	0.008
$(-424, 0, -424)$	0.030	0.021	0.017	0.022	0.012	0.007
$(-424, 0, 424)$	0.029	0.025	0.018	0.020	0.014	0.008
$(-400, 0, 0)$	0.025	0.023	0.015	0.019	0.016	0.009
$(-353, 0, -353)$	0.029	0.019	0.016	0.023	0.012	0.007
$(-353, 0, 353)$	0.027	0.025	0.017	0.019	0.014	0.008
$(-300, 0, 0)$	0.027	0.019	0.013	0.019	0.016	0.009
$(-282, 0, -282)$	0.029	0.017	0.014	0.024	0.011	0.007
$(-282, 0, 282)$	0.025	0.023	0.016	0.017	0.014	0.008
$(-212, 0, -212)$	0.031	0.013	0.011	0.027	0.010	0.006
$(-212, 0, 212)$	0.023	0.021	0.014	0.015	0.014	0.007
$(0, 0, -600)$	0.034	0.021	0.017	0.024	0.012	0.007
$(0, 0, -500)$	0.033	0.020	0.017	0.024	0.012	0.007
$(0, 0, -400)$	0.031	0.018	0.015	0.023	0.011	0.006
$(0, 0, -300)$	0.027	0.015	0.013	0.023	0.010	0.006
$(0, 0, 300)$	0.023	0.024	0.016	0.015	0.014	0.008
$(0, 0, 400)$	0.025	0.025	0.017	0.017	0.014	0.008
$(0, 0, 500)$	0.027	0.026	0.018	0.018	0.014	0.008
$(0, 0, 600)$	0.029	0.026	0.018	0.019	0.014	0.008
$(212, 0, -212)$	0.029	0.020	0.015	0.023	0.013	0.007
$(212, 0, 212)$	0.025	0.026	0.013	0.018	0.017	0.007
$(282, 0, -282)$	0.034	0.022	0.017	0.025	0.014	0.007
$(282, 0, 282)$	0.027	0.026	0.015	0.018	0.016	0.007
$(300, 0, 0)$	0.046	0.039	0.038	0.035	0.044	0.037
$(353, 0, -353)$	0.036	0.023	0.017	0.025	0.014	0.007
$(353, 0, 353)$	0.029	0.026	0.016	0.018	0.015	0.007
$(400, 0, 0)$	0.025	0.022	0.014	0.023	0.019	0.011
$(424, 0, -424)$	0.037	0.023	0.018	0.025	0.014	0.007
$(424, 0, 424)$	0.029	0.026	0.017	0.018	0.015	0.008
$(500, 0, 0)$	0.026	0.024	0.016	0.019	0.017	0.010
$(600, 0, 0)$	0.027	0.025	0.017	0.017	0.016	0.009

Table 7.7: KL divergences for selected points in the X—Y plane

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$(-600, 0, 0)$	0.028	0.026	0.018	0.019	0.016	0.008
$(-500, 0, 0)$	0.026	0.025	0.017	0.019	0.016	0.008
$(-424, -424, 0)$	0.034	0.024	0.019	0.017	0.014	0.009
$(-424, 424, 0)$	0.023	0.025	0.018	0.012	0.015	0.008
$(-400, 0, 0)$	0.025	0.023	0.015	0.019	0.016	0.009
$(-353, -353, 0)$	0.036	0.024	0.018	0.019	0.014	0.009
$(-353, 353, 0)$	0.024	0.025	0.018	0.012	0.015	0.008
$(-300, 0, 0)$	0.027	0.019	0.013	0.019	0.016	0.009
$(-282, -282, 0)$	0.039	0.022	0.018	0.021	0.013	0.009
$(-282, 282, 0)$	0.024	0.023	0.017	0.012	0.015	0.008
$(-212, -212, 0)$	0.044	0.020	0.016	0.026	0.013	0.009
$(-212, 212, 0)$	0.029	0.021	0.016	0.015	0.015	0.009
$(0, -600, 0)$	0.027	0.025	0.020	0.013	0.014	0.009
$(0, -500, 0)$	0.028	0.025	0.019	0.014	0.014	0.009
$(0, -400, 0)$	0.030	0.024	0.018	0.016	0.014	0.009
$(0, -300, 0)$	0.032	0.023	0.016	0.018	0.014	0.009
$(0, 300, 0)$	0.020	0.021	0.015	0.011	0.013	0.007
$(0, 400, 0)$	0.021	0.023	0.017	0.010	0.013	0.008
$(0, 500, 0)$	0.021	0.024	0.018	0.010	0.014	0.008
$(0, 600, 0)$	0.021	0.025	0.019	0.010	0.014	0.009
$(212, -212, 0)$	0.027	0.023	0.014	0.018	0.015	0.008
$(212, 212, 0)$	0.020	0.020	0.015	0.009	0.012	0.008
$(282, -282, 0)$	0.028	0.024	0.017	0.017	0.015	0.009
$(282, 282, 0)$	0.020	0.022	0.017	0.009	0.013	0.009
$(300, 0, 0)$	0.046	0.039	0.038	0.035	0.044	0.037
$(353, -353, 0)$	0.029	0.025	0.018	0.016	0.014	0.009
$(353, 353, 0)$	0.020	0.024	0.019	0.010	0.013	0.009
$(400, 0, 0)$	0.025	0.022	0.014	0.023	0.019	0.011
$(424, -424, 0)$	0.029	0.025	0.019	0.016	0.014	0.009
$(424, 424, 0)$	0.020	0.024	0.019	0.010	0.013	0.009
$(500, 0, 0)$	0.026	0.024	0.016	0.019	0.017	0.010
$(600, 0, 0)$	0.027	0.025	0.017	0.017	0.016	0.009

The predictive aspect of the uncertainty model is shown on Figures 7.10 and 7.11, that depict the relative uncertainty in the acceleration foreseen by the model, measured in percentages of $\sqrt{\frac{\mathrm{trace}(P_{\mathbf{a}}(\mathbf{r}))}{\|\mathbf{a}(\mathbf{r})\|}}$. These maps were obtained by sampling three uniform 10-meter spaced, orthogonal planar grids. It can be seen that the uncertainty in the acceleration is greatest close to the shape, and decreases as the distance to the surface increases. Moreover, the impact of the correlation length is apparent in the uncertainty levels. Increasing the correlation lengths causes the vertices deviations to interact in a constructive manner, effectively contributing to increasing the volume (hence the mass) uncertainty. It can be seen in all cases that the decay in the uncertainty is initially really quick upon leaving the shape, but becomes much slower as the distance increases.

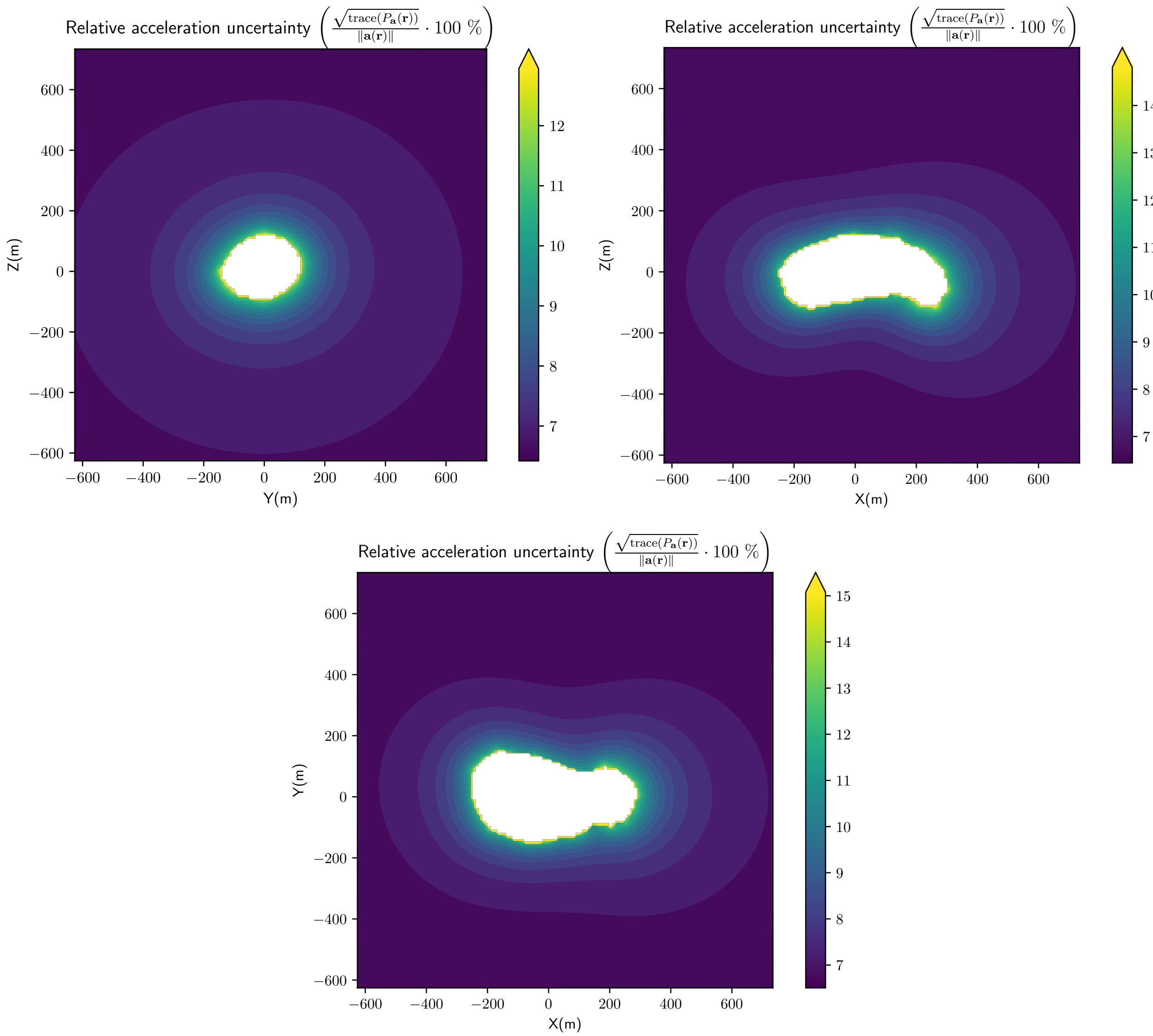


Figure 7.10: Relative uncertainty contours around Itokawa-8, with the inside of Itokawa shown in white (Case 1, 2 and 3)

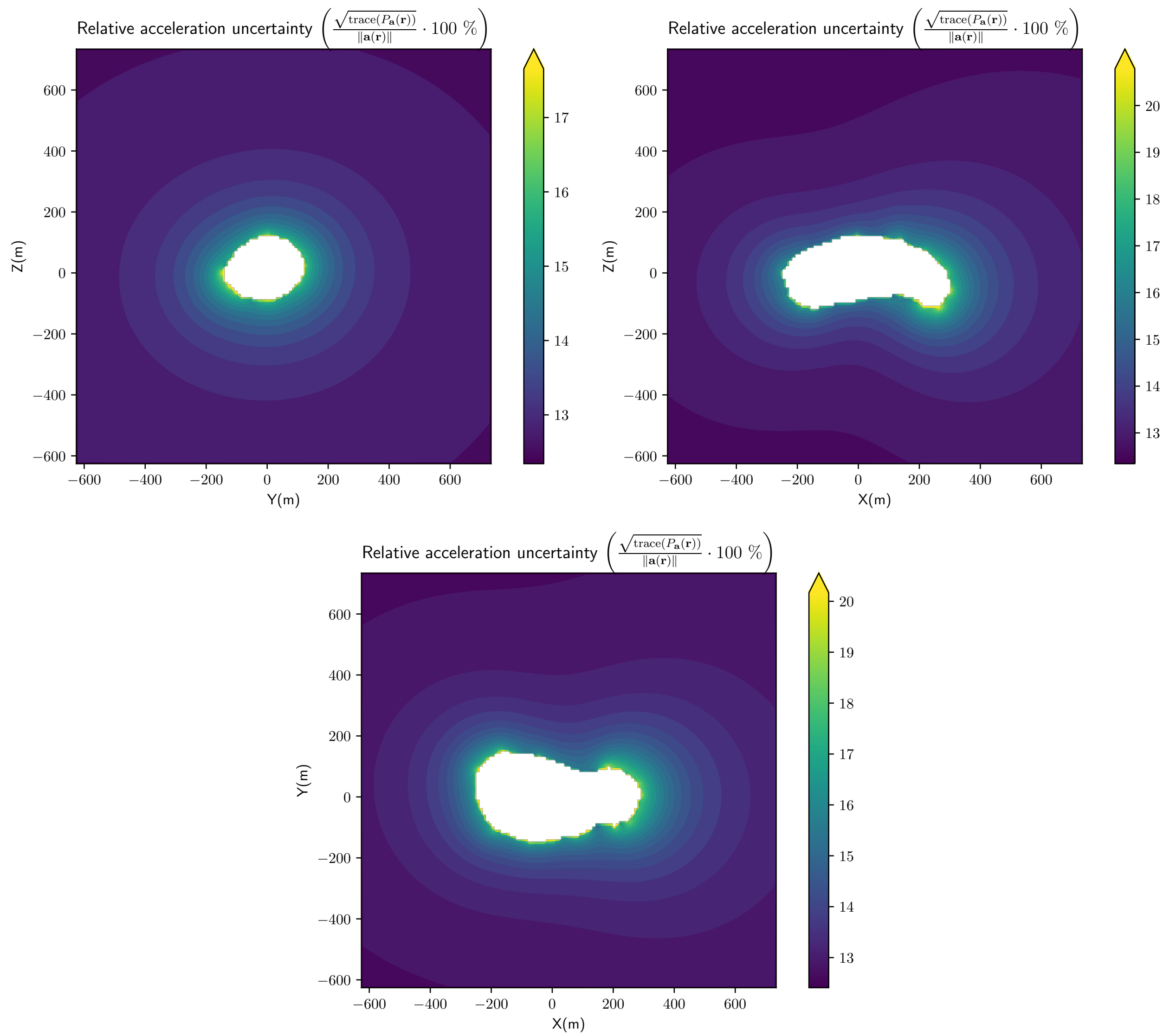


Figure 7.11: Relative uncertainty contours around Itokawa-8, with the inside of Itokawa shown in white (Case 4, 5 and 6)

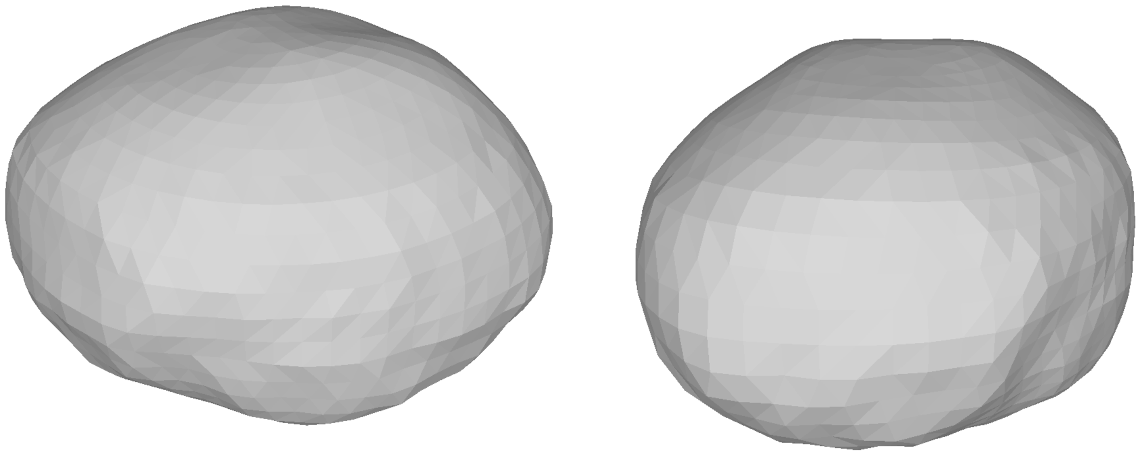


Figure 7.12: (16) Psyche shape model

7.5.2 (16) Psyche

The slope uncertainty model is now demonstrated over the surface of asteroid (16) Psyche, the target of the eponymous incoming NASA mission. Psyche is thought to be the remnant of a protoplanet core, and exhibits a visible/near-infrared characteristic of metal-rich M-class Asteroids [113]. The radar shape model of Psyche shown on Figure 7.12 was reconstructed based on radar observations collected by the Arecibo radio telescope, the only facility on Earth capable of collecting radar observations of main-belt objects [114]. The knowledge of Psyche is marked by fairly large uncertainties in the dimensions, density and pole directions, and thus represents an interesting test subject for the proposed gravity uncertainty model. The lack of information on the dynamical environment about Psyche is a strong mission design driver, and stable orbits are typically preferred over other trajectories that may be more fruitful from a science standpoint [115, 116]. Being able to quantify gravity uncertainties arising from the shape would help relax the mission design by providing more insight into the expected variability in the small body dynamical environment.

7.5.2.1 Foreword on covariance regularization

Uncertainty quantification over Psyche by means of the proposed method requires some special care. The considered shape model is comprised of $N_C = 1148$ vertices, where the Itokawa-8 model was only comprised of $N_C = 386$ vertices. This stark increase causes the associated shape covariance matrix to considerably grow in size. If memory availability is not a concern for ground-based computations, numerical errors very much are, as the extraction of the covariance square root by means of a Cholesky decomposition is likely to fail as the covariance matrix gets bigger. The spectral decomposition of the covariance used in 6.2 has been found to be generally more stable than the Cholesky equivalent, but requires the covariance matrix to remain positive semi-definite to be usable as is. This positive semi-definiteness can be violated either by construction of the covariance, or by numerical errors that creep-in during the eigenvalue computation process. The shape covariance matrix must thus be investigated before extracting its square root or running it through the linearized uncertainty quantification pipeline, and appropriate measures be taken so as to ensure that it remains positive semi-definite at all times.

A simple yet satisfying covariance regularization scheme consists in first computing the eigenvalue decomposition of the prescribed vertices covariance $P_{\mathbf{CC}}$ as in

$$P_{\mathbf{CC}} = UDU^T \tag{7.155}$$

where D is the diagonal matrix of eigenvalues, then by regularizing D itself. That is, a new diagonal matrix of eigenvalues D' is defined through

$$D'(i,i) = \max\left(0, D(i,i)\right) \ \forall i \in [0 \ldots 3N_C - 1] \tag{7.156}$$

The zero-clamped eigenvalue matrix D' can be combined with the original eigenvectors to produce a well-behaved covariance matrix through $P'_{\mathbf{CC}} = UD'U^T$ along with the covariance square root $\sqrt{P'_{\mathbf{CC}}} = U\sqrt{D'}U^T$.

The necessity of the covariance regularization scheme is now demonstrated over an example involving the Psyche shape model, where $P_{\mathbf{CC}}$ was generated with $\sigma = 5$ km and $l = 50$ km. $P_{\mathbf{CC}}$

was found to have 412 negative eigenvalues, with the largest one in absolute value in the order of 10^{-2}. These eigenvalues obviously need to be clamped to zero before extracting the square root. Figure 7.13 provide a side-by-side comparison of the original shape covariance next to the regularized one. It is striking to notice that the sparsity of the original covariance is lost through the regularization. Although this may seem to be a high price to pay, it nonetheless ensures that the covariance used in the linearized uncertainty quantification pipeline and the covariance square root used in the Monte-Carlo sampling of the shape are consistent with each other.

7.5.2.2 Gravity field uncertainty due to polar shape model errors

The actual uncertainty in the Psyche shape model is not uniformly distributed, but concentrated around the poles. This is a common issue in radar astronomy as a number of published asteroid shape models feature poor observability of high latitudes [57] [60] [117]. This paragraph thus explores the application of local uncertainty regions over the poles of Psyche. A local uncertainty region centered at $\mathbf{C}_c$ and characterized by a correlation distance l and standard deviation σ can be formed by adding the following 3x3 partition to the proper block in $P_{\mathbf{CC}}$:

$$P_{\mathbf{C}_i\mathbf{C}_j} \mathrel{+}= \sigma^2 e^{-\frac{1}{2l^2}\left(\|\mathbf{C}_i-\mathbf{C}_c\|^2+\|\mathbf{C}_j-\mathbf{C}_c\|^2+2\|\mathbf{C}_i-\mathbf{C}_j\|^2\right)}\hat{n}_i\hat{n}_j^T \tag{7.157}$$

The use of $+=$ allow uncertainty regions to overlap, provided that the initial covariance $P_{\mathbf{CC}}$ has all its components set to zero at initialization. This expression covers all cases, when $i = j$ as well as when $i \neq j$. This covariance matrix partition is set to zero should either of the distances $\|\mathbf{C}_i - \mathbf{C}_c\|$, $\|\mathbf{C}_j - \mathbf{C}_c\|$ or $\|\mathbf{C}_i - \mathbf{C}_j\|$ become greater than $3l$.

Inertial acceleration uncertainty The proposed method is employed to define two polar uncertainty regions on Psyche, at -90 and 90 degrees of latitude, centered about vertices 0 and 1147 respectively. Both regions feature the same noise standard deviation and correlation length, respectively set to 10 km and 75 km. 5000 Monte-Carlo shapes were drawn to construct the acceleration covariances and means to compare against the analytical prediction of the gravity uncertainty. The bulk density of the asteroid was set to 4500 $\mathrm{kg/m^3}$. A handful of the corresponding

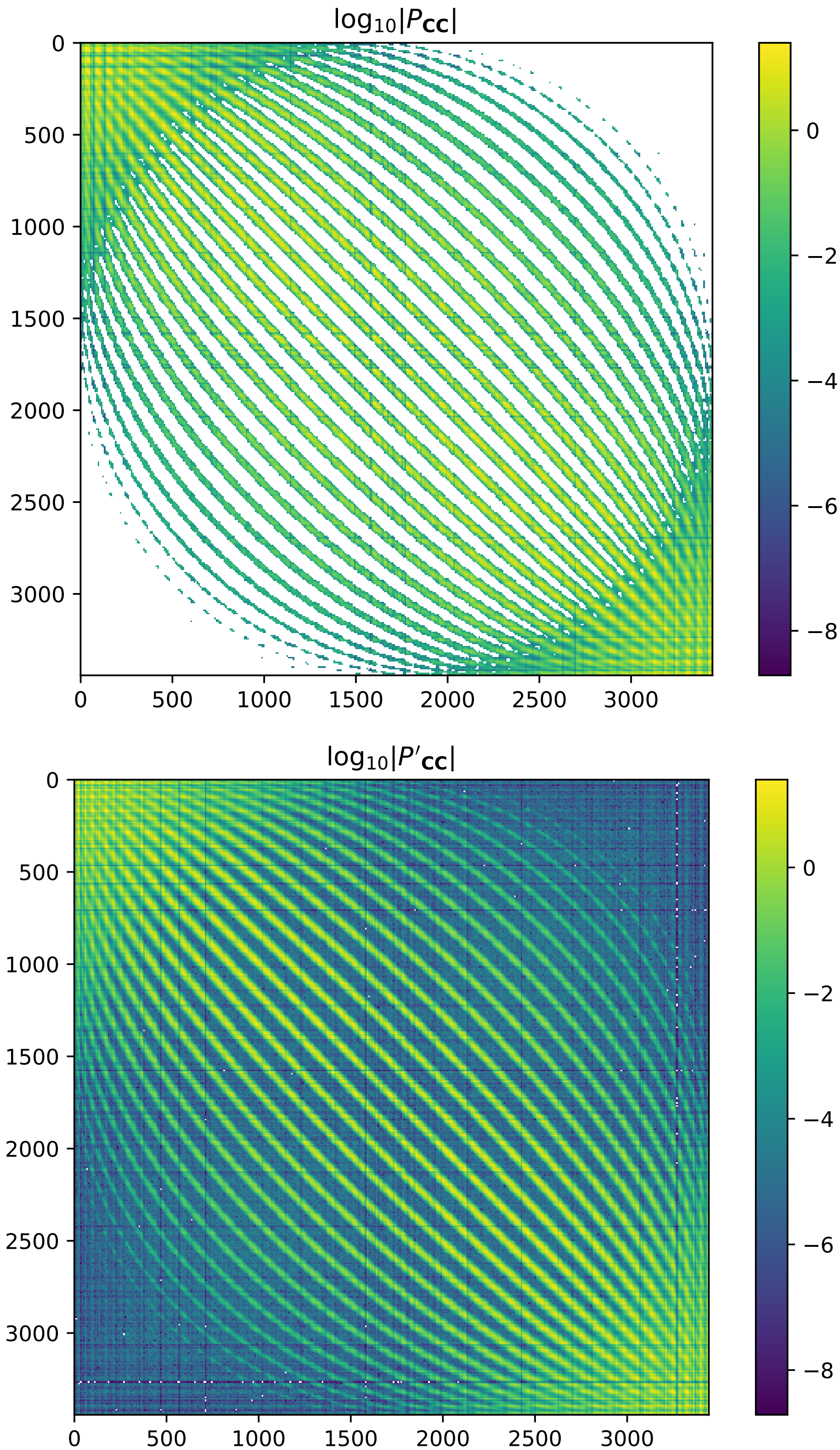
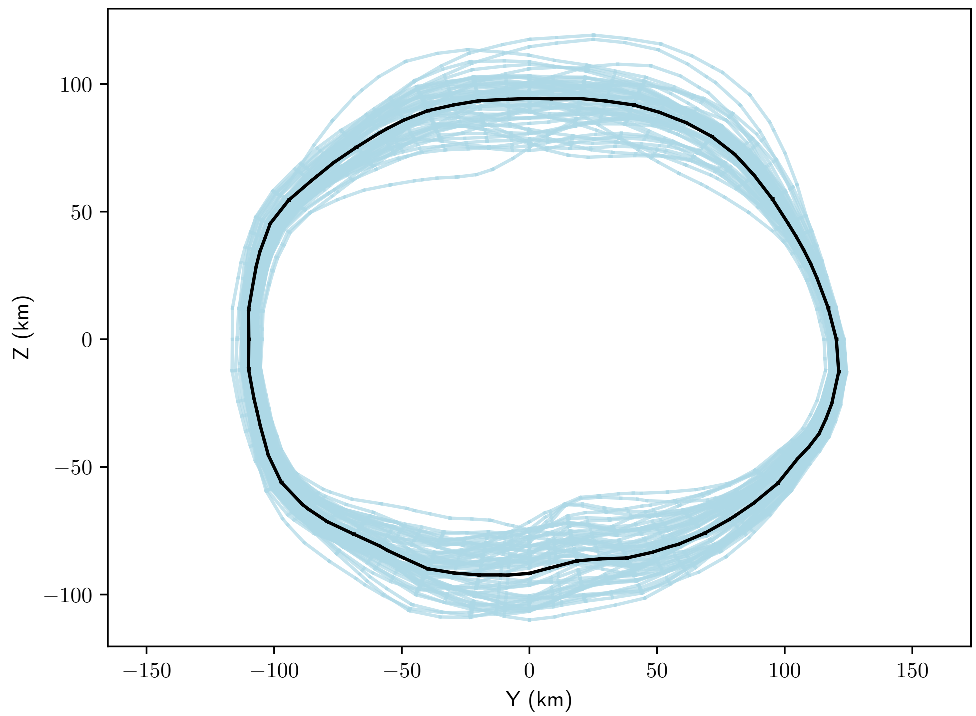


Figure 7.13: Top: $P_{\mathbf{CC}}$. Bottom: $P'_{\mathbf{CC}}$. Components of $P'_{\mathbf{CC}}$ less in absolute value than the minimum of $P'_{\mathbf{CC}}$ are shown in white

shape outcomes from the Monte-Carlo are shown on Figure 7.14. The validation of the uncertainty model in the X—Y plane is demonstrated on Figure 7.15, as the worst L2-norm covariance error is only 1.186 % . This is also confirmed by the KL divergence metric, which features very homogeneous values across the whole set of considered locations, as illustrated by the marginal outliers. The actual prediction in the gravity uncertainty in the X—Y plane is shown on Figure 7.16. The structure of the acceleration uncertainty map closely matches that of the underlying shape error, with a maximum uncertainty over the polar regions, but also features a constant background uncertainty of about 4% all around the body.

Inertial acceleration uncertainty, constant mass The same case as in the previous paragraph was run, but this time with the additional constraint that the mass of Psyche must remain constant as its shape varies. That is, $M = \rho V$ must be conserved. This can only happen if the first variation in the density satisfies $\delta\rho = -\delta V \rho / V$. The partials of the potential and acceleration were thus simply augmented with $-\frac{U(\mathbf{r})\rho}{V}\left(\frac{\partial V}{\partial \mathbf{C}}\right)$ and $-\frac{\mathbf{a}(\mathbf{r})\rho}{V}\left(\frac{\partial V}{\partial \mathbf{C}}\right)$ to account for this constraint, where the potential and acceleration are evaluated at the reference density. The error in the prediction of the uncertainty in the acceleration is shown on Figure 7.17. The agreement between the Monte-Carlo and the analytical model is confirmed. The analytical prediction in the uncertainty can be found on Figure 7.18. The major difference with the previous case lies in the rapid decay in the uncertainty as the queried point moves further away from the shape. The gravity field structure converges towards that of a point-mass as the point moves further away from the shape. Because the standard gravitational of the shape is constant, there is thus no uncertainty left in the point-mass gravity, causing the acceleration covariance to nullify away from the shape.

Surface gravity slopes The PGM uncertainty model can also applied to the quantification of the uncertainties in the surface slopes of Psyche. The rotation period of Psyche was set to 15105.4 s, in accordance with the reported estimate in [114]. This reference states that the rotation period is known within ± 0.000001 hours, which led to considering no uncertainty in the angular velocity magnitude, thus setting $\delta\omega = 0$. The two polar uncertainty regions are however sufficient to induce local and global uncertainties in the slope, by respectively affecting the facet normal and facet



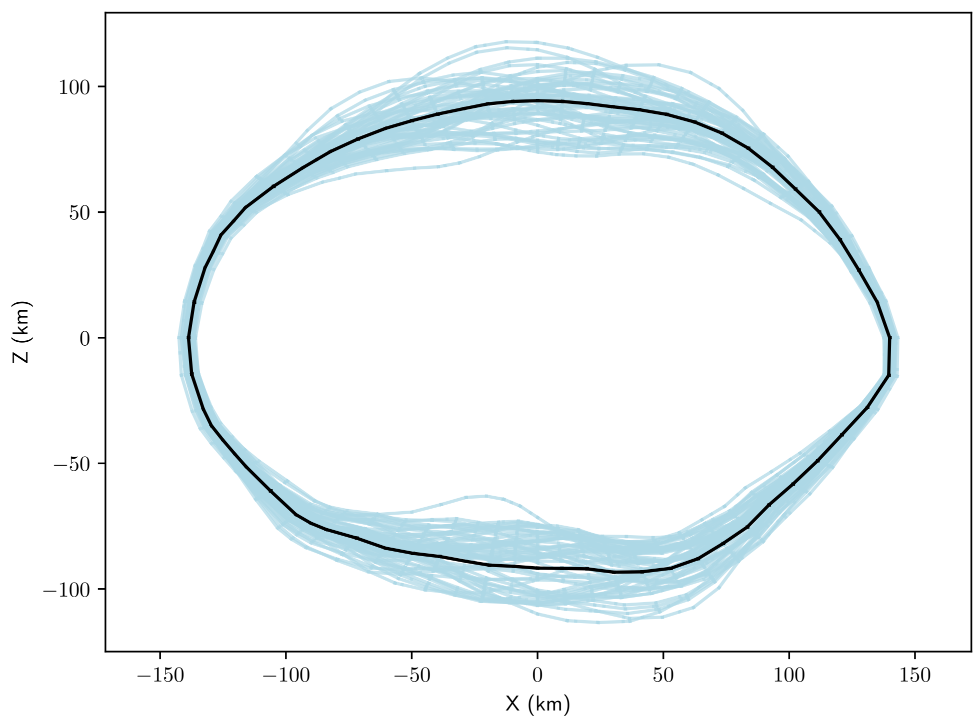


Figure 7.14: 60 Monte-Carlo shape outcomes (lightblue) overlaid over the reference (16) Psyche shape model (black)

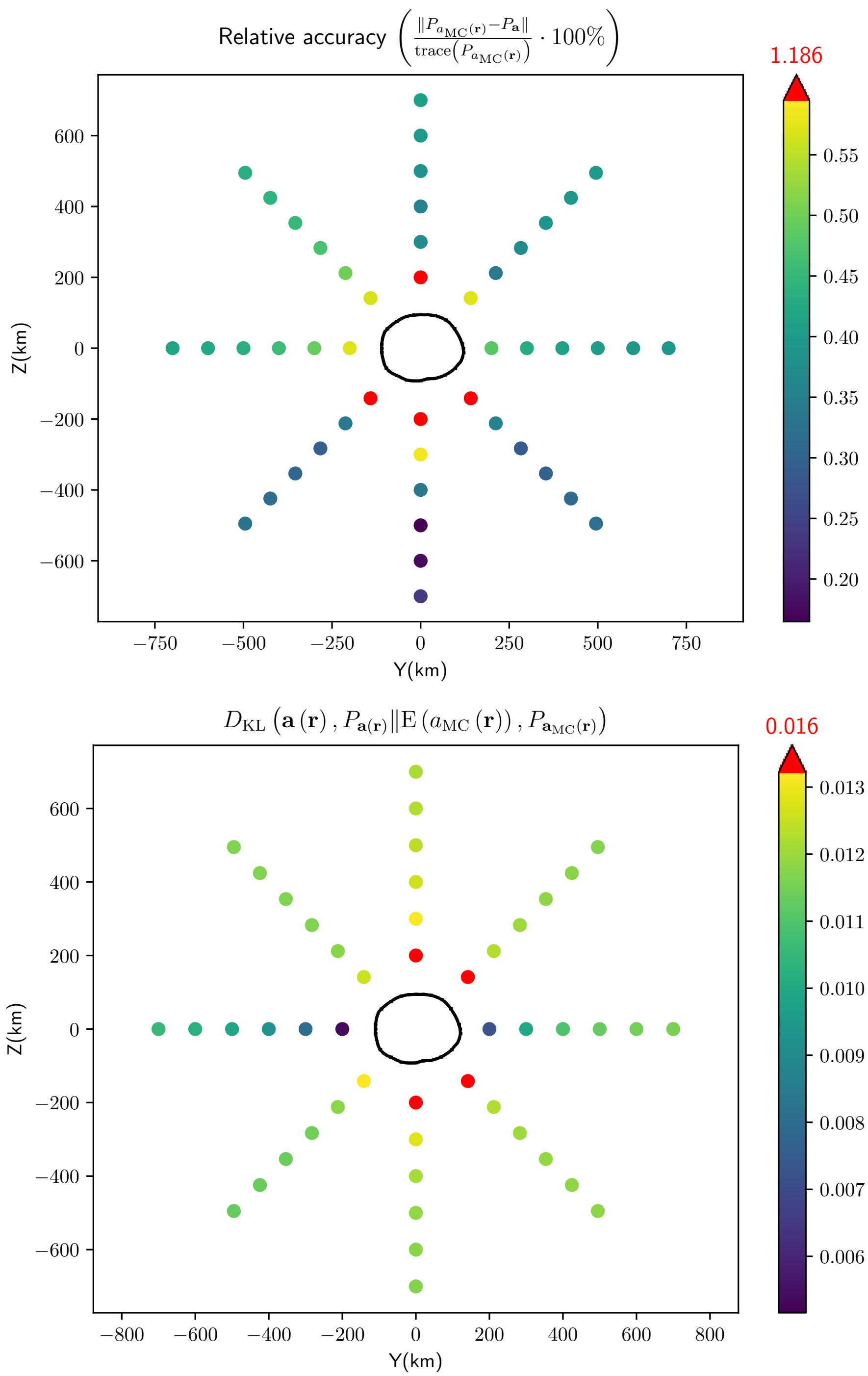


Figure 7.15: Relative accuracy of analytical gravity uncertainty prediction

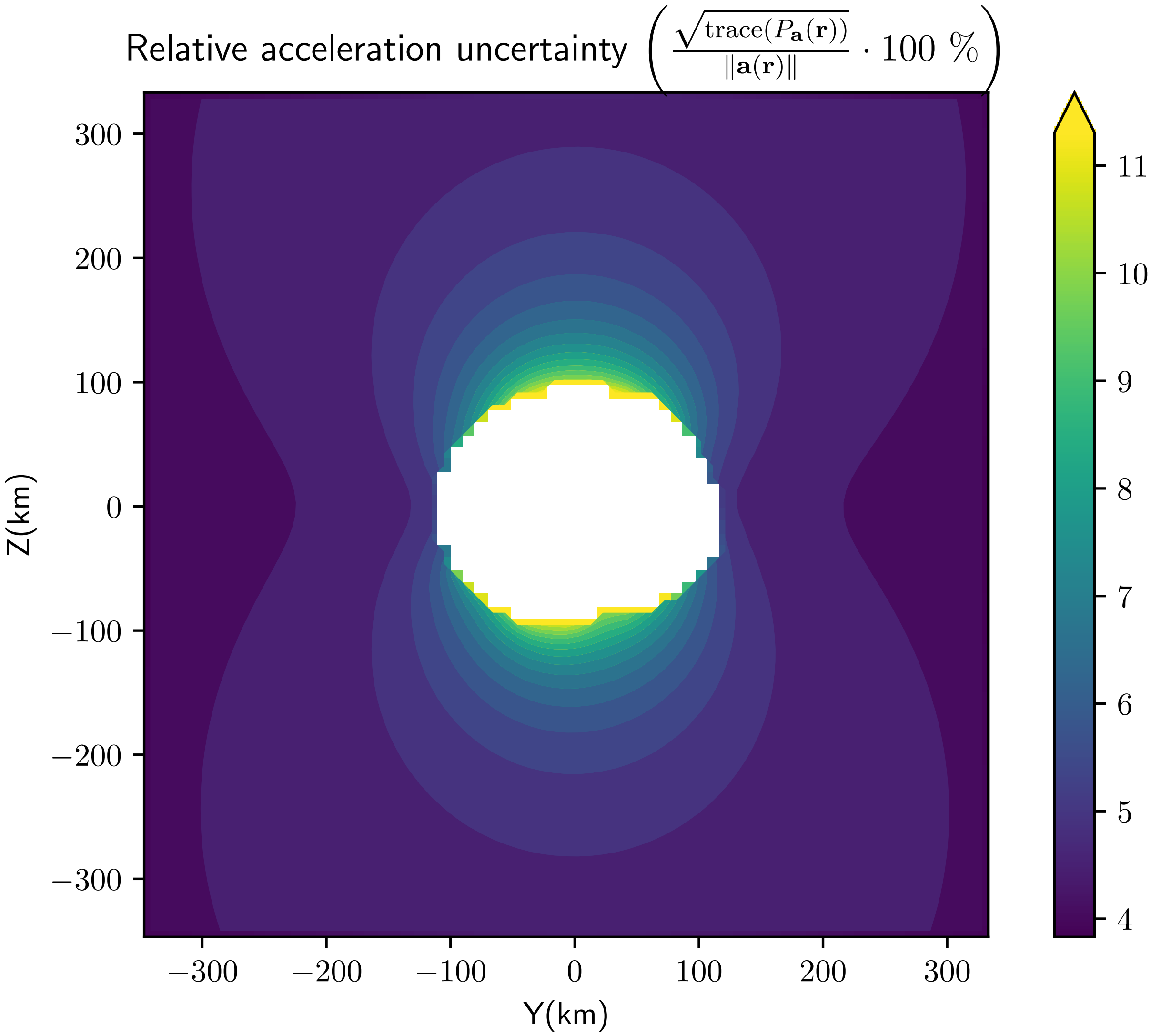


Figure 7.16: Prediction of gravity acceleration uncertainty around (16) Psyche

Table 7.8: Facet center coordinates and associated reference slopes

	Facet 0	Facet 1500
Coordinates (km)	$(4.94 \quad 2.853 \quad 94.14)^T$	$(125.544 \quad -4.605 \quad -3.5761)^T$
Slope (deg)	0.382	18.087

body-fixed acceleration. The statistics of the slopes distribution at two sample facets (facet 0 and facet 1500) were computed and compared to the analytical predictions arising from the linearized acceleration gravity model. Table 7.8 provides the Cartesian coordinates of the facet centers as well as the gravitational slopes evaluated at the facet centers of the reference shape. The resulting statistics in the slopes distribution and prediction errors are shown on Table 7.9. It is clear from this table that the uncertainty in the slopes is not well captured in all cases. Facet 0 lies at the center of one of the two uncertainty regions, and is thus highly perturbed by the resulting terrain motion. On the contrary, facet 1500 is much closer to the equator, such that the local shape uncertainty at this facet's center is limited. The much better performance of the slope uncertainty prediction at Facet 1500 thus simply stems from the fact that the direction of the surface at this facet varies much less than at Facet 0.

In conclusion, Figure 7.19 depicts a side-by-side comparison of the gravitational slopes along with their corresponding uncertainty measure over the surface of Psyche. Although polar region are not necessarily well captured by the proposed linearized uncertainty model, these maps are

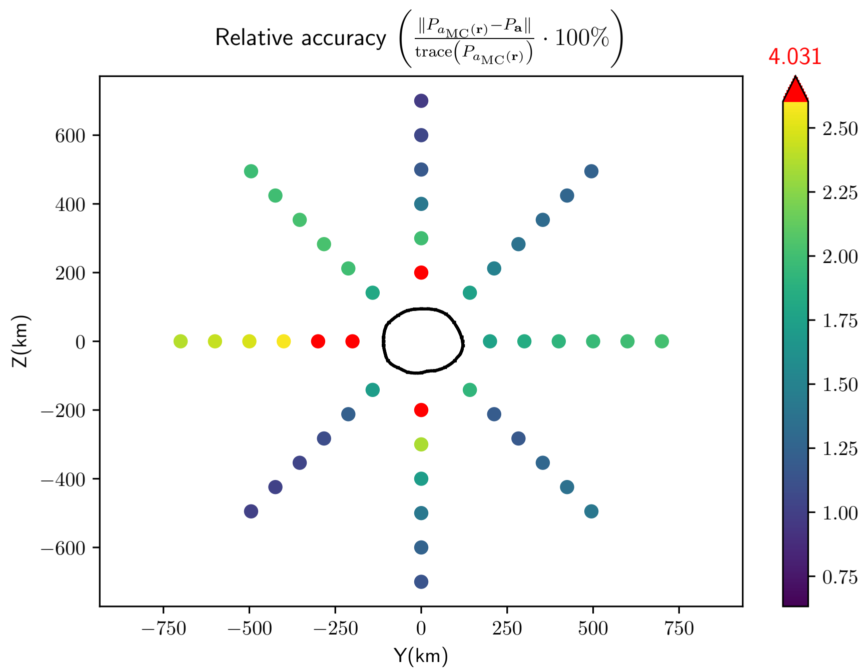
Table 7.9: Comparison of the standard deviation of the slopes distribution from the Monte-Carlo samples and those obtained from the analytical uncertainty model

Standard deviation	Facet 0	Facet 1500
Monte-Carlo (deg)	4.5	1.882
Predicted (deg)	7.04	1.8557
Relative error (%)	-59.19	3.095

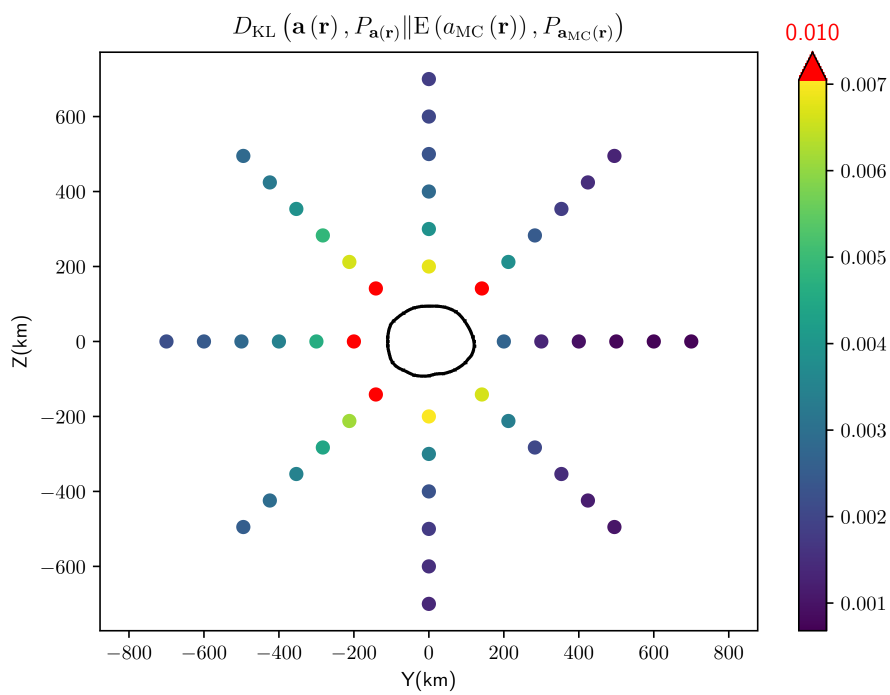
nonetheless providing valuable insight into the surface geophysical environmental of imperfectly known small bodies.

7.6 Performance

Some insight into the performance of the linearized gravity uncertainty model can be inferred from the respective runtimes of the uncertainty grid evaluation and that of the Monte-Carlo sampling of the selected points in the gravity field validation. For instance, the simulation that was run to produce Figure 7.15 featured an X—Y grid comprised of 4761 points. Evaluating the analytical prediction of the acceleration covariance over each of these points took a total of 644.3 seconds on the designated simulation computer. On the other hand, sampling the acceleration over the 5000 different shape models at each of the 48 selected points took 64.2 seconds. Extrapolating the Monte-Carlo runtime to the entire grid yields a rough estimate of 6363.3 seconds. For this given number of Monte-Carlo samples, the analytical gravity uncertainty quantification model is thus roughly 10 times faster than the Monte-Carlo. The implementation of this code is available as part of the Small Body Geophysical Analysis ToolBox, (SBGAT), in addition to the methods presented in Chapter 6. SBGAT is available on MacOS and Linux systems, and can be retrieved from GitHub at `https://github.com/bbercovici/SBGAT` or directly installed through Homebrew on Macs.



(a) L2-norm relative error



(b) Kullback–Leibler divergence

Figure 7.17: Prediction error in acceleration uncertainty around (16) Psyche when enforcing constant mass

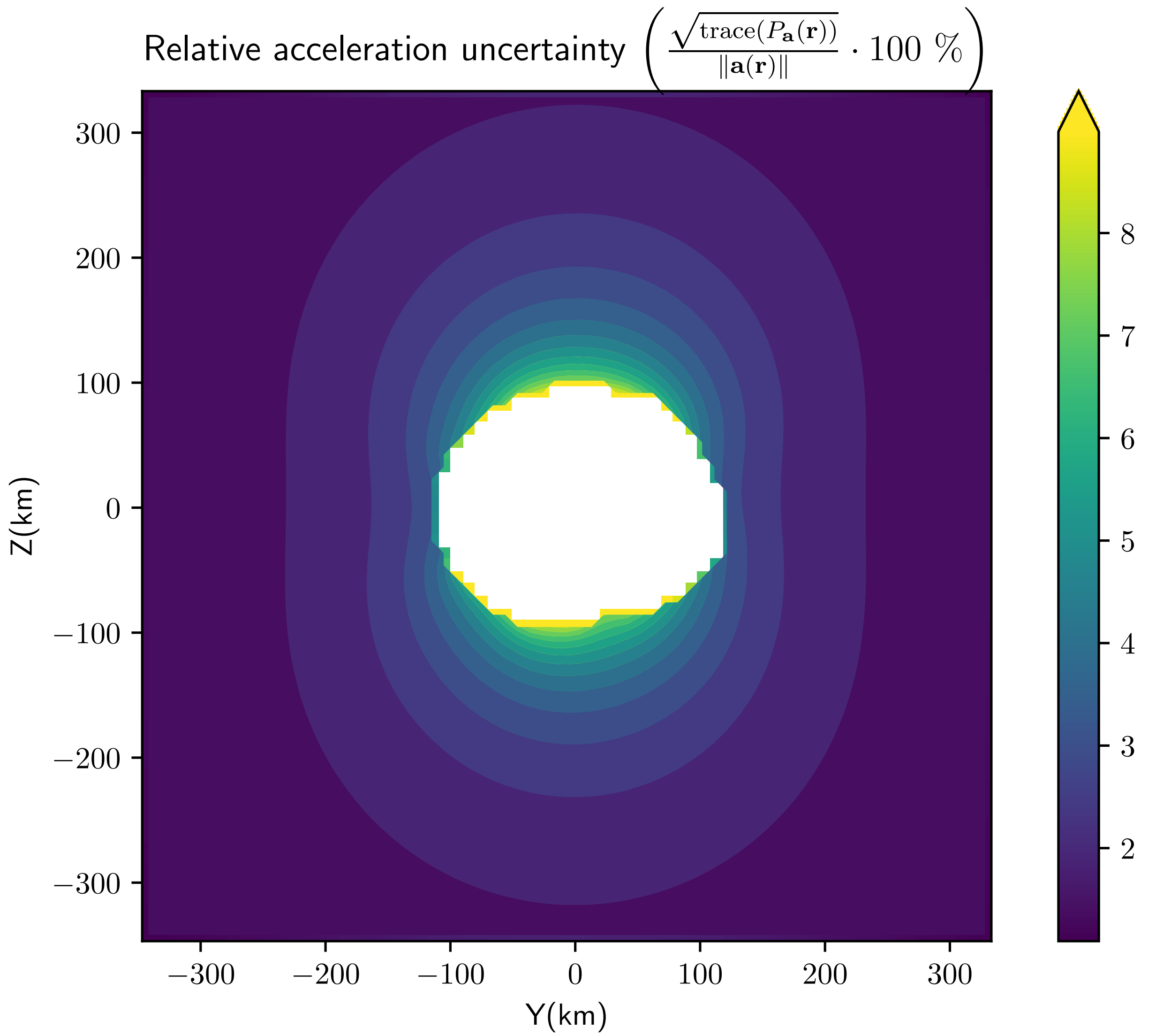
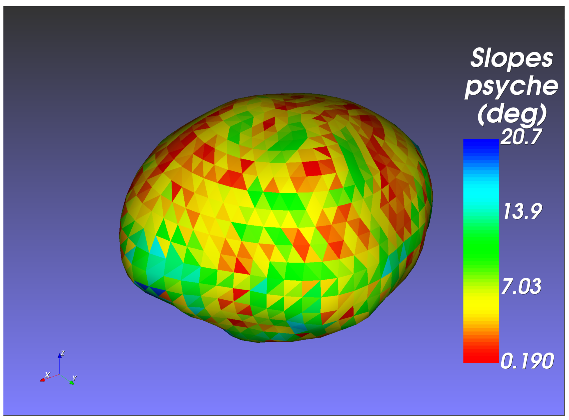
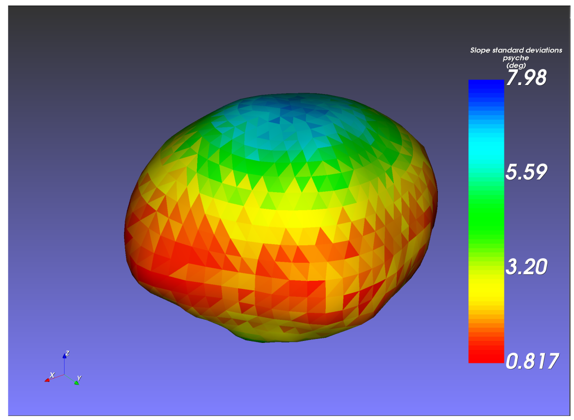


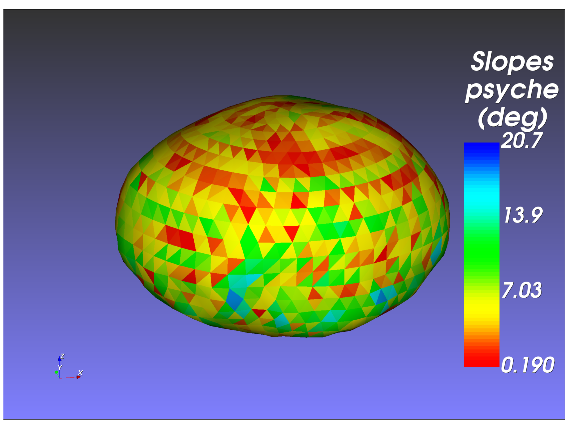
Figure 7.18: Prediction in the acceleration uncertainty around (16) Psyche when enforcing constant mass



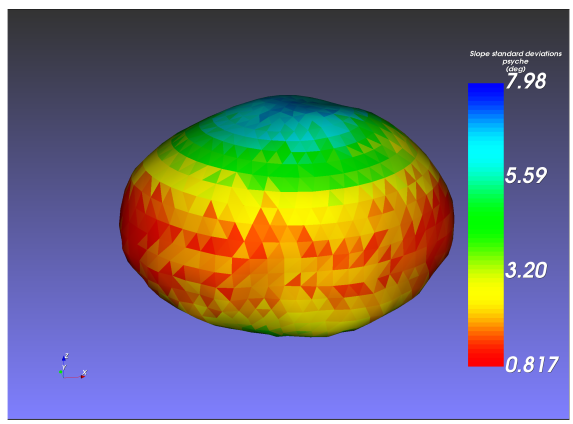
(a) Slopes (deg), first view



(b) Slope uncertainties (deg), first view



(c) Slopes (deg), second view



(d) Slope uncertainties (deg), second view

Figure 7.19: Side-by-side comparison of gravitational slopes (left) and associated analytical one-sigma standard deviations (right) over Psyche

Chapter 8

Model-based navigation

8.1 Initial orbit determination

This section demonstrates how a vector of rigid transform invariant $\boldsymbol{\epsilon}$ constraining the successive instrument positions relative to the center-of-mass of the object of interest can be constructed. Figure 8.2 depicts the spacecraft and its Lidar sensor in the proximity of a small body as the spacecraft flies along its trajectory. Let two paired point clouds $\{\mathbf{D}_i\}_{i=1}^{p}$ and $\{\mathbf{S}_i\}_{i=1}^{p}$ be collected at successive times t_{k-1} and t_k and registered by means of ICP in the instrument frame at t_{k-1}.

Assuming a perfect ICP registration, every source/destination pair indexed by i yields approximate overlap of the two paired points:

$$M_k \ {}^{\mathcal{L}_k}\left(\mathbf{L}_k\mathbf{S}_i\right) + \mathbf{X}_k \simeq \ {}^{\mathcal{L}_{k-1}}\left(\mathbf{L}_{k-1}\mathbf{D}_i\right) \tag{8.1}$$

where the instrument frames at the successive times $\mathcal{L}_{k-1}$ and $\mathcal{L}_k$ have been explicitly specified, along with the instrument location at successive times $\mathbf{L}_{k-1}$ and $\mathbf{L}_k$. Let $\mathbf{F}$ be a fictitious body-fixed surface feature on the small body, fixed in its body frame $\mathcal{B}$. Contrary to the point pairs that never associate exactly overlapping features, $\mathbf{F}$ exactly satisfies

$$M_k \ {}^{\mathcal{L}_k}\left(\mathbf{L}_k\mathbf{F}_k\right) + \mathbf{X}_k = \ {}^{\mathcal{L}_{k-1}}\left(\mathbf{L}_{k-1}\mathbf{F}_{k-1}\right) \tag{8.2}$$

Looking at the frames showing on both sides of the equation, it is clear that M_k' can be written as

$$M_k = [\mathcal{L}_{k-1}\mathcal{L}_k] = [\mathcal{L}_{k-1}\mathcal{B}_{k-1}][\mathcal{B}_k\mathcal{L}_k] = [\mathcal{L}\mathcal{B}](t_{k-1})[\mathcal{B}\mathcal{L}](t_k) \tag{8.3}$$

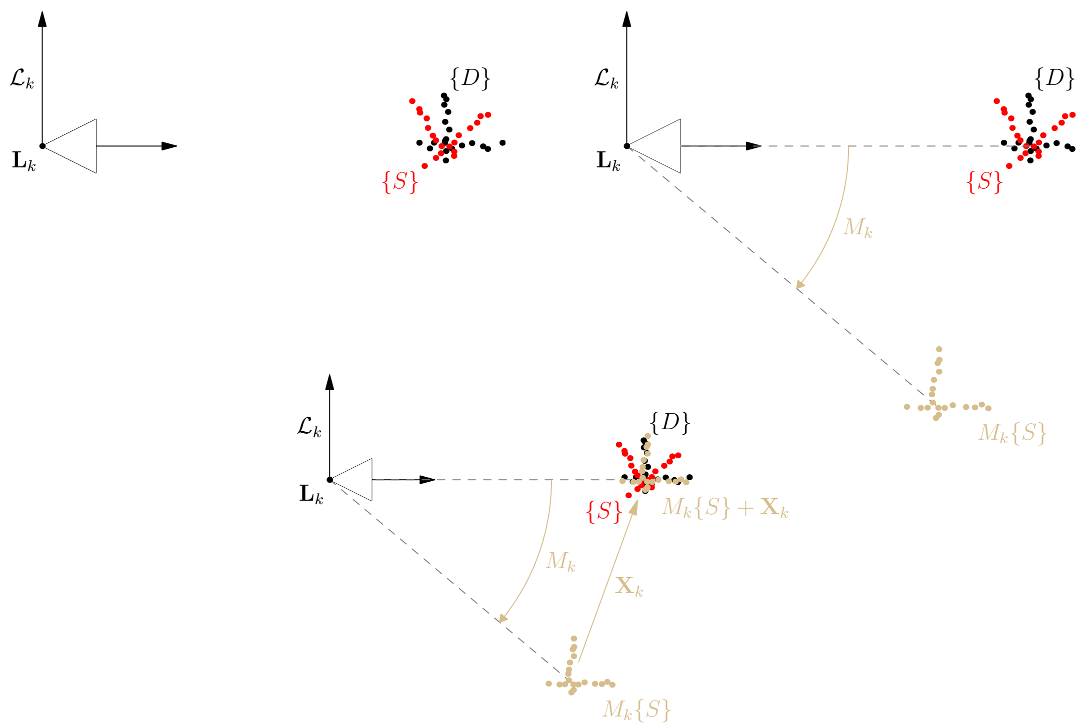


Figure 8.1: Notional depiction of the ICP process. Top left: the source and destination point clouds are acquired in the instrument frame at two successive times. Top right: the rotational component of the rigid transform rotates the source point cloud about the instrument's origin. Bottom: the translational component of the rigid transform finally aligns the areas of overlap in the two point clouds

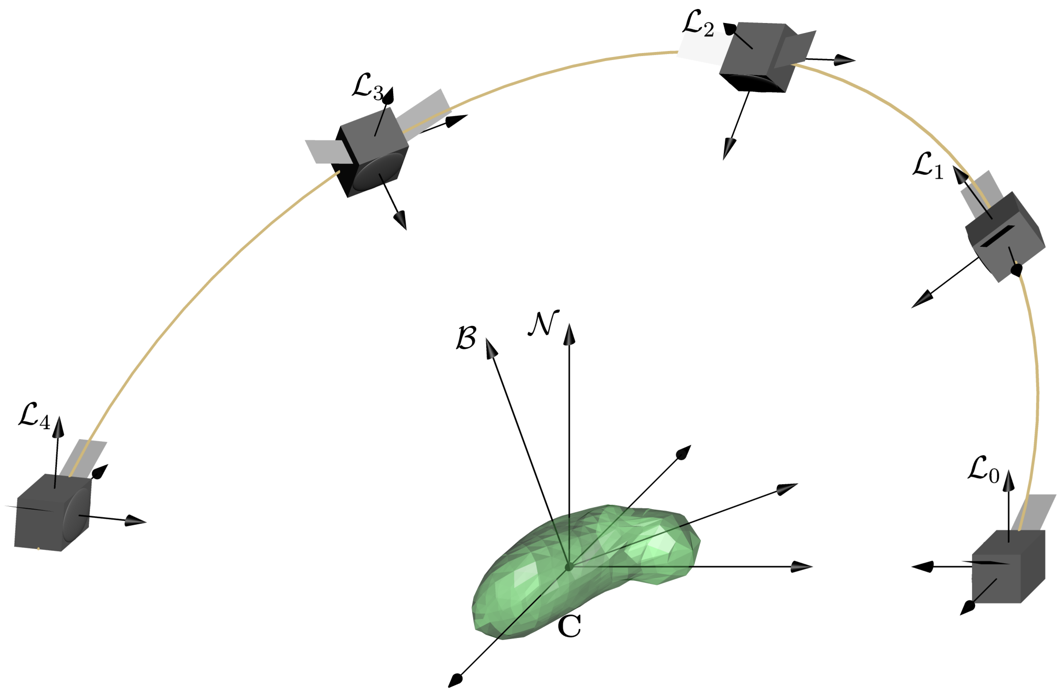


Figure 8.2: Notional depiction of the spacecraft on its trajectory, the imaged small body whose barycenter is located at **C**, the inertial and body-fixed frames $\mathcal{N}$ and $\mathcal{B}$ along with the successive instrument frames $\mathcal{L}_k$

8.1.1 Incorporating spacecraft attitude knowledge

Assuming that the spacecraft attitude $[\mathcal{LN}]$ is known at every measurement time, $M_k\ ^{\mathcal{L}_k}(\mathbf{L}_k\mathbf{F}_k)$ can be further expanded into

$$M_k\ ^{\mathcal{L}_k}(\mathbf{L}_k\mathbf{F}_k) = [\mathcal{LN}](t_{k-1})[\mathcal{NB}](t_{k-1})[\mathcal{BN}](t_k)[\mathcal{NL}](t_k)^{\mathcal{L}_k}(\mathbf{L}_k\mathbf{F}_k) \tag{8.4}$$

$$= [\mathcal{LN}](t_{k-1})[\mathcal{NB}](t_{k-1})[\mathcal{BN}](t_k)^{\mathcal{N}}(\mathbf{L}_k\mathbf{F}_k) \tag{8.5}$$

Introducing the sequential rigid transform of parametrization $\mathbf{I}'_k = \begin{pmatrix}\mathbf{X}'^T_k & \boldsymbol{\sigma}'^T_k\end{pmatrix}^T$, such that

$$M'_k\left(\boldsymbol{\sigma}'_k\right) \equiv [\mathcal{NL}](t_{k-1})M_k[\mathcal{LN}](t_k) \tag{8.6}$$

$$= [\mathcal{NB}](t_{k-1})[\mathcal{BN}](t_k) \tag{8.7}$$

$$\mathbf{X}'_k \equiv [\mathcal{NL}](t_{k-1})\mathbf{X}_k \tag{8.8}$$

the equation satisfied by $\mathbf{F}$ becomes

$$M'_k\ ^{\mathcal{N}}(\mathbf{L}_k\mathbf{F}_k) + \mathbf{X}'_k = \ ^{\mathcal{N}}(\mathbf{L}_{k-1}\mathbf{F}_{k-1}) \tag{8.9}$$

8.1.2 Leveraging the barycentric rotation

Introducing the small body's center-of-mass $\mathbf{C}$,

$$M'_k\ ^{\mathcal{N}}(\mathbf{L}_k\mathbf{C} + \mathbf{C}\mathbf{F}_k) + \mathbf{X}'_k = \ ^{\mathcal{N}}(\mathbf{L}_{k-1}\mathbf{C}) + \ ^{\mathcal{N}}(\mathbf{C}\mathbf{F}_{k-1}) \tag{8.10}$$

Note that $\mathbf{C}$ is the origin of frames $\mathcal{N}$ and $\mathcal{B}$. The above equation becomes

$$M'_k\ ^{\mathcal{N}}(\mathbf{L}_k\mathbf{C}) + \mathbf{X}'_k = \ ^{\mathcal{N}}(\mathbf{L}_{k-1}\mathbf{C}) + ^{\mathcal{N}}(\mathbf{C}\mathbf{F}_{k-1}) - M'_k\ ^{\mathcal{N}}(\mathbf{C}\mathbf{F}_k) \tag{8.11}$$

Since $\mathbf{F}$ has undergone pure rotational motion about $\mathbf{C}$ between t_{k-1} and t_k,

$$^{\mathcal{N}}(\mathbf{C}\mathbf{F}_{k-1}) - M'_k\ ^{\mathcal{N}}(\mathbf{C}\mathbf{F}_k) = \mathbf{0} \tag{8.12}$$

As a result, the unknown spacecraft location in the $\mathcal{N}$ frame at successive observation times satisfies

$$M_k' \ {}^{\mathcal{N}}(\mathbf{L}_k\mathbf{C}) + \mathbf{X}_k' - \ {}^{\mathcal{N}}(\mathbf{L}_{k-1}\mathbf{C}) = \mathbf{0} \tag{8.13}$$

Introducing the spacecraft position $\mathbf{r}_k \equiv \ {}^{\mathcal{N}}(\mathbf{CL}_k)$, it finally becomes

$$M_k'\mathbf{r}_k + \mathbf{X}_k' - \mathbf{r}_{k-1} = \mathbf{0} \tag{8.14}$$

Assuming that N such measurements are available, the determination of the spacecraft position relative to the center-of-mass at the epoch time can be reformulated into solving the problem

$$\min_{\mathbf{r}_0, \dot{\mathbf{r}}_0, \boldsymbol{\Theta}} \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \tag{8.15}$$

with

$$\boldsymbol{\epsilon} = \begin{pmatrix} \mathbf{r}_0 - M_1' \ \mathbf{r}_1 + \mathbf{X}_1' \\ \mathbf{r}_1 - M_2' \ \mathbf{r}_2 + \mathbf{X}_2' \\ \vdots \\ \mathbf{r}_{N-1} - M_N' \ \mathbf{r}_N + \mathbf{X}_N' \end{pmatrix} \tag{8.16}$$

At this stage, the trajectory takes no particular form as the observation residuals are only a function of the position.

8.1.3 Extracting $(M_k', \mathbf{X}_k')$ from $\left(\tilde{M}_k, \tilde{\mathbf{X}}_k\right)$

For practical purposes, it is desireable to perform registration not from the $\mathcal{L}_k$ frame to $\mathcal{L}_{k-1}$, but to a reference stitching frame $\mathcal{L}_0$. In this case, the registration equation reads

$$\tilde{M}_k \ {}^{\mathcal{L}_k}(\mathbf{L}_k\mathbf{F}_k) + \tilde{\mathbf{X}}_k = \tilde{M}_{k-1} \ {}^{\mathcal{L}_{k-1}}(\mathbf{L}_{k-1}\mathbf{F}_{k-1}) + \tilde{\mathbf{X}}_{k-1} \tag{8.17}$$

where $\left(\tilde{M}_k, \tilde{\mathbf{X}}_k\right)$ is the absolute rigid transform mapping from $\mathcal{L}_k$ to the stiching frame $\mathcal{L}_0$:

$$\tilde{M}_k = [\mathcal{LN}](t_0)[\mathcal{NB}](t_0)[\mathcal{BN}](t_k)[\mathcal{NL}](t_k) \tag{8.18}$$

$(M'_k, \mathbf{X}'_k)$ can thus be extracted from

$$\mathbf{X}'_k = [\mathcal{N}\mathcal{L}](t_{k-1})\tilde{M}^T_{k-1}\left(\tilde{\mathbf{X}}_k - \tilde{\mathbf{X}}_{k-1}\right) \tag{8.19}$$

$$M'_k = [\mathcal{N}\mathcal{L}](t_{k-1})\tilde{M}^T_{k-1}\tilde{M}_k[\mathcal{L}\mathcal{N}](t_k) \tag{8.20}$$

One needs to derive the first variation in $\mathbf{I}'_k$ given those in $\tilde{\mathbf{I}}_k, \tilde{\mathbf{I}}_{k-1}$. A first-order Taylor expansion of Equation (8.19) about the a-priori rigid transform yields

$$\bar{\mathbf{X}}'_k + \delta\mathbf{X}'_k \simeq [\mathcal{N}\mathcal{L}](t_{k-1})\delta\tilde{M}^T_{k-1}\bar{\tilde{M}}^T_{k-1}\left(\bar{\tilde{\mathbf{X}}}_k + \delta\tilde{\mathbf{X}}_k - \bar{\tilde{\mathbf{X}}}_{k-1} - \delta\tilde{\mathbf{X}}_{k-1}\right) \tag{8.21}$$

Introducing the incremental rotation matrices and associated MRP parametrizations, which under a small angle assumption reduces to [68]

$$\delta\tilde{M}_k = I_3 - 4[\widetilde{\tilde{\boldsymbol{\sigma}}_k}] \tag{8.22}$$

the first-order Taylor expansion of Equation (8.19) becomes

$$\bar{\mathbf{X}}'_k + \delta\mathbf{X}'_k \simeq [\mathcal{N}\mathcal{L}](t_{k-1})\delta\tilde{M}^T_{k-1}\bar{\tilde{M}}^T_{k-1}\left(\bar{\tilde{\mathbf{X}}}_k + \delta\tilde{\mathbf{X}}_k - \bar{\tilde{\mathbf{X}}}_{k-1} - \delta\tilde{\mathbf{X}}_{k-1}\right) \tag{8.23}$$

$$\simeq [\mathcal{N}\mathcal{L}](t_{k-1})\left(I_3 + 4[\widetilde{\tilde{\boldsymbol{\sigma}}_{k-1}}]\right)\bar{\tilde{M}}^T_{k-1}\left(\bar{\tilde{\mathbf{X}}}_k + \delta\tilde{\mathbf{X}}_k - \bar{\tilde{\mathbf{X}}}_{k-1} - \delta\tilde{\mathbf{X}}_{k-1}\right) \tag{8.24}$$

Expanding the above equation and retaining only first-order terms,

$$\delta\mathbf{X}'_k \simeq [\mathcal{N}\mathcal{L}](t_{k-1})\left(I_3 + 4[\widetilde{\tilde{\boldsymbol{\sigma}}_{k-1}}]\right)\bar{\tilde{M}}^T_{k-1}\left(\bar{\tilde{\mathbf{X}}}_k + \delta\tilde{\mathbf{X}}_k - \bar{\tilde{\mathbf{X}}}_{k-1} - \delta\tilde{\mathbf{X}}_{k-1}\right) \tag{8.25}$$

$$= [\mathcal{N}\mathcal{L}](t_{k-1})\left[\bar{\tilde{M}}^T_{k-1}\left(\delta\tilde{\mathbf{X}}_k - \delta\tilde{\mathbf{X}}_{k-1}\right) + 4[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_{k-1}}]\bar{\tilde{M}}^T_{k-1}\left(\bar{\tilde{\mathbf{X}}}_k - \bar{\tilde{\mathbf{X}}}_{k-1}\right)\right] \tag{8.26}$$

Introducing the shortcut

$$\bar{\mathbf{a}}_k \equiv \bar{\tilde{M}}^T_{k-1}\left(\bar{\tilde{\mathbf{X}}}_k - \bar{\tilde{\mathbf{X}}}_{k-1}\right) \tag{8.27}$$

$$\bar{U}_k \equiv -4[\widetilde{\bar{\mathbf{a}}_k}] \tag{8.28}$$

such that

$$4[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_{k-1}}]\bar{\tilde{M}}^T_{k-1}\left(\bar{\tilde{\mathbf{X}}}_k - \bar{\tilde{\mathbf{X}}}_{k-1}\right) = \bar{U}_k\delta\tilde{\boldsymbol{\sigma}}_{k-1} \tag{8.29}$$

The first variation in $\mathbf{X}'_k$ reads

$$\delta\mathbf{X}'_k = [\mathcal{N}\mathcal{L}](t_{k-1})\left[\bar{\tilde{M}}^T_{k-1}\left(\delta\tilde{\mathbf{X}}_k - \delta\tilde{\mathbf{X}}_{k-1}\right) + \bar{U}_k\delta\tilde{\boldsymbol{\sigma}}_{k-1}\right] \tag{8.30}$$

The same process must be repeated with $\delta M'_k$ and its associated parametrization $\delta\boldsymbol{\sigma}'_k$.

A first-order Taylor expansion of Equation (8.20) about the a-priori rigid transform yields

$$\bar{M}'_k \delta M'_k = [\mathcal{NL}](t_{k-1})\delta\tilde{M}^T_{k-1}\bar{\tilde{M}}^T_{k-1}\bar{\tilde{M}}_k\delta\tilde{M}_k[\mathcal{LN}](t_k) \tag{8.31}$$

Introducing a similar MRP parametrization of the incremental error matrices as before,

$$\bar{M}'_k\left(I_3 - 4[\widetilde{\delta\boldsymbol{\sigma}'_k}]\right) = [\mathcal{NL}](t_{k-1})\left(I_3 + 4[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_{k-1}}]\right)\bar{\tilde{M}}^T_{k-1}\bar{\tilde{M}}_k\left(I_3 - 4[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_k}]\right)[\mathcal{LN}](t_k) \tag{8.32}$$

Retaining only the first order terms of interest,

$$\begin{aligned} -4\bar{M}'_k[\widetilde{\delta\boldsymbol{\sigma}'_k}] &= 4[\mathcal{NL}](t_{k-1})[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_{k-1}}]\bar{\tilde{M}}^T_{k-1}\bar{\tilde{M}}_k[\mathcal{LN}](t_k) \\ &\quad - 4[\mathcal{NL}](t_{k-1})\bar{\tilde{M}}^T_{k-1}\bar{\tilde{M}}_k[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_k}][\mathcal{LN}](t_k) \end{aligned} \tag{8.33}$$

or

$$[\widetilde{\delta\boldsymbol{\sigma}'_k}] = \bar{M}'^T_k[\mathcal{NL}](t_{k-1})\left(\bar{\tilde{M}}^T_{k-1}\bar{\tilde{M}}_k[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_k}] - [\widetilde{\delta\tilde{\boldsymbol{\sigma}}_{k-1}}]\bar{\tilde{M}}^T_{k-1}\bar{\tilde{M}}_k\right)[\mathcal{LN}](t_k) \tag{8.34}$$

which can be rewritten as

$$[\widetilde{\delta\boldsymbol{\sigma}'_k}] = \bar{M}'^T_k[\mathcal{NL}](t_{k-1})\bar{A}_k[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_k}][\mathcal{LN}](t_k) - \bar{M}'^T_k[\mathcal{NL}](t_{k-1})[\widetilde{\delta\tilde{\boldsymbol{\sigma}}_{k-1}}]\bar{A}_k[\mathcal{LN}](t_k) \tag{8.35}$$

with

$$\bar{A}_k \equiv \bar{\tilde{M}}^T_{k-1}\bar{\tilde{M}}_k \tag{8.36}$$

$$\tag{8.37}$$

The underlying MRP can now be accessed. The cross-product application $\mathbf{x} \mapsto \mathbf{z} \times \mathbf{x} = [\widetilde{\mathbf{z}}]\mathbf{x}$ being parametrized by $\mathbf{z} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix}$,

$$[\widetilde{\mathbf{z}}] = \begin{bmatrix} 0 & -z_2 & z_1 \\ z_2 & 0 & -z_0 \\ -z_1 & z_0 & 0 \end{bmatrix} \tag{8.38}$$

$$z_0 = \hat{e}_2^T \widetilde{[\mathbf{z}]} \hat{e}_1 \tag{8.39}$$

$$z_1 = \hat{e}_0^T \widetilde{[\mathbf{z}]} \hat{e}_2 \tag{8.40}$$

$$z_2 = \hat{e}_1^T \widetilde{[\mathbf{z}]} \hat{e}_0 \tag{8.41}$$

with

$$\hat{e}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \ \hat{e}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \hat{e}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{8.42}$$

Letting $\widetilde{[\mathbf{z}]} \equiv \widetilde{[\delta \boldsymbol{\sigma}'_k]}$,

$$\delta \boldsymbol{\sigma}'_k = \begin{pmatrix} \hat{e}_2^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \bar{A}_k \widetilde{[\delta \tilde{\boldsymbol{\sigma}}_k]} [\mathcal{L}\mathcal{N}](t_k) \hat{e}_1 - \hat{e}_2^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \widetilde{[\delta \tilde{\boldsymbol{\sigma}}_{k-1}]} \bar{A}_k [\mathcal{L}\mathcal{N}](t_k) \hat{e}_1 \\ \hat{e}_0^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \bar{A}_k \widetilde{[\delta \tilde{\boldsymbol{\sigma}}_k]} [\mathcal{L}\mathcal{N}](t_k) \hat{e}_2 - \hat{e}_0^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \widetilde{[\delta \tilde{\boldsymbol{\sigma}}_{k-1}]} \bar{A}_k [\mathcal{L}\mathcal{N}](t_k) \hat{e}_2 \\ \hat{e}_1^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \bar{A}_k \widetilde{[\delta \tilde{\boldsymbol{\sigma}}_k]} [\mathcal{L}\mathcal{N}](t_k) \hat{e}_0 - \hat{e}_1^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \widetilde{[\delta \tilde{\boldsymbol{\sigma}}_{k-1}]} \bar{A}_k [\mathcal{L}\mathcal{N}](t_k) \hat{e}_0 \end{pmatrix} \tag{8.43}$$

Finally,

$$\delta \boldsymbol{\sigma}'_k = \left[\frac{\partial \boldsymbol{\sigma}'_k}{\partial \mathbf{Z}_k} \right] \delta \mathbf{Z}_k \tag{8.44}$$

with

$$\mathbf{Z}_k \equiv \begin{pmatrix} \tilde{\boldsymbol{\sigma}}_{k-1} \\ \tilde{\boldsymbol{\sigma}}_k \end{pmatrix} \tag{8.45}$$

and

$$\left[\frac{\partial \boldsymbol{\sigma}'_k}{\partial \mathbf{Z}_k} \right] \equiv \begin{bmatrix} \hat{e}_2^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \widetilde{[\bar{\mathbf{C}}_k^1]} & -\hat{e}_2^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \bar{A}_k \widetilde{[\bar{\mathbf{B}}_k^1]} \\ \hat{e}_0^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \widetilde{[\bar{\mathbf{C}}_k^2]} & -\hat{e}_0^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \bar{A}_k \widetilde{[\bar{\mathbf{B}}_k^2]} \\ \hat{e}_1^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \widetilde{[\bar{\mathbf{C}}_k^0]} & -\hat{e}_1^T \bar{M}_k'^T [\mathcal{N}\mathcal{L}](t_{k-1}) \bar{A}_k \widetilde{[\bar{\mathbf{B}}_k^0]} \end{bmatrix} \tag{8.46}$$

$$\mathbf{B}_k^i \equiv [\mathcal{L}\mathcal{N}](t_k) \hat{e}_i \tag{8.47}$$

$$\bar{\mathbf{C}}_k^i \equiv \bar{A}_k [\mathcal{L}\mathcal{N}](t_k) \hat{e}_i \tag{8.48}$$

So

$$\delta \mathbf{I}'_k = \left[\frac{\partial \mathbf{I}'_k}{\partial \tilde{\mathbf{V}}_k} \right] \delta \tilde{\mathbf{V}}_k \tag{8.49}$$

where $\tilde{\mathbf{V}}_k$ holds the states of the two consecutive rigid transforms with respect to the stitching frame

$$\tilde{\mathbf{V}}_k \equiv \begin{pmatrix} \tilde{\mathbf{I}}_{k-1} \\ \tilde{\mathbf{I}}_k \end{pmatrix} \tag{8.50}$$

and

$$\left[\frac{\partial \mathbf{I}'_k}{\partial \tilde{\mathbf{V}}_k}\right] = \begin{bmatrix} -[\mathcal{NL}](t_{k-1})\tilde{\bar{M}}^T_{k-1} & [\mathcal{NL}](t_{k-1})\bar{U}_k & [\mathcal{NL}](t_{k-1})\tilde{\bar{M}}^T_{k-1} & \mathbf{0}_{33} \\ \mathbf{0}_{33} & \left[\frac{\partial \boldsymbol{\sigma}'_k}{\partial \mathbf{Z}_k}\right]_{(:,0:2)} & \mathbf{0}_{33} & \left[\frac{\partial \boldsymbol{\sigma}'_k}{\partial \mathbf{Z}_k}\right]_{(:,3:5)} \end{bmatrix} \tag{8.51}$$

In conclusion, the first variation in the parametrization of the sequential rigid transforms can be related to that of the absolute rigid transforms through

$$\delta \mathbf{I}' \equiv \begin{pmatrix} \delta \mathbf{I}'_1 \\ \delta \mathbf{I}'_2 \\ \vdots \\ \delta \mathbf{I}'_{N-1} \end{pmatrix} \tag{8.52}$$

$$= \begin{bmatrix} \frac{\partial \mathbf{I}'_1}{\partial \tilde{\mathbf{V}}_0} & \frac{\partial \mathbf{I}'_1}{\partial \tilde{\mathbf{V}}_1} & \cdots & \frac{\partial \mathbf{I}'_1}{\partial \tilde{\mathbf{V}}_{N-1}} \\ \frac{\partial \mathbf{I}'_2}{\partial \tilde{\mathbf{V}}_0} & \frac{\partial \mathbf{I}'_2}{\partial \tilde{\mathbf{V}}_1} & \cdots & \frac{\partial \mathbf{I}'_2}{\partial \tilde{\mathbf{V}}_{N-1}} \\ \vdots & \vdots & & \\ \frac{\partial \mathbf{I}'_{N-1}}{\partial \tilde{\mathbf{V}}_0} & \frac{\partial \mathbf{I}'_{N-1}}{\partial \tilde{\mathbf{V}}_1} & \cdots & \frac{\partial \mathbf{I}'_{N-1}}{\partial \tilde{\mathbf{V}}_{N-1}} \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{\mathbf{V}}_0}{\partial \tilde{\mathbf{I}}_0} & \frac{\partial \tilde{\mathbf{V}}_0}{\partial \tilde{\mathbf{I}}_1} & \cdots & \frac{\partial \tilde{\mathbf{V}}_0}{\partial \tilde{\mathbf{I}}_{N-1}} \\ \frac{\partial \tilde{\mathbf{V}}_1}{\partial \tilde{\mathbf{I}}_0} & \frac{\partial \tilde{\mathbf{V}}_1}{\partial \tilde{\mathbf{I}}_1} & \cdots & \frac{\partial \tilde{\mathbf{V}}_1}{\partial \tilde{\mathbf{I}}_{N-1}} \\ \vdots & \vdots & & \\ \frac{\partial \tilde{\mathbf{V}}_N}{\partial \tilde{\mathbf{I}}_0} & \frac{\partial \tilde{\mathbf{V}}_{N-1}}{\partial \tilde{\mathbf{I}}_1} & \cdots & \frac{\partial \tilde{\mathbf{V}}_{N-1}}{\partial \tilde{\mathbf{I}}_{N-1}} \end{bmatrix} \begin{pmatrix} \delta \tilde{\mathbf{I}}_0 \\ \delta \tilde{\mathbf{I}}_1 \\ \delta \tilde{\mathbf{I}}_2 \\ \vdots \\ \delta \tilde{\mathbf{I}}_{N-1} \end{pmatrix} \tag{8.53}$$

$$= \left[\frac{\partial \mathbf{I}'}{\partial \tilde{\mathbf{V}}}\right] \left[\frac{\partial \tilde{\mathbf{V}}}{\partial \tilde{\mathbf{I}}}\right] \delta \tilde{\mathbf{I}} \tag{8.54}$$

A few observations can be made. First, if it is clear that the matrices $\left[\frac{\partial \mathbf{I}'}{\partial \tilde{\mathbf{V}}}\right]$ and $\left[\frac{\partial \tilde{\mathbf{V}}}{\partial \tilde{\mathbf{I}}}\right]$ will grow in size quickly as the number of rigid transforms increases, they are also very sparse. Second, it must be noted that $\delta \tilde{\mathbf{I}}_0 \equiv \mathbf{0}$ since the first absolute rigid transform is frozen. Third, $\tilde{\mathbf{V}}_0 \equiv \tilde{\mathbf{I}}_0$ while $\tilde{\mathbf{V}}_k \equiv \left(\tilde{\mathbf{I}}^T_{k-1} \quad \tilde{\mathbf{I}}^T_k\right)^T$ for $k > 0$.

8.1.4 Uncertainty propagation and filtering

A given rigid transform invariant reads

$$\boldsymbol{\epsilon}_{k+1} = \mathbf{r}_k - M'_{k+1}\ \mathbf{r}_{k+1} + \mathbf{X}'_{k+1} \equiv 0 \tag{8.55}$$

Linearizing about the a-priori states $(\bar{\mathbf{r}}_k, \bar{\mathbf{r}}_{k+1})$ and the mean rigid transform $\left(\bar{M}'_{k+1}, \bar{\mathbf{X}}'_{k+1}\right)$,

$$\boldsymbol{\epsilon}_{k+1} = \bar{\mathbf{r}}_k + \delta \mathbf{r}_k - \bar{M}'_{k+1} \delta M'_{k+1}\ \ (\bar{\mathbf{r}}_{k+1} + \mathbf{r}_{k+1}) + \bar{\mathbf{X}}'_{k+1} + \delta \mathbf{X}'_{k+1} \tag{8.56}$$

The DCM $\delta M'_{k+1}$ is an incremental rotation measure, parametrized as

$$\delta M'_{k+1} \equiv I_3 - 4[\widetilde{\delta \boldsymbol{\sigma}'_{k+1}}] \tag{8.57}$$

where $\delta\boldsymbol{\sigma}_{k+1}$ is the Modified Rodrigues Parameters set describing δM_{k+1}. Discarding any second-order term and higher in $\boldsymbol{\epsilon}_k$,

$$\begin{aligned}\boldsymbol{\epsilon}_{k+1} \simeq\ & \bar{\mathbf{r}}_k + \delta\mathbf{r}_k - \bar{M}'_{k+1}\left(I_3 - 4[\widetilde{\delta\boldsymbol{\sigma}'_{k+1}}]\right)(\bar{\mathbf{r}}_{k+1} + \mathbf{r}_{k+1}) \\ & + \bar{\mathbf{X}}'_{k+1} + \delta\mathbf{X}'_{k+1} \qquad (8.58) \\ \simeq\ & \bar{\mathbf{r}}_k + \delta\mathbf{r}_k - \bar{M}'_{k+1}\bar{\mathbf{r}}_{k+1} - \bar{M}'_{k+1}\delta\mathbf{r}_{k+1} \\ & - 4\bar{M}'_{k+1}[\widetilde{\mathbf{r}_{k+1}}]\delta\boldsymbol{\sigma}'_{k+1} + \bar{\mathbf{X}}'_{k+1} + \delta\mathbf{X}'_{k+1} \qquad (8.59) \\ & \qquad (8.60)\end{aligned}$$

The deviation in the state at the epoch time reads

$$\delta\mathbf{x}_0 \equiv \begin{pmatrix} {}^{\mathcal{N}}\delta\bar{\mathbf{r}}_0 \\ {}^{\mathcal{N}}\delta\dot{\bar{\mathbf{r}}}_0 \\ \delta\boldsymbol{\Theta} \end{pmatrix} \qquad (8.61)$$

and the state-transition matrix of the spacecraft state

$$\Phi_k \equiv \frac{\partial\mathbf{x}_k}{\partial\mathbf{x}_0} \qquad (8.62)$$

When $\boldsymbol{\Theta}$ reduces to the orbited body's standard gravitational parameter μ, Φ_k obeys the matrix differential equation

$$\dot{\Phi}_k = \mathcal{A}_{|\mathbf{x}=\bar{\mathbf{x}},t}\Phi_k \qquad (8.63)$$

where the Jacobian of the dynamics reads

$$\mathcal{A}_{|\mathbf{x}=\bar{\mathbf{x}},t} \equiv \begin{bmatrix} \frac{\partial\dot{\mathbf{r}}}{\partial\mathbf{r}} & \frac{\partial\dot{\mathbf{r}}}{\partial\dot{\mathbf{r}}} & \frac{\partial\dot{\mathbf{r}}}{\partial\mu} \\ \frac{\partial\ddot{\mathbf{r}}}{\partial\mathbf{r}} & \frac{\partial\ddot{\mathbf{r}}}{\partial\dot{\mathbf{r}}} & \frac{\partial\ddot{\mathbf{r}}}{\partial\mu} \\ \frac{\partial\dot{\mu}}{\partial\mathbf{r}} & \frac{\partial\dot{\mu}}{\partial\dot{\mathbf{r}}} & \frac{\partial\dot{\mu}}{\partial\mu} \end{bmatrix} \qquad (8.64)$$

Taking the model dynamics as purely keplerian (e.g $\ddot{\mathbf{r}} = -\frac{\mu}{(\mathbf{r}^T\mathbf{r})^{3/2}}\mathbf{r}$), it reduces to

$$\mathcal{A}_{|\mathbf{x}=\bar{\mathbf{x}},t} \equiv \begin{bmatrix} 0_{33} & I_{33} & \mathbf{0}_3 \\ \frac{\mu}{(\mathbf{r}^T\mathbf{r})^{3/2}}\left(3\frac{\mathbf{r}\mathbf{r}^T}{\mathbf{r}^T\mathbf{r}} - I_{33}\right) & 0_{33} & -\frac{\mathbf{r}}{(\mathbf{r}^T\mathbf{r})^{3/2}} \\ \mathbf{0}_3^T & \mathbf{0}_3^T & 0 \end{bmatrix} \qquad (8.65)$$

Define

$$\mathbf{y}_k \equiv \bar{\mathbf{r}}_k - \bar{M}'_{k+1}\bar{\mathbf{r}}_{k+1} + \bar{\mathbf{X}}'_{k+1} \tag{8.66}$$

$$H_k \equiv \bar{M}'_{k+1}\Phi_{k+1,(0:2,:)} - \Phi_{k,(0:2,:)} \tag{8.67}$$

$$\delta\mathbf{I}'_k \equiv \begin{pmatrix}\delta\mathbf{X}'^T_{k+1} & \delta\boldsymbol{\sigma}'^T_{k+1}\end{pmatrix}^T \tag{8.68}$$

$$J_k \equiv \begin{bmatrix}-I_3 & 4\bar{M}'_{k+1}\widetilde{[\bar{\mathbf{r}}_{k+1}]}\end{bmatrix} \tag{8.69}$$

$$\boldsymbol{\mu}_k \equiv J_k\delta\mathbf{I}'_k \tag{8.70}$$

The k-th rigid transform invariant becomes

$$\boldsymbol{\epsilon}_k = \mathbf{0} = \mathbf{y}_k - H_k\delta\mathbf{x}_0 - \boldsymbol{\mu}_k \tag{8.71}$$

which is equivalent to the linear state-observation model

$$\mathbf{y}_k = H_k\delta\mathbf{x}_0 + \boldsymbol{\mu}_k \tag{8.72}$$

Stacking all these vectors and matrices in dedicated containers

$$\boldsymbol{\mu} \equiv \begin{pmatrix}\boldsymbol{\mu}_1\\ \vdots \\ \boldsymbol{\mu}_{N-1}\end{pmatrix},\ \mathbf{y} \equiv \begin{pmatrix}\mathbf{y}_1\\ \vdots \\ \mathbf{y}_{N-1}\end{pmatrix},\ H \equiv \begin{pmatrix}H_1\\ \vdots \\ H_{N-1}\end{pmatrix} \tag{8.73}$$

The above equations become

$$\mathbf{y} = H\delta\mathbf{x}_0 + \boldsymbol{\mu} \tag{8.74}$$

Since the first variation of $\mathbf{y}$ reads

$$\delta\mathbf{y} = \left[\frac{\partial\mathbf{y}}{\partial\mathbf{I}'}\right]\left[\frac{\partial\mathbf{I}'}{\partial\tilde{\mathbf{V}}}\right]\left[\frac{\partial\tilde{\mathbf{V}}}{\partial\tilde{\mathbf{I}}}\right]\delta\tilde{\mathbf{I}} \tag{8.75}$$

its covariance is given by

$$R \equiv \mathrm{E}\left(\mathbf{y}\mathbf{y}^T\right) \tag{8.76}$$

$$= \left[\frac{\partial\mathbf{y}}{\partial\mathbf{I}'}\right]\left[\frac{\partial\mathbf{I}'}{\partial\tilde{\mathbf{V}}}\right]\left[\frac{\partial\tilde{\mathbf{V}}}{\partial\tilde{\mathbf{I}}}\right]P_{\mathbf{I}}\left(\left[\frac{\partial\mathbf{y}}{\partial\mathbf{I}'}\right]\left[\frac{\partial\mathbf{I}'}{\partial\tilde{\mathbf{V}}}\right]\left[\frac{\partial\tilde{\mathbf{V}}}{\partial\tilde{\mathbf{I}}}\right]\right)^T \tag{8.77}$$

$$P_{\mathbf{I}} \equiv \mathrm{E}\left(\delta\tilde{\mathbf{I}}\delta\tilde{\mathbf{I}}^T\right) \tag{8.78}$$

The weighted-least square solution of the orbit determination problem, $\delta\hat{\mathbf{x}}_0 = \min\limits_{\delta\mathbf{x}_0} \boldsymbol{\mu}^T R^{-1} \boldsymbol{\mu}$ is thus given by

$$\left(H^T R^{-1} H\right) \delta\mathbf{x}_0 = H^T R^{-1} \mathbf{y} \tag{8.79}$$

The solved-for state deviation $\delta\mathbf{x}_0$ is added to the state $\mathbf{x}_0$ at the epoch time and the procedure is repeated until the filter converges.

8.1.5 State-observation correlations

Since J_k, (and by consequence $\left[\frac{\partial\mathbf{y}}{\partial\mathbf{I}'}\right]$) is a function of $\bar{\mathbf{r}}_{k+1}$, there will exist a correlation between $\boldsymbol{\mu}$ and $\delta\mathbf{x}_0$, in violation of the typical assumptions made in the statistical derivation of the Batch filter [118]. However, these correlations vanish if the uncertainties on the rotational components of $\tilde{\mathbf{I}}$ are zero.

8.1.6 Initialization of the Batch filter

Due to its non-linear nature, the Batch approach summarized in Equation (8.79) must be initialized reasonably well so as to have the Batch operating in a linear regime about the true state. In order to do so, a suitable a-priori state

$$\bar{\mathbf{x}}_0 = \begin{pmatrix} \bar{\mathbf{r}}_0 \\ \bar{\dot{\mathbf{r}}}_0 \\ \bar{\mu} \end{pmatrix} \tag{8.80}$$

must be provided. One should first note that the knowledge of the position relative to the center of mass yields the position at all subsequent times. Indeed, the first rigid transform invariant reads

$$\mathbf{r}_0 - M_1'\,\mathbf{r}_1 + \mathbf{X}_1' = \mathbf{0} \tag{8.81}$$

so

$$\mathbf{r}_1 = M_1^{T'}\left(\mathbf{r}_0 + \mathbf{X}_1'\right) \tag{8.82}$$

This procedure can be repeated for all the subsequent measurements. A very crude first guess of $\mathbf{r}_0$ can be obtained from the averaging of all the registered point clouds at the final observation time:

$$\bar{\bar{\mathbf{r}}}_0 \equiv -\frac{1}{\sum_{i=0}^{p} N_i} \sum_{i=0}^{p} \sum_{j=1}^{N_j} \mathbf{S}_j \tag{8.83}$$

A crude guess of the position at the time following the epoch is thus

$$\bar{\bar{\mathbf{r}}}_1 = M_1^{T'} \left(\bar{\bar{\mathbf{r}}}_0 + \mathbf{X}_1' \right) \tag{8.84}$$

and that of the initial velocity by

$$\dot{\bar{\bar{\mathbf{r}}}}_0 \equiv \frac{\bar{\bar{\mathbf{r}}}_1 - \bar{\bar{\mathbf{r}}}_0}{t_1 - t_0} \tag{8.85}$$

These values are too crude to be passed as-is to the filter. However, a relatively inexpensive refinement of these states can be obtained by applying a metaheuristic Particle-Swarm-Algorithm [101] to the minimization of $K = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}$. The functioning of PSO can be summarized like so : assuming that one is seeking to find the global extremum of $K : \mathbf{x} \in \mathbb{R}^p \mapsto K(\mathbf{x}) \in \mathbb{R}$ over $\mathbb{R}^p$, PSO proceeds by sampling a number of test values in $\mathbb{R}^p$, the *particles*. J is evaluated at each particle, before an information exchange phase takes place. At the end of this phase, the particles' position in state-space are updated based on: the *global* best state found by the population, the *local* best state found by each particle, and an *inertia* term accounting for the motion of the particles within the state-space. The optimizer does not stall if local minima are encountered, provided that the state space is sufficiently populated.

In the present case, the search space on $\mathbf{r}_0$ and $\dot{\mathbf{r}}_0$ is bound by square boxes of size 200 meters and $\frac{200}{t_0 - t_1}$ meters per second, and bound μ between 1 and 3 kg km^3 s^{-2} (the true μ of Itokawa being around 2.35 kg km^3 s^{-2}). A slight variation from the classical PSO is that $\mathbf{r}_0$ and $\dot{\mathbf{r}}_0$ are sampled normally about their crude values, as opposed to uniformly. The PSO was then ran for a fixed number of 30 iterations featuring 500 particles. The best-state found by the PSO was then dubbed $\bar{\mathbf{x}}_0$ and passed to the Batch filter for further refinement and covariance computation.

8.1.7 Uncertainty on closest-approach radius

A metric that is fundamental to spacecraft safety is the radius of perigee $r_p = a(1-e)$. The first variation in r_p is given by

$$\delta r_p = \begin{bmatrix} \frac{\partial r_p}{\partial \mathbf{r}_0} & \frac{\partial r_p}{\partial \dot{\mathbf{r}}_0} & \frac{\partial r_p}{\partial \mu} \end{bmatrix} \begin{pmatrix} \delta \mathbf{r}_0 \\ \delta \dot{\mathbf{r}}_0 \\ \delta \mu \end{pmatrix} \tag{8.86}$$

$$= \begin{bmatrix} \frac{\partial r_p}{\partial a} & \frac{\partial r_p}{\partial e} \end{bmatrix} \begin{bmatrix} \frac{\partial a}{\partial \mathbf{r}_0} & \frac{\partial a}{\partial \dot{\mathbf{r}}_0} & \frac{\partial a}{\partial \mu} \\ \frac{\partial e}{\partial \mathbf{r}_0} & \frac{\partial e}{\partial \dot{\mathbf{r}}_0} & \frac{\partial e}{\partial \mu} \end{bmatrix} \begin{pmatrix} \delta \mathbf{r}_0 \\ \delta \dot{\mathbf{r}}_0 \\ \delta \mu \end{pmatrix} \tag{8.87}$$

The variance on the closest-approach radius is thus given by

$$\mathbf{E}\left(\delta r_p^2\right) = \begin{bmatrix} \frac{\partial r_p}{\partial a} & \frac{\partial r_p}{\partial e} \end{bmatrix} \begin{bmatrix} \frac{\partial a}{\partial \mathbf{r}_0} & \frac{\partial a}{\partial \dot{\mathbf{r}}_0} & \frac{\partial a}{\partial \mu} \\ \frac{\partial e}{\partial \mathbf{r}_0} & \frac{\partial e}{\partial \dot{\mathbf{r}}_0} & \frac{\partial e}{\partial \mu} \end{bmatrix} P_{\mathbf{x}} \left(\begin{bmatrix} \frac{\partial r_p}{\partial a} & \frac{\partial r_p}{\partial e} \end{bmatrix} \begin{bmatrix} \frac{\partial a}{\partial \mathbf{r}_0} & \frac{\partial a}{\partial \dot{\mathbf{r}}_0} & \frac{\partial a}{\partial \mu} \\ \frac{\partial e}{\partial \mathbf{r}_0} & \frac{\partial e}{\partial \dot{\mathbf{r}}_0} & \frac{\partial e}{\partial \mu} \end{bmatrix} \right)^T \tag{8.88}$$

where

$$P_{\mathbf{x}} \equiv \left(H^T W H\right)^{-1} \tag{8.89}$$

The partial derivatives featured in the expression of δr_p directly stem from the expression of the conservation of energy and eccentricity under Keplerian dynamics, and present little challenge or interest. The spacecraft is thus at risk of colliding with the orbited object if its periapse radius is less than the radius of the circumscribing sphere of the orbited small body.

8.1.8 Results

The proposed initial orbit determination approach is demonstrated in the context of a mission in the close vicinity of asteroid Itokawa, whose circumscribing radius is equal to 308 m. The asteroid was undergoing a principal rotation about the $\hat{Z}$ axis of the barycentered initial frame at a fixed period of 12 hours. In what follows, the absolute rigid transforms $\left(\tilde{M}_k, \tilde{\mathbf{X}}_k\right)$ are prescribed as normally distributed about their true value. The covariance and correlation of the zero-mean registration errors are given on Table 8.1. Although the magnitude of the rotational registration error may appear small, its effect is greatly amplified by the relative distance separating the instrument to the measured point cloud. In practice, the prescribed translational and rotational registration errors have comparative magnitudes at a distance of 1000 meters.

Table 8.1: Second statistical moments of the registration errors. No time correlations is considered between the absolute rigid transform errors

Quantity	Amount	Unit
$\mathrm{E}\left(\delta\tilde{\mathbf{X}}_k \delta\tilde{\mathbf{X}}_k^T\right)$	I_3	m^2
$\mathrm{E}\left(\delta\tilde{\boldsymbol{\sigma}}_k \delta\tilde{\boldsymbol{\sigma}}_k^T\right)$	$10^{-8} I_3$	-
$\mathrm{E}\left(\delta\tilde{\boldsymbol{\sigma}}_k \delta\tilde{\mathbf{X}}_k^T\right)$	0_{33}	m

Table 8.2: Simulation parameters, low eccentricity

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
e	0.25	0.25	0.25	0.25	0.25	0.25
Orbit fraction	0.25	0.25	0.25	0.50	0.50	0.50
N	5.00	7.00	10.00	5.00	7.00	10.00

Table 8.3: Simulation parameters, deep-impact trajectory

	Case 7	Case 8	Case 9	Case 10	Case 11	Case 12
e	0.75	0.75	0.75	0.75	0.75	0.75
Orbit fraction	0.25	0.25	0.25	0.50	0.50	0.50
N	5.00	7.00	10.00	5.00	7.00	10.00

Table 8.4: Simulation parameters, shallow-impact trajectory

	Case 13	Case 14	Case 15	Case 16	Case 17	Case 18
e	0.69	0.69	0.69	0.69	0.69	0.69
Orbit fraction	0.25	0.25	0.25	0.50	0.50	0.50
N	5.00	7.00	10.00	5.00	7.00	10.00

Several approach scenarii are simulated. They differ by the orbit eccentricity e, the fraction of the orbit that the observation arc covers and the number of collected point clouds N. The different scenarii parameters are listed on Table 8.2, 8.3 and 8.4. All scenarios share the same orbit-semi major axis ($a = 1000$ m), inclination ($i = 1.4$ rad), right-ascension of ascending node ($\Omega = 0.2$ rad), longitude of perigee ($\omega = 0.3$ rad) and mean anomaly at epoch ($M_0 = 3$ rad). All trajectories thus begin close to the apoapse of the considered orbits. Three families of scenarii are investigated. The first family correspond to a low-eccentricity trajectory. The second family puts the spacecraft on a collision course with the asteroid with $r_p = 250$ m . The third family corresponds to a more shallow collision course with $r_p = 305$ m.

A 600-run Monte-Carlo simulation was run for each case. The second-moment about the mean of the 600 state estimates was computed from their distribution and shown on Tables 8.5,

Table 8.5: Monte-Carlo outcome distribution statistics, low eccentricity cases

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
σ_X (m)	0.691	0.560	0.477	0.501	0.467	0.350
σ_Y (m)	1.216	1.016	0.907	0.454	0.369	0.297
σ_Z (m)	13.100	11.115	9.348	2.903	2.936	2.435
$\sigma_{\dot{X}}$ (mm/s)	0.189	0.154	0.141	0.038	0.031	0.025
$\sigma_{\dot{Y}}$ (mm/s)	0.124	0.105	0.095	0.030	0.028	0.021
$\sigma_{\dot{Z}}$ (mm/s)	0.138	0.118	0.096	0.065	0.060	0.050
σ_μ (m^3/s^2)	0.021	0.018	0.015	0.009	0.009	0.007

8.6 and 8.7. The predicted statistics coming from the inverse of $P_{\mathbf{x}} = \left(H^T W H\right)^{-1}$ for each Monte-Carlo outcome were compared to the computed ones, with their average relative difference shown on Tables 8.8, 8.9 and 8.10.

All Monte-Carlo runs for each case converged to a state estimate. For the sake of illustration, detailed results of Case 15 are shown on Figure 8.3. Tables 8.5, 8.6 and 8.7 clearly show that an increase in the number of observations or in the observation arc duration yields smaller variances. Even more importantly, Tables 8.8, 8.9 and 8.10 demonstrate the good agreement between the prediction of the state uncertainties and the effective ones. The larger position error along the $\hat{Z}$ direction is effectively lined up with the asteroid's rotation axis. Since the principal rotation implies that all M'_k roughly share the same principal rotation axis, any constant error along the rotation axis will cancel out in $\boldsymbol{\epsilon}$. The problem nonetheless remains observable thanks to the dynamics that prevent constant position biases to remain undetected.

Tables 8.11, 8.12 and 8.13 furthermore demonstrate that an overwhelming majority of the MC outcomes yielded a radius-of-perigee estimation error consistent with the effective one. The proposed filter is thus suitable to initial orbit determination about an unknown small body, thanks to its ability to detect potential collision courses.

Table 8.6: Monte-Carlo outcome distribution statistics, deep-impact trajectory cases

	Case 7	Case 8	Case 9	Case 10	Case 11	Case 12
σ_X (m)	0.827	0.677	0.551	0.519	0.419	0.337
σ_Y (m)	1.303	1.112	0.895	0.501	0.420	0.329
σ_Z (m)	23.451	20.648	17.436	4.063	2.841	2.393
$\sigma_{\dot{X}}$ (mm/s)	0.247	0.205	0.164	0.039	0.033	0.028
$\sigma_{\dot{Y}}$ (mm/s)	0.136	0.118	0.098	0.024	0.022	0.016
$\sigma_{\dot{Z}}$ (mm/s)	0.106	0.091	0.077	0.059	0.043	0.038
σ_μ (m^3/s^2)	0.039	0.034	0.028	0.011	0.008	0.007

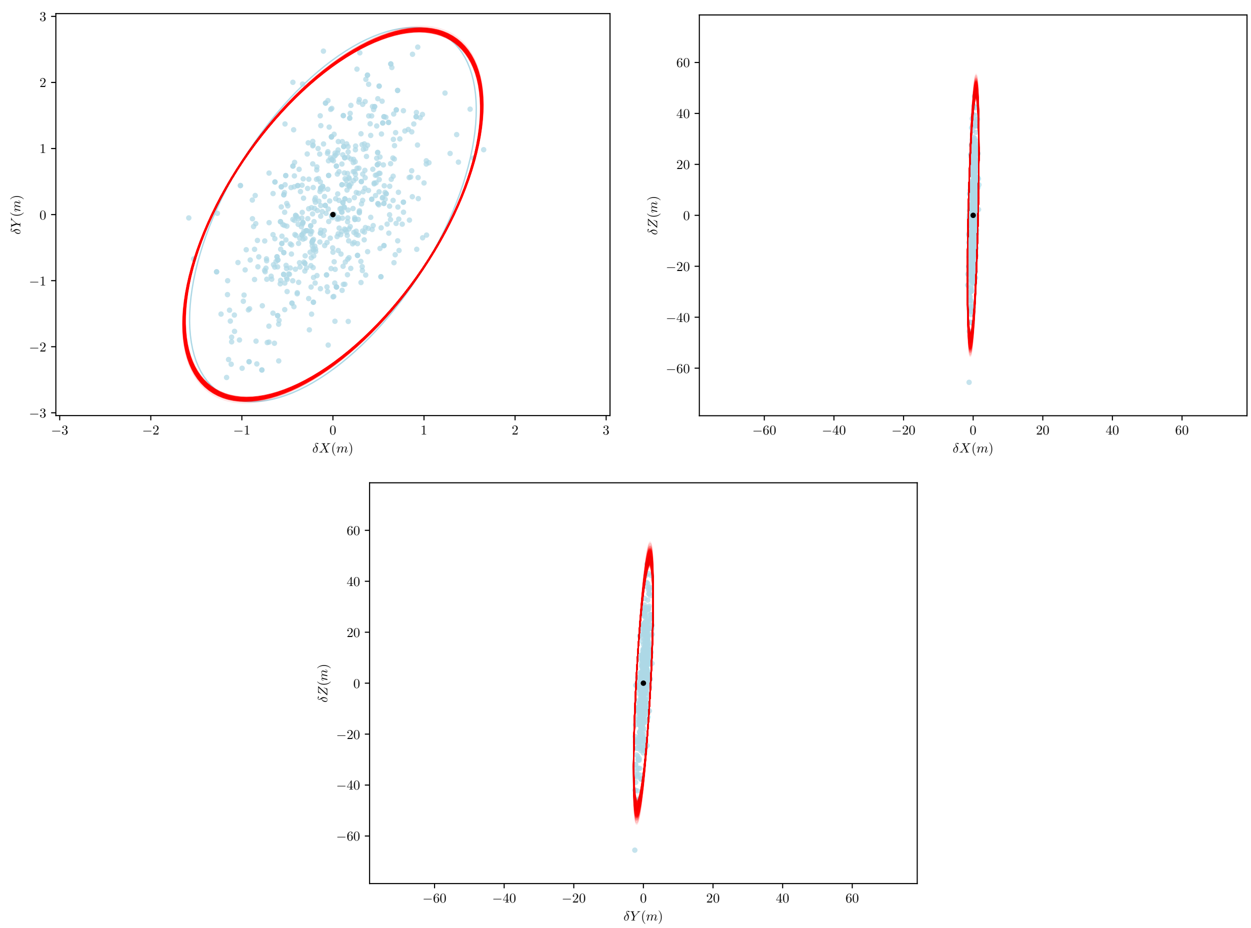
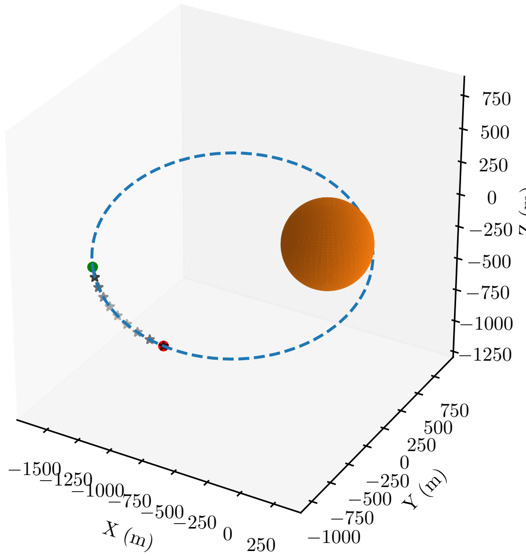


Figure 8.3: 3σ position covariance envelopes predicted from the $P_{\mathbf{x}}$ (red) and those computed from the state estimate distribution (lightblue), Case 15. The good agreement between the ensemble of predicted covariances and the one arising from the estimate distribution is apparent



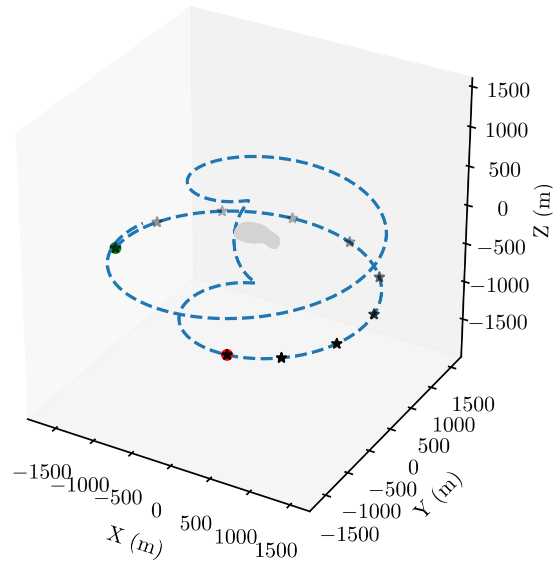


Figure 8.4: Inertial trajectory (left) and body-frame trajectory (right), Case 15. The orange sphere represents the circumscribing radius measured from the center of mass of the asteroid. The black stars denote observation times. The green and red dot represent the beginning and end of the observation arc respectively.

Table 8.7: Monte-Carlo outcome distribution statistics, shallow-impact trajectory cases

	Case 13	Case 14	Case 15	Case 16	Case 17	Case 18
σ_X (m)	0.838	0.620	0.525	0.516	0.412	0.352
σ_Y (m)	1.280	1.004	0.944	0.489	0.413	0.328
σ_Z (m)	23.070	18.403	16.612	4.184	3.081	2.577
$\sigma_{\dot{X}}$ (mm/s)	0.250	0.181	0.166	0.041	0.032	0.027
$\sigma_{\dot{Y}}$ (mm/s)	0.139	0.107	0.101	0.025	0.021	0.016
$\sigma_{\dot{Z}}$ (mm/s)	0.112	0.094	0.079	0.062	0.050	0.041
σ_μ (m^3/s^2)	0.036	0.030	0.026	0.011	0.009	0.007

Table 8.8: Average relative difference between MC distribution statistics and predicted statistics, low eccentricity cases

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$\Delta\sigma_X$ (%)	2.965	2.786	1.262	7.625	3.841	7.495
$\Delta\sigma_Y$ (%)	0.420	0.646	3.469	2.443	3.318	5.350
$\Delta\sigma_Z$ (%)	2.691	3.418	2.684	7.143	3.182	4.933
$\Delta\sigma_{\dot{X}}$ (%)	0.572	2.295	4.661	5.519	1.180	2.898
$\Delta\sigma_{\dot{Y}}$ (%)	0.937	0.339	4.555	6.151	2.938	8.858
$\Delta\sigma_{\dot{Z}}$ (%)	1.258	4.368	1.794	2.364	2.265	2.629
$\Delta\sigma_\mu$ (%)	3.262	2.266	1.539	5.385	2.724	2.875

Table 8.9: Average relative difference between MC distribution statistics and predicted statistics, deep-impact trajectory cases

	Case 7	Case 8	Case 9	Case 10	Case 11	Case 12
$\Delta\sigma_X$ (%)	2.881	0.366	0.280	0.907	0.855	2.683
$\Delta\sigma_Y$ (%)	0.550	1.555	4.952	1.519	2.552	2.510
$\Delta\sigma_Z$ (%)	4.601	3.172	3.050	3.301	9.047	4.166
$\Delta\sigma_{\dot{X}}$ (%)	0.673	1.518	3.935	5.856	1.272	1.459
$\Delta\sigma_{\dot{Y}}$ (%)	0.836	1.081	3.265	0.069	10.866	0.331
$\Delta\sigma_{\dot{Z}}$ (%)	3.092	1.526	1.739	1.613	11.137	1.022
$\Delta\sigma_\mu$ (%)	0.526	2.538	3.728	1.146	10.349	1.748

Table 8.10: Average relative difference between MC distribution statistics and predicted statistics, shallow-impact trajectory cases

	Case 13	Case 14	Case 15	Case 16	Case 17	Case 18
$\Delta\sigma_X$ (%)	0.430	8.169	3.851	1.394	1.146	2.569
$\Delta\sigma_Y$ (%)	0.782	8.211	1.328	1.532	0.373	3.360
$\Delta\sigma_Z$ (%)	3.901	4.951	2.607	1.743	3.155	1.135
$\Delta\sigma_{\dot{X}}$ (%)	3.750	9.666	1.144	0.542	2.802	1.659
$\Delta\sigma_{\dot{Y}}$ (%)	2.167	7.510	0.582	1.800	4.275	0.689
$\Delta\sigma_{\dot{Z}}$ (%)	1.004	0.485	1.003	2.917	1.022	0.176
$\Delta\sigma_\mu$ (%)	2.812	1.037	0.237	1.220	2.782	0.331

Table 8.11: Agreement between radius-of-perigee estimation error and effective error, low eccentricity cases

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$\|r_p - \hat{r}_p\| \leq 3\sigma_{rp}$ (%)	99.667	99.833	99.500	100.000	100.000	99.667
$\|r_p - \hat{r}_p\| > 3\sigma_{rp}$ (%)	0.333	0.167	0.500	0.000	0.000	0.333

Table 8.12: Agreement between radius-of-perigee estimation error and effective error, deep-impact trajectory cases

	Case 7	Case 8	Case 9	Case 10	Case 11	Case 12
$\|r_p - \hat{r}_p\| \leq 3\sigma_{rp}$ (%)	99.333	99.833	100.000	99.833	99.500	99.833
$\|r_p - \hat{r}_p\| > 3\sigma_{rp}$ (%)	0.667	0.167	0.000	0.167	0.500	0.167

Table 8.13: Agreement between radius-of-perigee estimation error and effective error, shallow-impact trajectory cases

	Case 13	Case 14	Case 15	Case 16	Case 17	Case 18
$\|r_p - \hat{r}_p\| \leq 3\sigma_{rp}$ (%)	99.833	100.000	99.833	100.000	99.667	100.000
$\|r_p - \hat{r}_p\| > 3\sigma_{rp}$ (%)	0.167	0.000	0.167	0.000	0.333	0.000

8.2 Relative Navigation using flash-Lidar Data

A short overview of the model-based navigation filter of Dietrich et al. is provided below [119]

1. At every observation time, a computed point cloud is generated from the on-board shape model and on-board a-priori spacecraft state. This point cloud effectively produces a number of computed range measurements $\bar{\rho}_i$, to be compared with the noisy range measurements produced by the Lidar instrument $\tilde{\rho}_i$
2. The range residuals $\tilde{\rho}_i - \bar{\rho}_i$ are processed so as to extract the estimated deviation in the spacecraft position $\delta\hat{\mathbf{r}}$. The latter is solved by means of an iterated batch filter, which effectively returns a position measurement and an associated measurement covariance matrix $R_{\mathbf{r}}$
3. The position measurement and its covariance $R_{\mathbf{r}}$ are provided to an Extended Kalman Filter, which updates the estimated states

This thesis extends the methods developed by Dietrich et al. in several ways. First, the estimated state is augmented with the small body attitude, angular velocity, standard-gravitational parameter and spacecraft cannonball SRP model. Second, the small body shape model is no longer exactly known. Third, the uncertainty model developed in the previous section is used within the filter to account for shape reconstruction error and sensor noise.

8.2.1 Position estimation from range measurements

An individual range measurement reads

$$\rho_i = \hat{u}_i^T \left(\mathbf{P} - \mathbf{r}\right) \tag{8.90}$$

where $\mathbf{P}$ is a surface point on the ray-traced object, $\mathbf{r}$ being the spacecraft position relative to the object center $\mathbf{O}$ and $\hat{u}_i$ the unit vector directing the ray. Assuming that the ray-traced element is a Bezier triangle, the surface point coordinates $\mathbf{P}$ are given by

$$\mathbf{P} = \mathbf{P}(\boldsymbol{\chi}) \equiv \sum_{|\mathbf{i}|=n} B_{\mathbf{i}}^{n}(\boldsymbol{\chi})\,\mathbf{C}_{\mathbf{i}} \tag{8.91}$$

Introducing a slight deviation in the spacecraft position $\delta\mathbf{r}$ compared to the nominal position $\bar{\mathbf{r}}$, a first order expansion of the range yields

$$\rho = \hat{u}^T(\mathbf{P}(\boldsymbol{\chi}) - \mathbf{r}) \tag{8.92}$$

$$= \hat{u}^T(\mathbf{P}(\bar{\boldsymbol{\chi}} + \delta\boldsymbol{\chi}) - \mathbf{r} - \delta\mathbf{r}) \tag{8.93}$$

$$= \hat{u}^T\left(\mathbf{P}(\bar{\boldsymbol{\chi}}) + \left[\frac{\partial \mathbf{P}}{\partial \boldsymbol{\chi}}\right]\delta\boldsymbol{\chi} - \mathbf{r} - \delta\mathbf{r}\right) \tag{8.94}$$

$$= \bar{\rho} + \hat{u}^T\left(\left[\frac{\partial \mathbf{P}}{\partial \boldsymbol{\chi}}\right]\delta\boldsymbol{\chi} - \delta\mathbf{r}\right) \tag{8.95}$$

$\left[\frac{\partial \mathbf{P}}{\partial \boldsymbol{\chi}}\right] = [\mathbf{E}_0, \mathbf{E}_1]$ is a 3×2 matrix holding two vectors tangent to the surface at $\bar{\mathbf{P}}$. As a result,

$$\hat{n}^T\delta\mathbf{P} = \hat{n}^T\left[\frac{\partial \mathbf{P}}{\partial \boldsymbol{\chi}}\right]\delta\boldsymbol{\chi} = 0 = \hat{n}^T\hat{u}\delta\rho + \hat{n}^T\delta\mathbf{r} \tag{8.96}$$

which gives the same partial derivative in the planar facet case :

$$\frac{\partial \rho}{\partial \mathbf{r}} = -\frac{\mathbf{n}^T}{\mathbf{n}^T\hat{u}} = -\frac{\hat{n}^T}{\hat{n}^T\hat{u}} \tag{8.97}$$

where

$$\mathbf{n} = \mathbf{E}_0 \times \mathbf{E}_1 \tag{8.98}$$

However, there exists a difference when the ray-traced shape is a Bezier patch of degree strictly greater than one. Indeed, the partial derivative in Equation (8.97) is erroneous as it is unable to capture the curvature on the surface element. Figure 8.5 demonstrates this claim, as the true $\delta\mathbf{P}$ does not match the one computed from the first-order approximation $\delta\mathbf{P}^{(1)}$: the second-order error $\alpha\hat{u}$ is not captured by the partial derivative. This second-order error reduces as the filter iterates.

8.2.2 Attitude estimation from range measurements

8.2.2.1 Formulation

Given a range measurement ρ between the spacecraft $\mathbf{S}$ and a surface point $\mathbf{P}$ along the direction $\hat{u}$

$$\rho = \hat{u}^T (\mathbf{P} - \mathbf{S}) \tag{8.99}$$

$$\rho = \frac{\hat{n}^T (\mathbf{V} - \mathbf{S})}{\hat{u}^T \hat{n}} \tag{8.100}$$

where $\mathbf{V}$ is a point belonging to the tangent plane of normal $\hat{n}$ at $\mathbf{P}$. Let $\mathcal{B}$ and $\mathcal{N}$ be the asteroid body-fixed and centered-inertial frames respectively. The goal is to estimate a parametrization of the direction cosine matrix $[\mathcal{BN}]$. Let $\mathcal{B}'$ be an intermediate frame such that

$$[\mathcal{BN}] = [\mathcal{BB}'][\mathcal{B}'\mathcal{N}] \tag{8.101}$$

Let $\boldsymbol{\sigma}_{\mathcal{B}'/\mathcal{N}}$ be the a-priori attitude:

$$\boldsymbol{\sigma}_{\mathcal{B}'/\mathcal{N}} = \bar{\boldsymbol{\sigma}} \tag{8.102}$$

The error MRP measuring the deviation between $\mathcal{B}$ and $\mathcal{B}'$ is written

$$\boldsymbol{\sigma}_{\mathcal{B}'/\mathcal{B}} = \boldsymbol{\sigma} \tag{8.103}$$

Under the assumption of a small error, the DCM $[\mathcal{BB}']$ can be linearized into [68]

$$[\mathcal{BB}'] \simeq I_3 - 4[\tilde{\boldsymbol{\sigma}}] \tag{8.104}$$

The normal $\hat{n}$ and the tangent point $\mathbf{V}$ are available in the $\mathcal{B}$ frame while the measurement direction $\hat{u}$ and the spacecraft position are expressed in the $\mathcal{N}$ frame. As a result, making the frame dependencies explicit in the range measurement yields

$$\rho = \frac{\hat{n}^{\mathcal{B},T} \left(\mathbf{V}^{\mathcal{B}} - \mathbf{S}^{\mathcal{B}}\right)}{\hat{n}^{\mathcal{B},T} \hat{u}^{\mathcal{B}}} \tag{8.105}$$

So

$$\rho = \frac{\hat{n}^{\mathcal{B},T}\left(\mathbf{V}^{\mathcal{B}} - \mathbf{S}^{\mathcal{B}}\right)}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}}} \tag{8.106}$$

$$\simeq \hat{n}^{\mathcal{B},T}\left(\mathbf{V}^{\mathcal{B}} - \left[I_3 - 4[\tilde{\boldsymbol{\sigma}}]\right]\mathbf{S}^{\mathcal{B}'}\right)\frac{1 + \frac{4\hat{n}^{\mathcal{B},T}[\tilde{\boldsymbol{\sigma}}]\hat{u}^{\mathcal{B}'}}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}'}}}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}'}} \tag{8.107}$$

$$\simeq \frac{\hat{n}^{\mathcal{B},T}}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}'}}\left(\mathbf{V}^{\mathcal{B}} - \mathbf{S}^{\mathcal{B}'} + 4[\tilde{\boldsymbol{\sigma}}]\mathbf{S}^{\mathcal{B}'}\right)\left(1 + \frac{4\hat{n}^{\mathcal{B},T}[\tilde{\boldsymbol{\sigma}}]\hat{u}^{\mathcal{B}'}}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}'}}\right) \tag{8.108}$$

$$\simeq \left(\bar{\rho} + \frac{4\hat{n}^{\mathcal{B},T}[\tilde{\boldsymbol{\sigma}}]\mathbf{S}^{\mathcal{B}'}}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}'}}\right)\left(1 + \frac{4\hat{n}^{\mathcal{B},T}[\tilde{\boldsymbol{\sigma}}]\hat{u}^{\mathcal{B}'}}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}'}}\right) \tag{8.109}$$

$$\simeq \bar{\rho} + \frac{4\hat{n}^{\mathcal{B},T}}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}'}}\left(-\mathbf{S}^{\mathcal{B}'} - \bar{\rho}\hat{u}^{\mathcal{B}'}\right) \times \boldsymbol{\sigma} \tag{8.110}$$

Therefore, the partial derivative relating the change in a range measurement to a change in the attitude is

$$\frac{\partial \rho}{\partial \boldsymbol{\sigma}} = -\frac{4\hat{n}^{\mathcal{B},T}}{\hat{n}^{\mathcal{B},T}\hat{u}^{\mathcal{B}'}}\widetilde{[\mathbf{P}^{\mathcal{B}}]} \tag{8.111}$$

8.2.2.2 MRP switching

The MRP set is typically switched to its shadow set once it reaches the switching surface defined by $\boldsymbol{\sigma}^T\boldsymbol{\sigma} = 1$. This ensures that the attitude set remains always at least 180 degrees away from its singularity. The covariance matrix tracking the uncertainty in the estimated state must be updated to reflect this. Kaalgard and Schaub provide the expression of the covariance switching matrix mapping the covariance matrix from before to after the switching [120]. Given an estimated state of the form $\mathbf{X} = \begin{pmatrix}\mathbf{r}^T & \dot{\mathbf{r}}^T & \boldsymbol{\sigma}^T & \boldsymbol{\omega}^T\end{pmatrix}^T$, and denote $\bar{P}$ and $\bar{P}^s$ the state covariance matrices before and after the switching,

$$\bar{P}^s = \Theta\bar{P}\Theta^T \tag{8.112}$$

with

$$\Theta = \begin{bmatrix} I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \frac{2\boldsymbol{\sigma}\boldsymbol{\sigma}^T}{(\boldsymbol{\sigma}^T\boldsymbol{\sigma})^2} - \frac{I_3}{\boldsymbol{\sigma}^T\boldsymbol{\sigma}} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_3 \end{bmatrix} \tag{8.113}$$

It can thus be seen than only the MRP partition of the covariance is affected by the switching.

8.2.3 Estimation of standard gravitational parameter

The standard gravitational parameter of the small body can be estimated using the spherical harmonics of the gravity field arising from the reconstructed shape, assuming a constant density distribution. The partial derivative of the velocity with respect to this state is trivial, as it is nothing else but the same acceleration of gravity emanating from the reconstructed shape model, evaluated with a unit μ. Because the spherical harmonics coefficients evaluated from the shape model are not re-estimated, it is possible that slight biases in μ may still show in the estimator.

8.2.4 Estimation of SRP cannonball coefficient

The cannonball coefficient C_r can be estimated through the dynamics, since the partial derivative of the velocity with respect to C_r is the same acceleration evaluated at $C_r = 1$. An interesting point is that the eclipse model used to toggle the SRP force depending on whether it is eclipsed behind the small makes the estimation of this state coupled with the estimated shape model.

8.2.5 Navigation filter and Iterated Extended Kalman Filter

The navigation filter estimates the spacecraft position $\mathbf{r}$, velocity $\dot{\mathbf{r}}$, small body attitude $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}}$ and angular velocity $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}$, small body standard gravitational parameter μ and spacecraft cannonball coefficient C_R by means of an Iterated Extended Kalman Filter (IEFK). The estimated state is thus formally defined as

$$\mathbf{X} = \begin{pmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \\ \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}} \\ \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \\ \mu \\ C_R \end{pmatrix} \tag{8.114}$$

The pseudo code describing the functioning of the IEKF is provided below in Algorithm 2, where $\hat{\phi}_{k-1}^{k}$ designates the integrated estimated dynamics between t_{k-1} and t_k and (Γ, Q) the process noise state-transition matrix and covariance respectively.

Algorithm 2 Navigation filter with embedded least-squares filter

procedure NavigationFilter

Initialization:

  Given: $\bar{\mathbf{X}}_0$, $\hat{P}_0$, $\bar{\mathcal{S}}$, Q, $\mathbf{G}$, H, ϕ, Γ, tol

Main loop:

  for k in $[\![1 \ldots N_{\text{times}}]\!]$ **do**

Time update:

    $\bar{\mathbf{X}}_k, \Phi_{k,k-1} \leftarrow \hat{\phi}_{k-1}^{k}(\hat{\mathbf{X}}_{k-1})$

    $\bar{P}_k \leftarrow \Phi_{k,k-1}\hat{P}_{k-1}\Phi_{k,k-1}^T + \Gamma_{k,k-1}Q\Gamma_{k,k-1}^T$

Measurement update:

    for i in $[\![1 \ldots N_{\text{iterations}}]\!]$ **do**

      $\tilde{\mathbf{Y}}_k^i, R_k^i \leftarrow \text{LeastSquares}\left(\bar{\mathbf{X}}_k^i, \bar{\mathcal{S}}; \mathbf{X}_k, \mathcal{S}\right)$

      $K_k^i \leftarrow \bar{P}_k H^T \left(H\bar{P}_k H^T + R_k^i\right)^{-1}$

      $\mathbf{Z}_k^i \leftarrow K_k^i \left(\tilde{\mathbf{Y}}_k^i - \mathbf{G}\left(\bar{\mathbf{X}}_k^i, t_k\right)\right)$

      $\bar{\mathbf{X}}_k^{i+1} \leftarrow \bar{\mathbf{X}}_k^i + \mathbf{Z}_k^i$

      $\hat{P}_{k,\text{temp}} \leftarrow \left(I_3 - K_k^i\right)\bar{P}_k\left(I_3 - K_k^i\right)^T + K_k^i R_k^i K_k^{iT}$

      $\gamma_k^i \leftarrow \mathbf{Z}_k^{iT}\hat{P}_{k,\text{temp}}^{-1}\mathbf{Z}_k^i$

      if $\frac{\gamma_{k,\text{old}}^i - \gamma_k^i}{\gamma_{k,\text{old}}^i} < \text{tol}$ **then**

        break

      else

        $\gamma_{k,\text{old}}^i \leftarrow \gamma_k^i$

    $\hat{P}_k \leftarrow \hat{P}_{k,\text{temp}}$

8.2.5.1 Observation model

The measurement model of the IEKF is

$$\mathbf{Y}_k = \mathbf{G}\left(\mathbf{X}_k, t_k\right) = \begin{pmatrix} \mathbf{r}(t_k) \\ \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}}(t_k) \end{pmatrix} \tag{8.115}$$

As a result, the state-observation matrix of the IEKF is constant and equal to

$$H = \begin{bmatrix} I_3 & 0_{33} & 0_{33} & 0_{33} & 0_{13} & 0_{13} \\ 0_{33} & 0_{33} & I_3 & 0_{33} & 0_{13} & 0_{13} \end{bmatrix} \tag{8.116}$$

More details on this observation model can be found in 8.2.6.

8.2.6 Least-squares filter

The task of the least-squares filter embedded within the navigation filter described in Algorithm 2 is to provide a measurement $\mathbf{Y}_k = \mathbf{G}\left(\mathbf{X}_k, t_k\right) = \begin{pmatrix} \mathbf{r}(t_k) \\ \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}}(t_k) \end{pmatrix}$ of the spacecraft position and asteroid attitude, along with a quantification of the measurement uncertainty by means of a covariance matrix R_k. $\mathbf{Y}_k$ is acquired by comparing range measurements over the asteroid to those computed on-board, using the asteroid a-priori state and reconstructed shape. The covariance matrix R_k accounts for both the sensor error and shape reconstruction error. Dietrich et al. determined that the uncertainty in the shape is best captured if looked at from a consider covariance perspective [121]. That is, any measure of the shape uncertainty should be incorporated into the filter by means of consider covariance analysis as opposed to simply inflating the Lidar noise standard-deviation, σ_{Lidar}. In the context of the least-squares filter, the shape error is simply modeled as a range bias of zero mean and standard deviation dictated by the directional uncertainty model previously derived. In this context, the consider covariance takes the form [122]

$$P = \Lambda^{-1} + SP_{cc}S^T \tag{8.117}$$

where P_{cc} stores the variances in the range measurements extracted from the uncertainty model and

$$S = \Lambda^{-1}H^TW \tag{8.118}$$

where H and W are the state-observation and weight matrices of the least-squares filter. The pseudo code of the least-squares filter is provided in Algorithm 3.

Algorithm 3 Position and attitude least-squares filter

procedure LeastSquares

Initialization:

  Given: $\bar{\mathbf{r}}(t_k)$, $\bar{\boldsymbol{\sigma}}_{\mathcal{B}/\mathcal{N}}(t_k)$, $\bar{\mathcal{S}}, \sigma_{\text{Lidar}}$

  $\bar{\mathbf{Y}}_k^0 \leftarrow \begin{pmatrix} \bar{\mathbf{r}}(t_k) \\ \bar{\boldsymbol{\sigma}}_{\mathcal{B}/\mathcal{N}}(t_k) \end{pmatrix}$

  $\tilde{\boldsymbol{\rho}}_k \leftarrow \text{Lidar}\left(\mathbf{Y}_k, \mathcal{S}\right)$

  $W \leftarrow \frac{1}{\sigma_{\text{Lidar}}^2} I_{N_{\text{ranges}}}$

Main loop:

  for i in $[\![1 \ldots N_{\text{iterations}}]\!]$ **do**

    $H \leftarrow 0_{N_{\text{ranges}} \times 6}$

    $\bar{\boldsymbol{\rho}}_k^i, \begin{pmatrix} \sigma_1^2 & \ldots & \sigma_{N_{\text{ranges}}}^2 \end{pmatrix} \leftarrow \text{Lidar}\left(\bar{\mathbf{Y}}_k^i, \bar{\mathcal{S}}\right)$

    $\delta\boldsymbol{\rho}_k^i \leftarrow \text{Subtract}\left(\tilde{\boldsymbol{\rho}}_k, \bar{\boldsymbol{\rho}}_k^i\right)$

    for p in $[\![1 \ldots N_{\text{ranges}}]\!]$ **do**

      $H(p,:) \leftarrow \begin{bmatrix} -\frac{\hat{n}_p^{\mathcal{N}',T}}{\hat{n}_p^{\mathcal{B},T}\hat{u}_p^{\mathcal{B}'}} & -\frac{4\hat{n}_p^{\mathcal{B},T}}{\hat{n}_p^{\mathcal{B},T}\hat{u}_p^{\mathcal{B}'}}\widetilde{\left[\mathbf{P}_p^{\mathcal{B}}\right]} \end{bmatrix}$

    $N \leftarrow H^T W \delta\boldsymbol{\rho}_k^i$

    $\Lambda \leftarrow H^T W H$

    $\bar{\mathbf{Y}}_k^i \leftarrow \text{Add}\left(\bar{\mathbf{Y}}_k^{i-1}, \Lambda^{-1} N\right)$

  $P_{cc} \leftarrow \text{diag}\begin{pmatrix} \sigma_1^2 & \ldots & \sigma_{N_{\text{ranges}}}^2 \end{pmatrix}$

  $S \leftarrow -\Lambda^{-1} H^T W$

  $R_k \leftarrow \Lambda^{-1} + S P_{cc} S^T$

return $\bar{\mathbf{Y}}_k^{N_{\text{iterations}}}$, R_k

8.2.7 Range measurement uncertainty

The diagonal consider covariance matrix P_{cc} used to augment the measurement covariance R_k in Algorithm (3) is computed from the tuned shape uncertainty model by populating each of the non-zeros components of P_{cc} with the consider variance obtained from Equation (5.25).

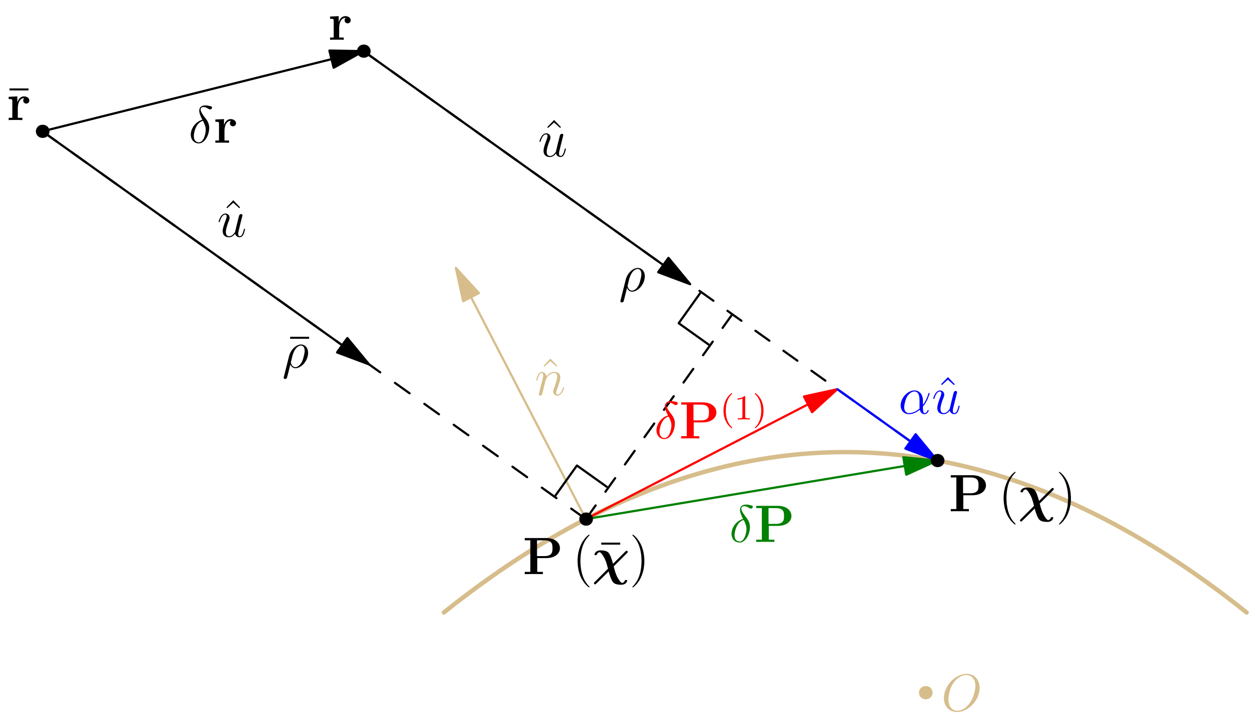


Figure 8.5: Measured ranges $\bar{\rho}$ and ρ before and after a displacement $\delta\mathbf{r}$ is added to the spacecraft location. α is non-zero when the patch degree is equal to one or when all control points are coplanar

Chapter 9

Mapping and navigation simulation results

This chapter applies the methods presented in Chapters 4, 5 and 8 to the survey, mapping and navigation of asteroid Itokawa in the context of a proximity mission. As a reminder to the reader, the key assumptions made in the developed framework are the perfect knowledge of the spacecraft attitude, the uniform-density nature of the orbited object as well as the knowledge of the small body ephemeris with respect to the Solar system barycenter. With this in mind, this chapter presents the results of the two phases - Survey/Mapping, and Navigation - when run with different sets of input parameters, such as the sensor noise, instrument frequency and rotational regime of the small body, so as to investigate the sensivity of the performance of the different in the relevant state-space parameters.

9.1 Survey and Mapping

The survey and mapping phase is concerned with the collection and registration of successive points clouds covering the targeted small body, the Itokawa 64 shape model [104]. The objective of this phase is to deliver a globally covering point cloud of the target, that is passed to the shape reconstruction pipeline described in Chapter 4. The survey and mapping phase thus produces a number of deliverables: the shape model, the uncertainty model capturing the shape reconstruction error, the spherical harmonics expansion arising from the shape, an a-priori navigation solution, a small body attitude state a-priori solution and a-priori standard gravitational parameter.

The results presented in this section explore the effect of a subspace of the phase's parameter space on the phase's deliverables. The goal of this sensitivity analysis is to identify the set of parameters that have the most effect on the pipeline convergence and quality. Although it is clear that testing every possible combination of input parameters is undeniably the most comprehensive way to carry out this analysis, the combinatorial explosions in the number of cases to consider calls for a different perspective. Two simulation subsets were run : the first one investigates the effect of camera frequency and retained number of points in the bundle-adjustment phase, while the second one assesses the robustness of the pipeline to non-principal rotation axis rotation and sensor noise.

It must be noted that the observation geometry introduces a coupling between the spacecraft position, attitude and small body coverage. That is, every initial spacecraft state will result in varying viewing geometries that will each yield a different global coverage of the small body, or lack thereof. Therefore, all cases investigated in this section are kicked-off with the same spacecraft initial state, so as to help defining the baseline case around which the sensitivity analysis can take place. The initial spacecraft cartesian state was obtained from the initial Keplerian state defined by the classical orbital elements $(a_0, e_0, i_0, \Omega_0, \omega_0, M_0)$, which are defined along with the other input parameters that are held constant in all simulations in Table 9.1.

9.1.1 Sensitivity to frequency and BA hierarchy level

The robustness of the framework was first investigated by varying the time elapsed between two successive point-cloud acquisitions T_{obs} along with the hierarchical setting h in the bundle adjustment. The integer h controls the maximum number of point-pairs that can be found between two point clouds respectively containing 2^{p_S} and 2^{p_D} points each, as in

$$N_{\text{pairs,max}} = \min\left(2^{p_S-h}, 2^{p_D-h}\right) \tag{9.1}$$

where $p_S - h > 0$ and $p_D - h > 0$. Setting h to a value greater than zero thus speeds up the bundle-adjustment phase by making it consider fewer point-pairs, at the potential expense of a diminished performance. The parameters that were varied in 9.1.1 are summarized in Table 9.2. All six cases were run using a one-sigma standard deviation on the range measurement of 0.5 m.

In addition, the asteroid was undergoing a constant-rate principal rotation about its largest inertia axis.

The globally covering point-clouds produced at the end of each scenarii do vary in size, as shown on Table 9.3. If the variation from Cases 1,2,3 to 4,5,6 was expected due to the greater number of observations collected in the latter cases, the variation that can be seen within each triplet is due by an incomplete loop closure at the final observation time, causing the last few point-clouds that have not been bundle-adjuster in a satisfactory fashion to be left-out of the globally covering point cloud.

The quality of the global point cloud acquired at the end of each simulation can be assessed by computing the Hausdorff distance from each of the point clouds to the true underlying asteroid shape model, with the resulting statistics shown on Table 9.4 . The distance distribution statistics appear to be relatively invariant across the considered cases, as exemplified by the side-by-side display of the distance-colored point clouds with the distance distribution histograms on Figure 9.1 and 9.2. The variation in the total simulation runtime shown on Table 9.6 appears to be more clearly related to the input parameters. It must be noted that this duration not only includes the accumulation of the successive point-clouds and their bundle-adjustment, but also the fitting of the Bezier Shape as well as the training of the underlying shape uncertainty model.

The quality of the Bezier shape models obtained in each case can be assessed in a similar manner than with the point clouds, by computing the Hausdorff distance from the Bezier shapes to the underlying polyhedral shape of the imaged small body. The statistics in the shape-to-shape Hausdorff distances are listed in Table 9.5, in addition to the side-by-side display of the distance-colored fit shapes with the distance distribution histograms on Figure 9.3 and 9.4. The quality of the shape fitting appears good in all cases, with the maximum shape fitting errors are located at the boulders that protrude the most out of the asteroid body. However, the apparent robustness of the proposed pipeline to varying instrument frequency and BA hierarchical level will only be fully assessed once the deliverables (that is, the shape model and its uncertainty model) are used to navigate about the small body.

Table 9.1: Parameters held constant throughout all survey and mapping simulation subsets

	Value	Unit
C_r	1.2	-
σ_{C_r}	0.1	-
a_0	1000	m
e_0	0.15	-
i_0	80	deg
Ω_0	0.57	deg
ω_0	17	deg
M_0	0	deg
T_{itokawa}	12	hours
ρ	1900	kg/m^3
$[\mathcal{BN}](t_0)$	I_3	-
$t_f - t_0$	200	hours
BA iterations	5	-
Shape fitter iterations	3	-
IOD-PSO iterations	100	-
IOD-PSO arc length	10	-
IOD-PSO particles	200	-
Edges in fit Bezier shape	2000	-

Table 9.2: Input parameters, Cases 1 to 6

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
h	0	1	2	0	1	2
T_{obs} (s)	3333	3333	3333	2500	2500	2500

Table 9.3: Size of globally-covering point cloud, Cases 1 to 6

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Point-cloud size	2,355,111	2,355,111	2,181,109	3,176,292	3,128,898	3,176,292

Table 9.4: Point-cloud-to-true-shape Hausdorff distance statistics, Case 1 to 6

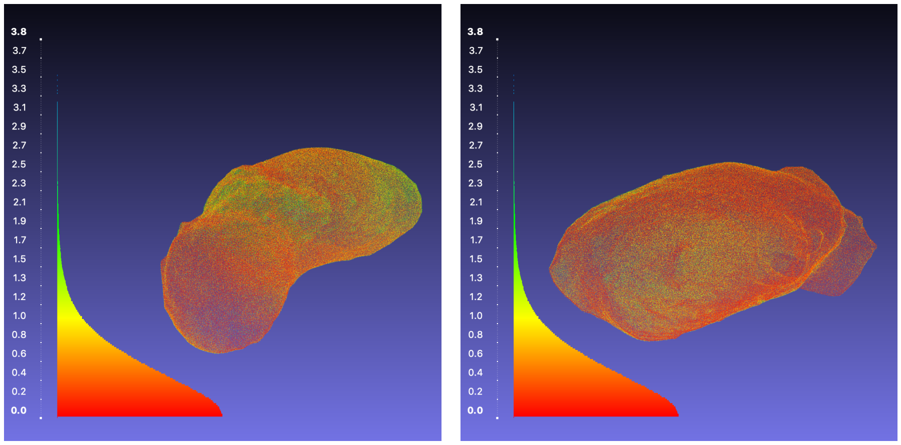
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Maximum (m)	3.849	3.019	3.444	3.742	3.7615	2.856
Mean (m)	0.435	0.372	0.376	0.428	0.4	0.378
Standard deviation (m)	0.365	0.302	0.312	0.338	0.323	0.299

Table 9.5: Fit-shape-to-true-shape Hausdorff distance statistics, Cases 1 to 6

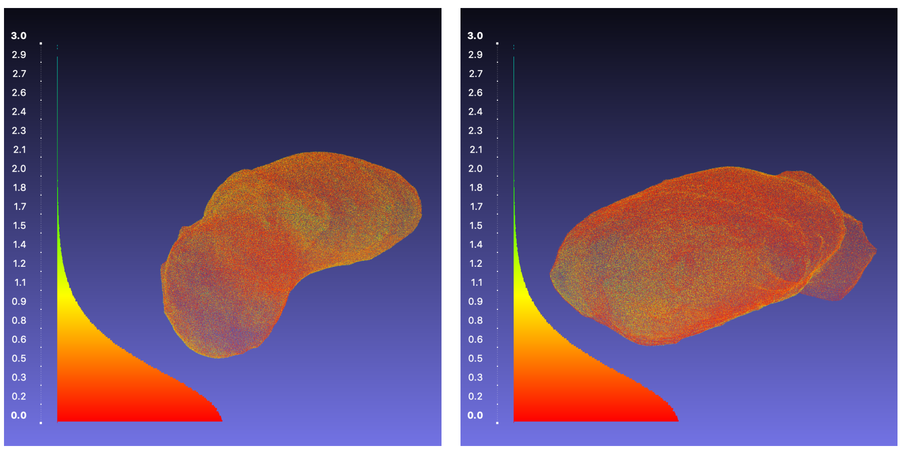
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Maximum (m)	5.116	3.965	3.589	4.703	3.958	2.848
Mean (m)	0.30185	0.264	0.2715	0.31076	0.3067	0.2963
Standard deviation (m)	0.2679	0.2414	0.2459	0.2767	0.263	0.2561

Table 9.6: Total simulation times, Cases 1 to 6

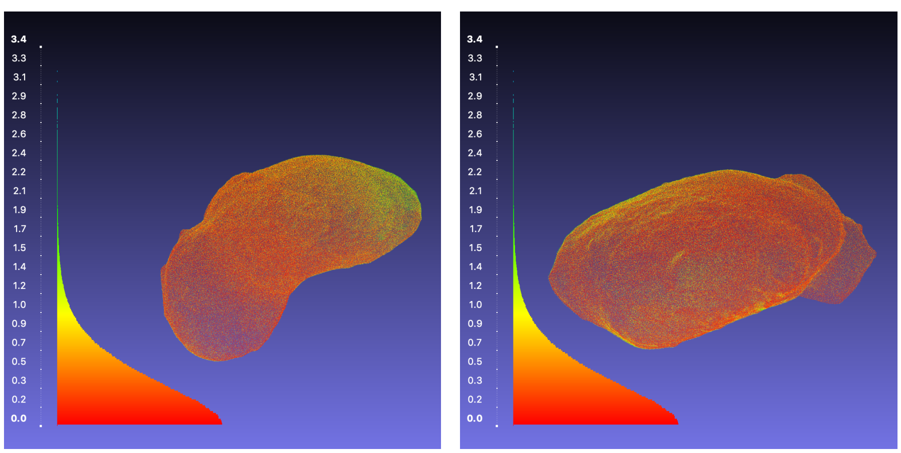
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Time (s)	1783.84	1529.95	1438.45	2542.48	2147.55	2094.81



(a) Case 1

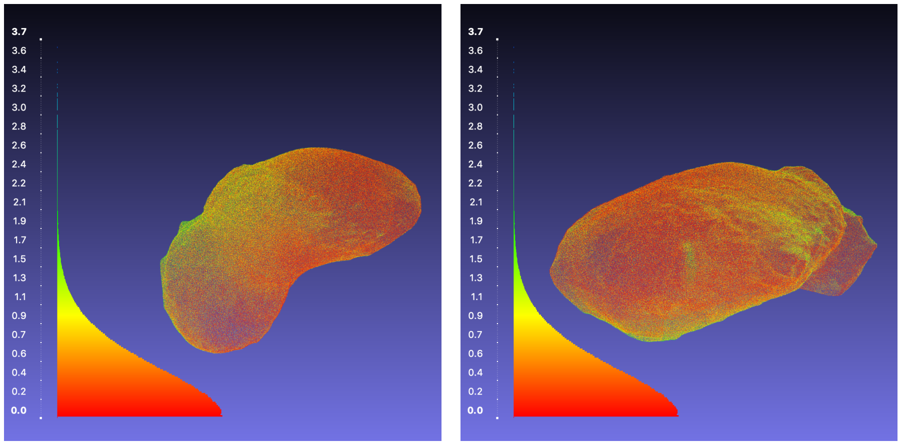


(b) Case 2

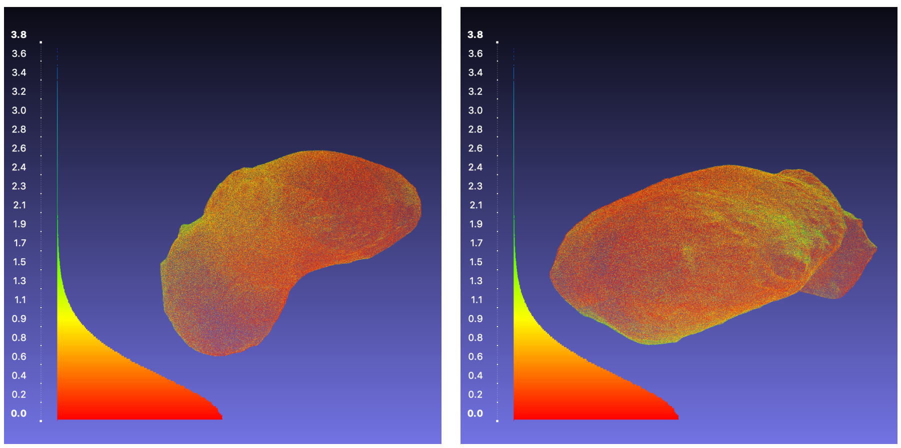


(c) Case 3

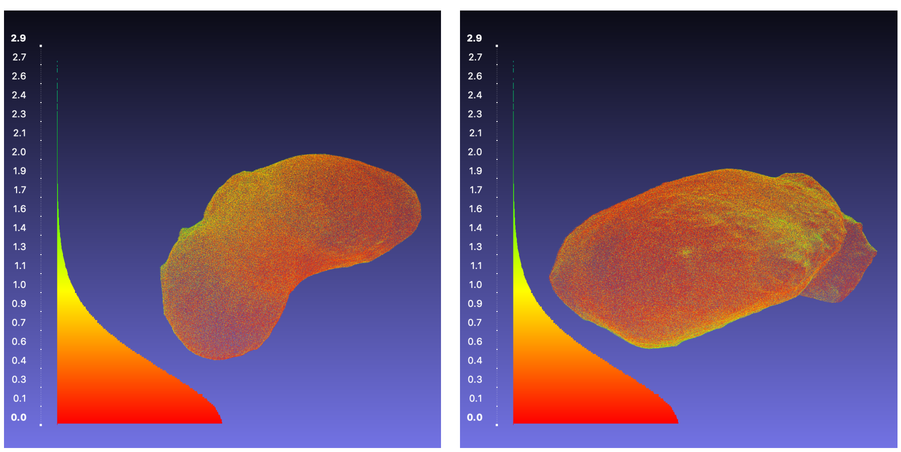
Figure 9.1: Normalized Hausdorff distance-colored point clouds and distribution histogram, Cases 1 to 3



(a) Case 4

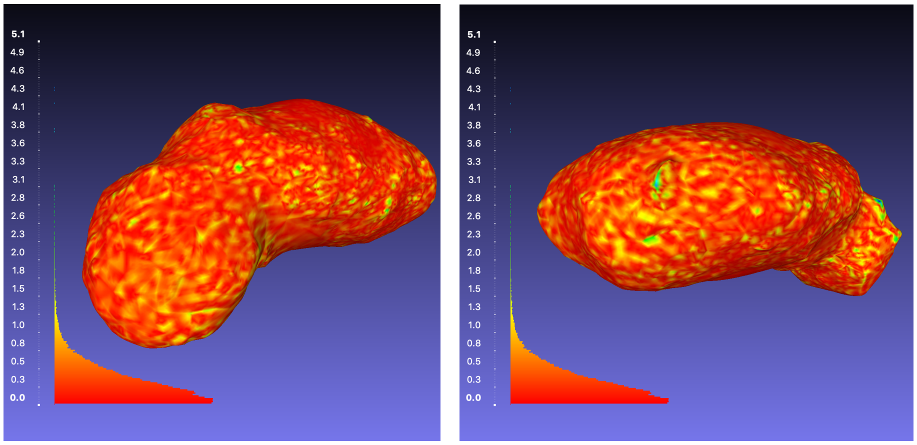


(b) Case 5

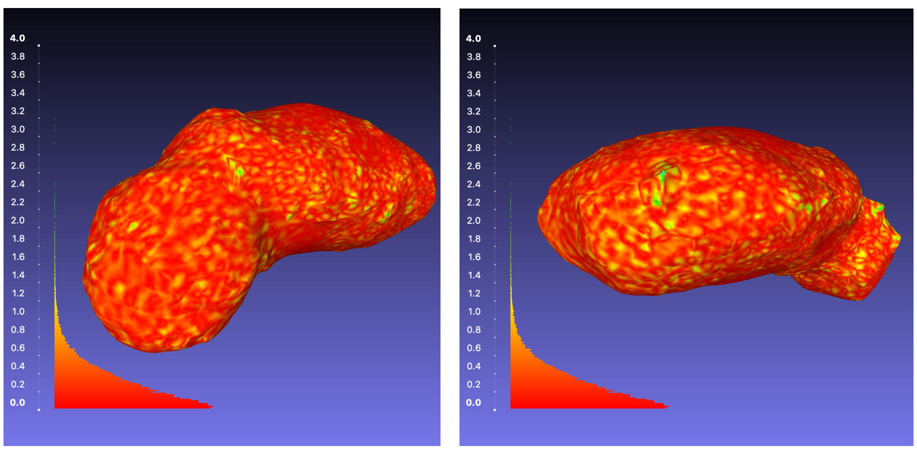


(c) Case 6

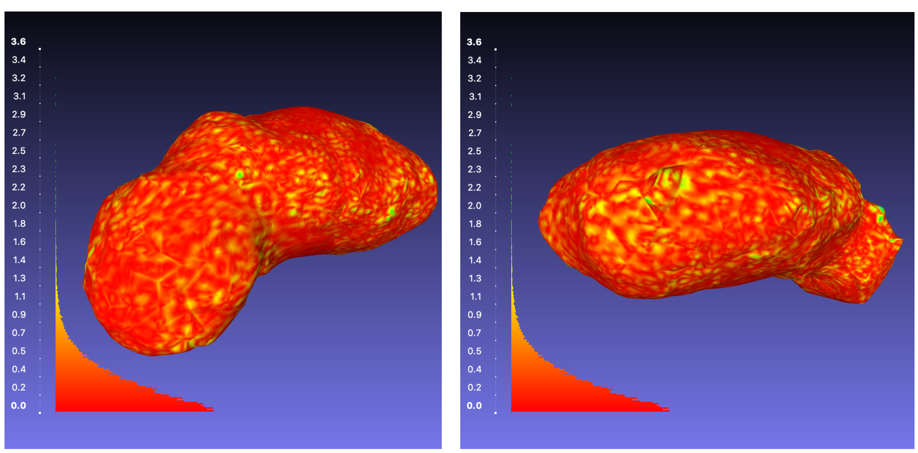
Figure 9.2: Normalized Hausdorff distance-colored point clouds and distribution histogram, Cases 4 to 6



(a) Case 1

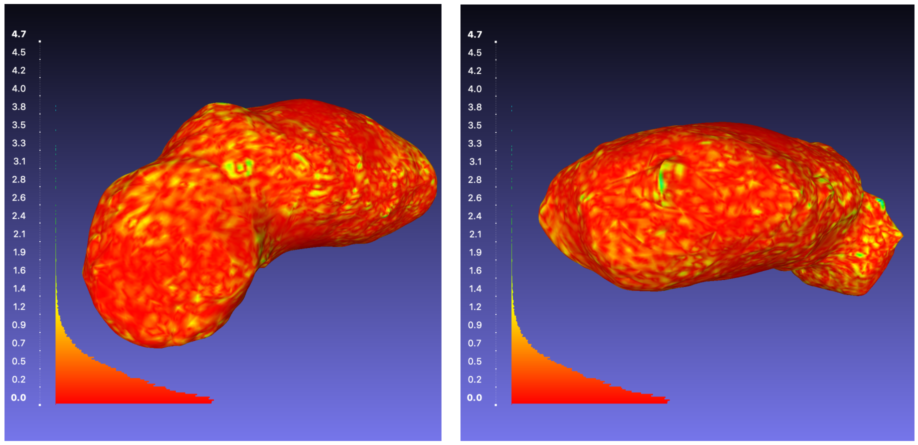


(b) Case 2

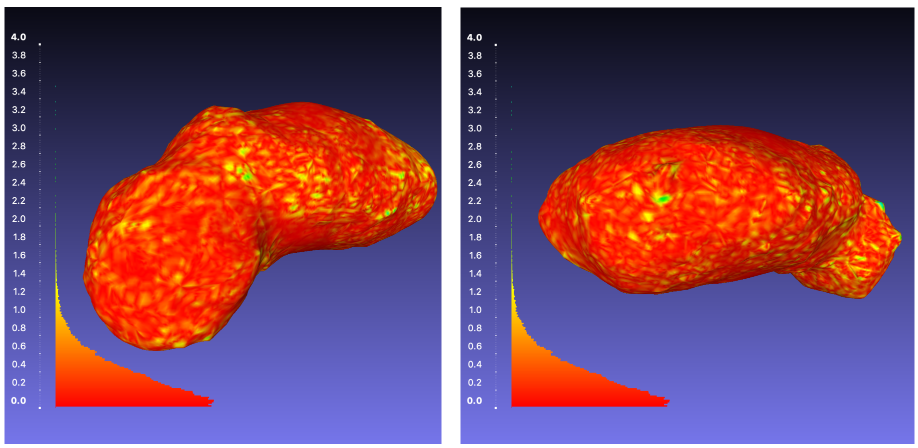


(c) Case 3

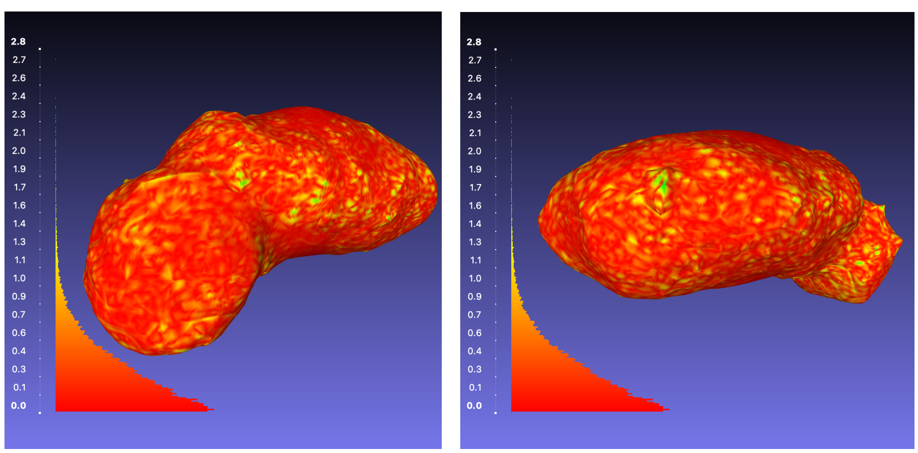
Figure 9.3: Normalized Hausdorff distance-colored fit shapes and distribution histogram, Cases 1 to 3



(a) Case 4



(b) Case 5



(c) Case 6

Figure 9.4: Normalized Hausdorff distance-colored fit shapes and distribution histogram, Cases 4 to 6

9.1.2 Sensitivity to non-principal spin, instrument frequency sensor noise

The robustness of the framework was further investigated by introducing an off-principal axis component to the spin axis of the asteroid, and by varying the standard deviation of the range masurement σ along with time elapsed between two observations T_{obs}. The bundle-adjustment hierarchical setting was kept to a constant value $h = 0$ throughout these simulations. The spin axis displacement from the maximum inertia axis was measured by the angle θ, detailed along with the rest of the input parameters on Table 9.7. No outstanding behavior appears to show in Table 9.8, as the sizes of the final point clouds exhibit no little variation. However, it is clear from Table 9.9 that the point-cloud-to-true-shape distances strongly differ based on the simulation inputs. The strongest sensitivity is clearly in the noise standard deviation, as the distance statistics in Cases 8 and 10 show. The off-axis spin does appear to have an effect, since the statistics in Case 7 are comparable to that in Case 1. It seems that increasing the instrument frequency is sufficient to compensate for the acceleration in the rotational dynamics, as seen in Case 9. These statistics can be complemented by the same side-by-side display as in the previous scenario. Figures 9.5 and 9.6 reproduce the results shown in Table 9.9 in more details : the lack of contrast in the color scheme is caused by the normalization of the histogram to a maximum value of 9 meters, the maximum distance value reported in Case 10. Because most of the points in the other cases exhibit much lower errors, the color scale appears compressed. Fit-shape-to-true-shape Hausdorff distances can be seen on Table 9.10, Figure 9.7 and 9.8. These results do not seem quite as affected by the outliers seen in Case 8 and Case 10 as those shown in Figures 9.5, 9.6 and Table 9.9. This is due to the fact that the Bezier shapes are obtained through the fitting of the global point clouds, which are comprised of millions of points, while the outliers are only a few hundred thousands.

Table 9.7: Input parameters in spin deviation, instrument frequency and sensor-noise, Cases 7 to 10

	Case 7	Case 8	Case 9	Case 10
θ (deg)	5	5	5	5
T_{obs} (s)	3333	3333	2500	2500
σ (m)	0.5	1	0.5	1

Table 9.8: Size of globally-covering point cloud, Cases 7 to 10

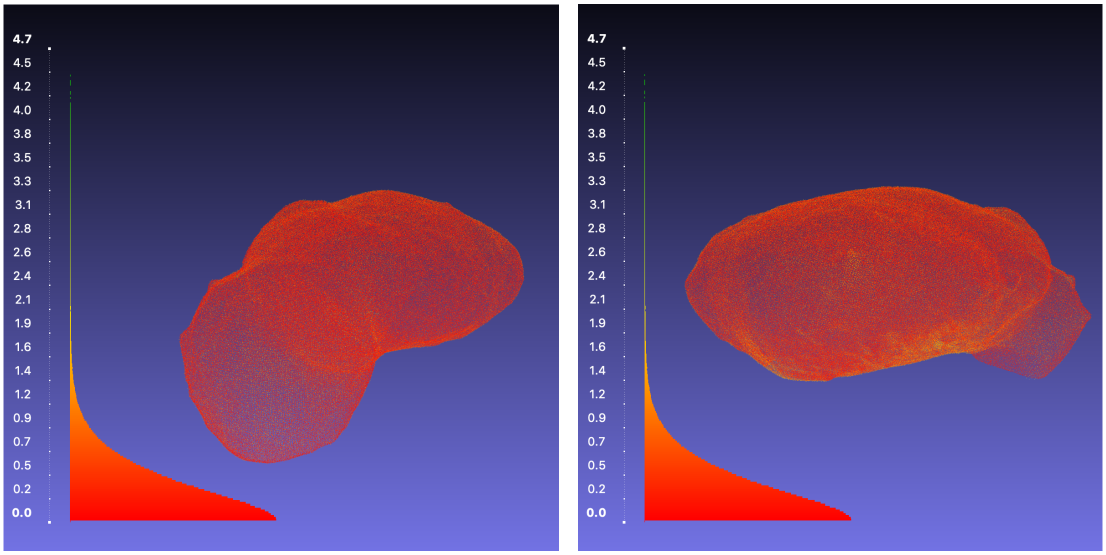
	Case 7	Case 8	Case 9	Case 10
Point-cloud size	2,359,034	2,347,105	3,164,355	3,164,355

Table 9.9: Point-cloud-to-true-shape Hausdorff distance statistics, Cases 7 to 10

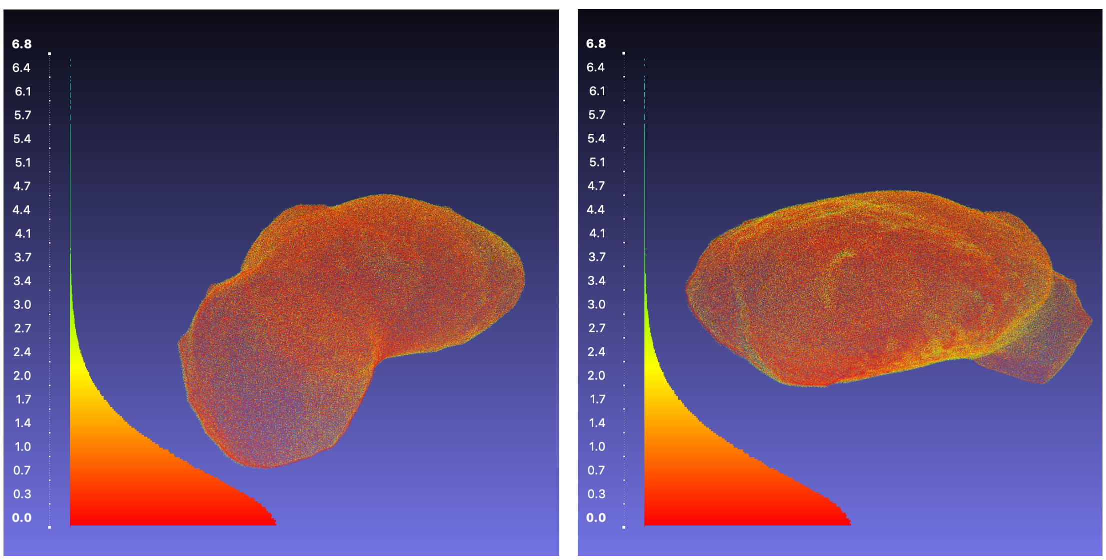
	Case 7	Case 8	Case 9	Case 10
Maximum (m)	4.714	6.755	2.961	8.995
Mean (m)	0.354	0.789	0.332	0.823
Standard deviation (m)	0.322	0.643	0.279	0.699

Table 9.10: Fit-shape-to-true-shape Hausdorff distance statistics for Cases 7 to 10

	Case 7	Case 8	Case 9	Case 10
Maximum (m)	4.071	3.27	3.57	3.97
Mean (m)	0.258	0.449	0.244	0.378
Standard deviation (m)	0.24	0.385	0.229	0.324

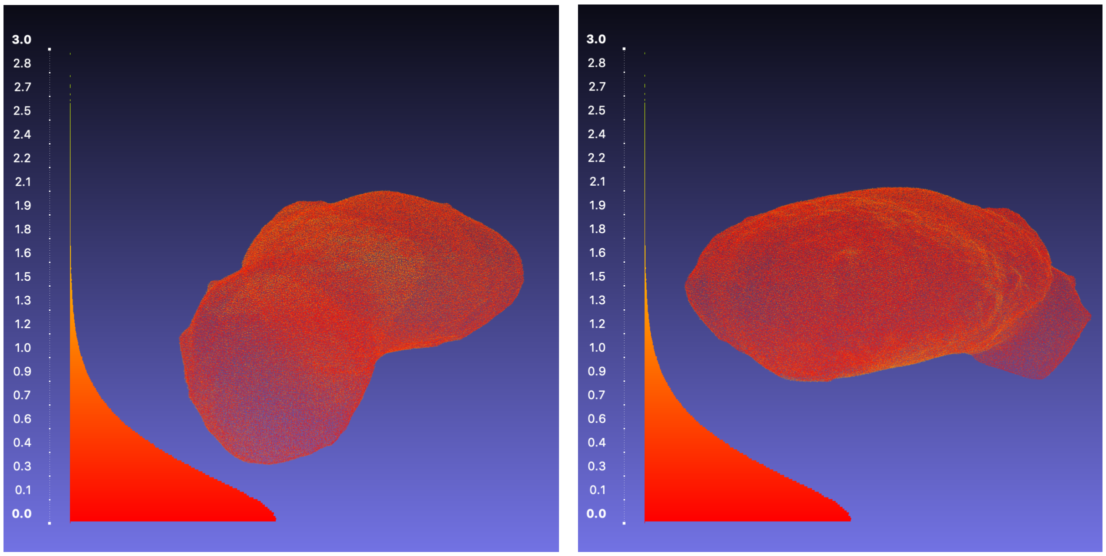


(a) Case 7

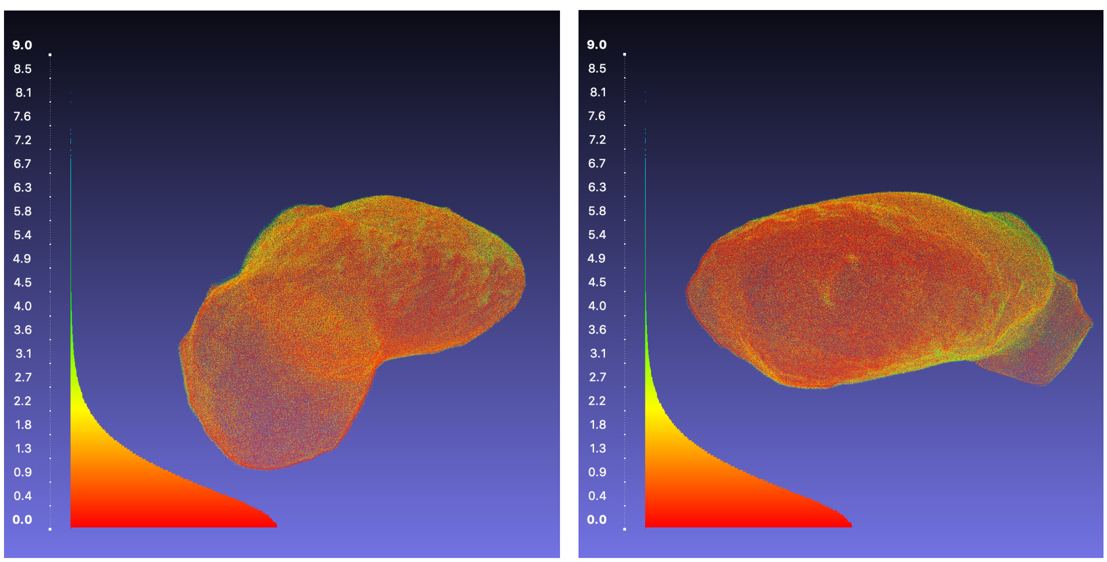


(b) Case 8

Figure 9.5: Normalized Hausdorff distance-colored point clouds and distribution histogram, Cases 7 and 8

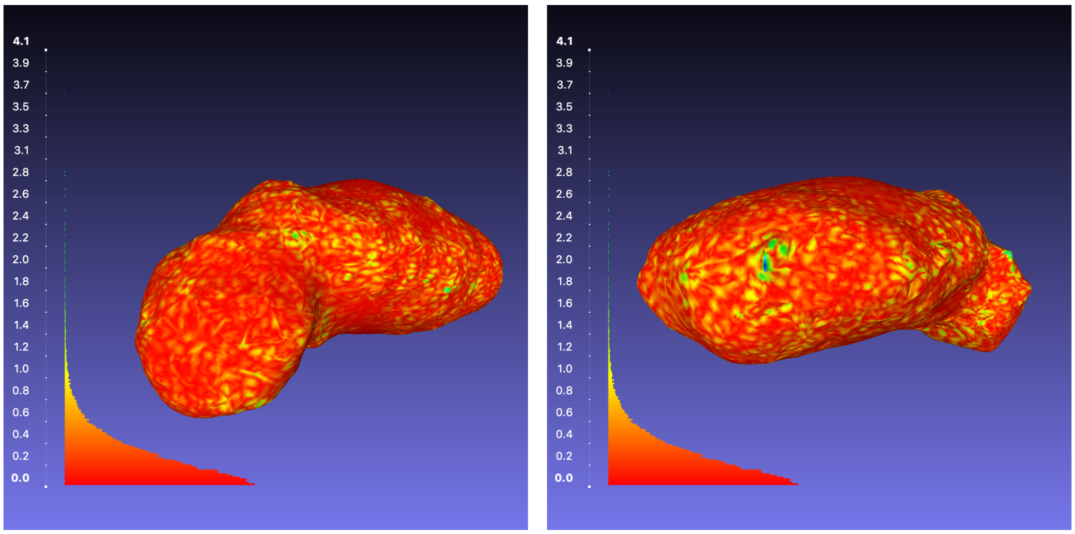


(a) Case 9

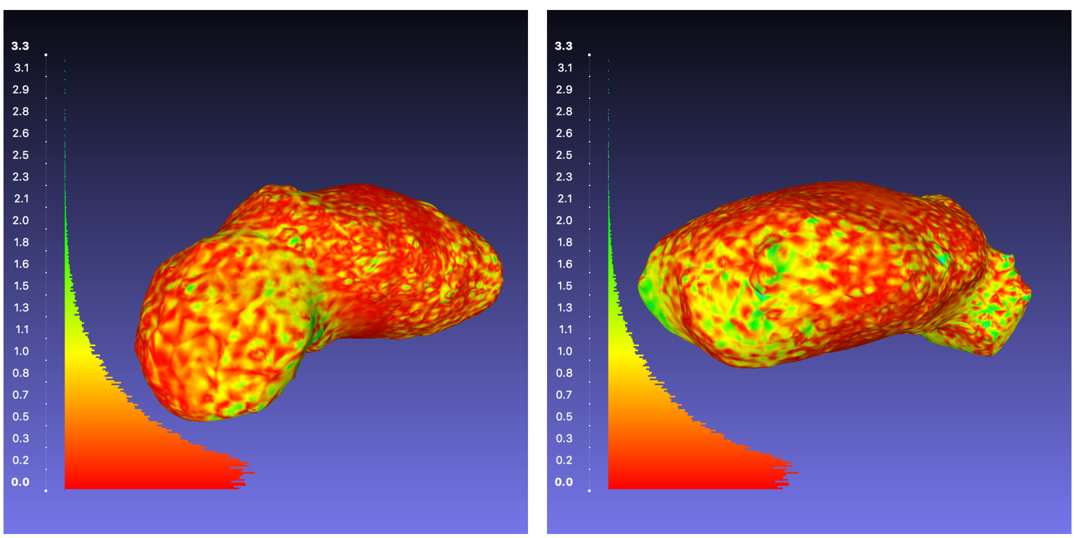


(b) Case 10

Figure 9.6: Normalized Hausdorff distance-colored point clouds and distribution histogram, Cases 9 and 10

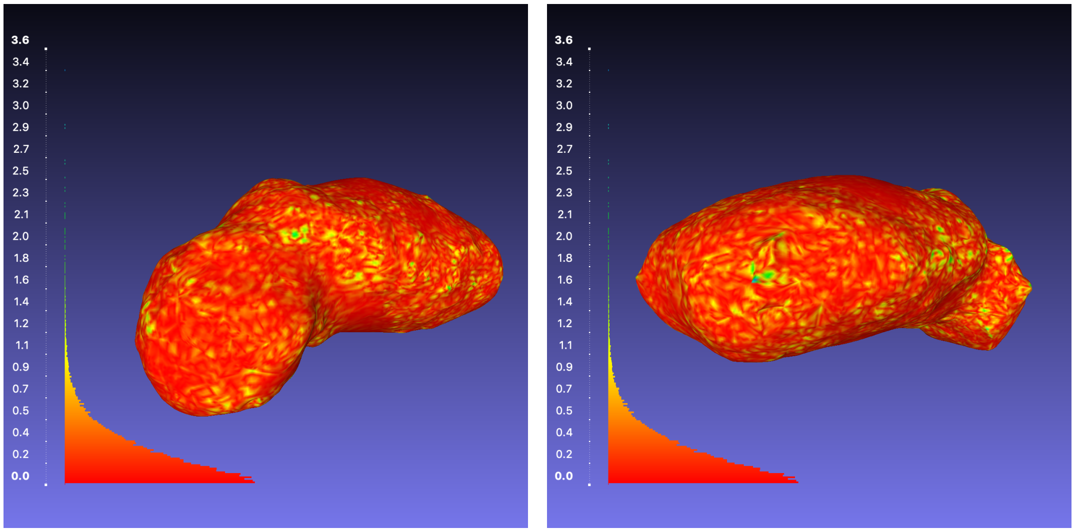


(a) Case 7

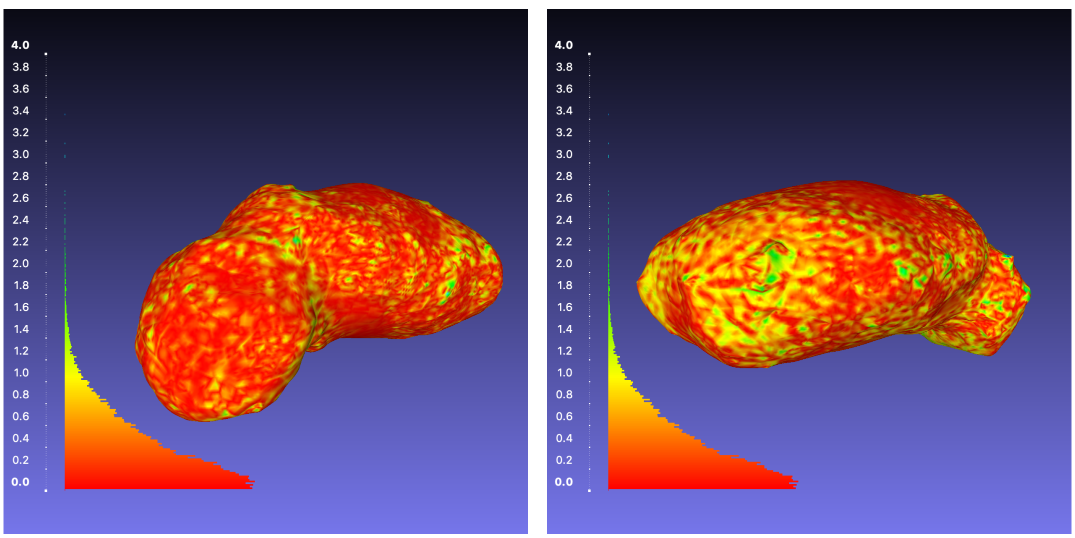


(b) Case 8

Figure 9.7: Normalized Hausdorff distance-colored fit shapes and distribution histogram, Cases 7 and 8



(a) Case 9



(b) Case 10

Figure 9.8: Normalized Hausdorff distance-colored fit shapes and distribution histogram, Cases 9 and 10

9.2 Model-based navigation

The estimated shapes, their uncertainty models as well as the estimate in the spacecraft and small body state obtained from the processing of the rigid transforms were provided to the augmented model-based navigation filter. The filter was kicked-off with the a-priori stemming from the processing of the successive rigid transforms and estimated center-of-mass location. The estimate of μ was simply taken as from the mean and standard deviations of the μ estimated in the successive IOD arcs. The uncertainty in C_r was set to 0.1, consistent with the difference between the true value (1.2) and a-priori value (set to 1.1). The deliverables from Cases 1, 4, 7, 8, 9 and 10 were combined with a variety of combinations of the Lidar line-of-sight sensor noise standard deviation σ_ρ, spacecraft acceleration and small-body angular velocity process noise standard deviations $\sigma_{\ddot{\mathbf{r}}}$ and $\sigma_{\dot{\boldsymbol{\omega}}}$ as well as instrument observation frequency $1/T_{\text{obs}}$. These combinations spanned a total of 96 simulations. The correspondance between each case and its designated set of inputs is compiled in Tables 9.12, 9.13, 9.14 and 9.15. The filtering performance of each cases is measured through the RMS of the state errors, summed-up over the N times comprising the 100-hour filtering arc:

$$\sigma_{\mathbf{r}} = \sqrt{\frac{\sum_{i=1}^{N} \|\delta\mathbf{r}(i) - \mathrm{E}\left(\delta\mathbf{r}\right)\|^2}{N-1}} \tag{9.2}$$

$$\sigma_{\dot{\mathbf{r}}} = \sqrt{\frac{\sum_{i=1}^{N} \|\delta\dot{\mathbf{r}}(i) - \mathrm{E}\left(\delta\dot{\mathbf{r}}\right)\|^2}{N-1}} \tag{9.3}$$

$$\sigma_{\Phi} = \sqrt{\frac{\sum_{i=1}^{N} \left[\delta\Phi(i) - \mathrm{E}\left(\delta\Phi\right)\right]^2}{N-1}} \tag{9.4}$$

$$\sigma_{\dot{\boldsymbol{\omega}}} = \sqrt{\frac{\sum_{i=1}^{N} \|\delta\dot{\boldsymbol{\omega}}(i) - \mathrm{E}\left(\delta\dot{\boldsymbol{\omega}}\right)\|^2}{N-1}} \tag{9.5}$$

$$\sigma_{\mu} = \sqrt{\frac{\sum_{i=1}^{N} \left[\delta\mu(i) - \mathrm{E}\left(\delta\mu\right)\right]^2}{N-1}} \tag{9.6}$$

$$\sigma_{C_r} = \sqrt{\frac{\sum_{i=1}^{N} \left[\delta C_r(i) - \mathrm{E}\left(\delta C_r\right)\right]^2}{N-1}} \tag{9.7}$$

Φ designates the magnitude of the principal rotation vector tracking the error between the true and estimated $[\mathcal{BN}]$ DCMs. The statistics in each of the RMS compiled over all cases is provided in Table 9.11. Some perspective in the best results can be obtained by comparing these to the navigation performance of past-flown asteroid missions. For instance, the JPL team in charge of navigating NEAR relative to Eros used a-priori uncertainties in the spacecraft velocity of 0.1 mm/s (one-sigma) [123]. This value is comparable to the best velocity RMS found in the parameter sweep. It must also be said that the values reported in Table 9.11 are not smoothed, and would thus be significantly lower had smoothing been applied to the navigation data, or had the RMS time window started later, once the different states have started to converge.

The RMS values for each case are listed in Tables 9.16 to 9.19. They are synthesized on Figure 9.11, which offers a synthetic view of the 96 navigation scenarii, where the color coding allows for immediate identification of how the RMS in the different state components varied across all cases. The normalization scheme applied to each RMS was such that the minimum RMS in a given state

Table 9.11: Mean, standard deviation and extremum values of the state RMS over all cases

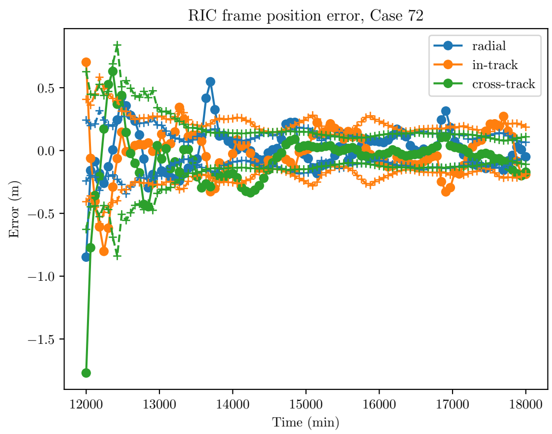
	Mean	Standard deviation	Minimum	Maximum
$\sigma_{\mathbf{r}}$ (m)	0.54	0.21	0.20	1.12
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.24	0.07	0.15	0.36
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.14	0.07	0.06	0.42
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	127.23	82.36	27.31	335.18
σ_{μ} (cm^3/s^2)	14813.83	8247.44	3036.49	30274.07
σ_{C_r}	0.37	0.27	0.03	0.72

was equated to 0, and the worst RMS to 1, using the extremum values found in Table 9.11. Figure 9.12 displays all the different cases, this time sorted by their mean over all 6 residuals. This allows for the immediate identification of the key trades in the Survey & Mapping and navigation phases input parameters. Finally, Figure 9.13 provides a different ordering, this time based on the input case from the Survey & Mapping phase that was provided to the navigation filter. For the sake of completeness, Figures 9.9 and 9.10 show the results from the cases exhibiting the best and worst performance of the filter over the navigation arc. The striking differences between the two cases lies in the quality of the velocity, standard gravitational and SRP coefficient estimates. An investigation of this behavior as well as the variation in the other RMS across all cases is carried out hereunder.

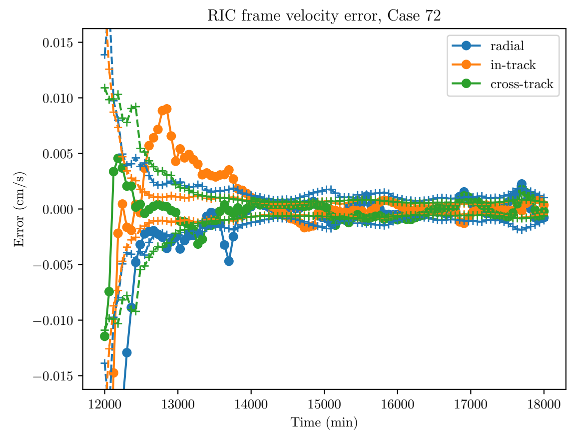
First of all, it appears that the RMS in the spacecraft velocity and C_r define the strongest boundary in the result space, roughly partitioning the simulations in two halves of equivalent case count. Figure 9.13 provides an unambiguous explanation, since it clearly demonstrates that the Input Cases 4, 10 and 9 systematically yielded good spacecraft velocity and C_r RMS. On the contrary, Input Cases 1, 7 and 8 systematically corresponded to poor C_r estimation performance, naturally affecting that of the spacecraft velocity. As seen on Table 9.2 and 9.7, Cases 1, 7 and 8 correspond to the smallest observation rate (one observation every 3333 seconds), whereas Cases 4, 10 and 9 were set to collect one point cloud every 2500 seconds. The consequence is not so much in the quality of the point cloud itself, since Table 9.5 show similar statistics for Cases 1 and 4, but in the initial velocity estimate at filter kick-off. The velocity estimate provided to the navigation filter originates from the processing of the successive rigid transforms assuming Keplerian dynamics. Less frequent observations leave more time for biases to creep in the dynamics of the observation arc, driving the velocity estimate outside of its confidence bounds. This intuition is confirmed by Figures 9.18 and 9.17 : the initial velocity estimate is way off in Navigation Case 76, causing the filter to spuriously update C_r, whereas the initial velocity estimate in Navigation Case 96 is nearly completely consistent with its covariance bounds.

The standard gravitational parameter does not seem to completely abide the same trends as C_r, although it must be noted that the maximum RMS on the standard gravitational parameter only amounts to $30,274.07\text{cm}^3/\text{s}^2$, 1.28% of the $2,360,000\text{cm}^3/\text{s}^2$ of Itokawa's standard gravitational parameter [110]. It appears that Navigation Case 88 featured excellent σ_μ RMSs, as opposed to the relative poor C_r results obtained in the same case. Conversely, Navigation Cases 92 and 22 feature comparatively good RMS values, with the exception of σ_μ that is much better in case 22. Navigation Case 22 was provided with the output from the Survey & Mapping Case 10, while Navigation Case 92 was provided with Survey & Mapping Case 9. Figure 9.19 shows that the difference in the two runs boils down to the larger a-priori uncertainty on the standard gravitational parameter in Case 92.

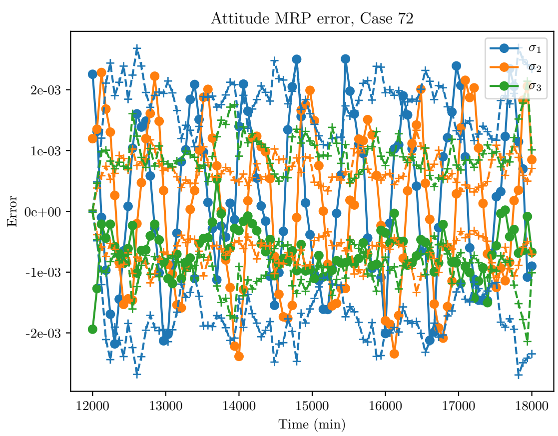
At this stage, it is sufficient to say that the driving parameter behind the performance of the navigation filter is the frequency at which observations were collected during the Survey & Mapping phase, as shown on Figure 9.14 that features a clear separation between all cases based on this criterion. The angular velocity is the last state component for which no obvious clustering has been found yet. It can actually be seen that the process noise on the angular acceleration is responsible for the variation in the performance, like shown by a side-to-side comparison of Figure 9.15 and 9.16.



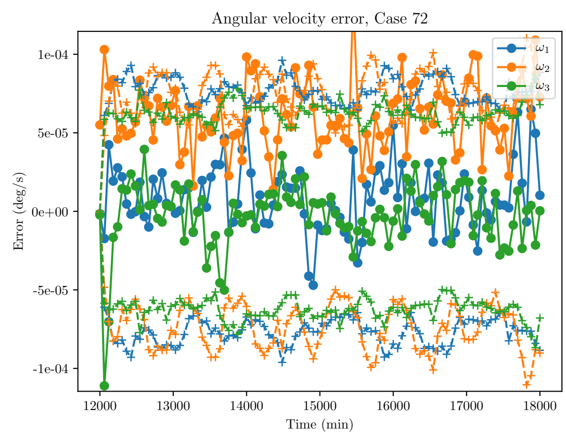
(a) RIC frame position errors



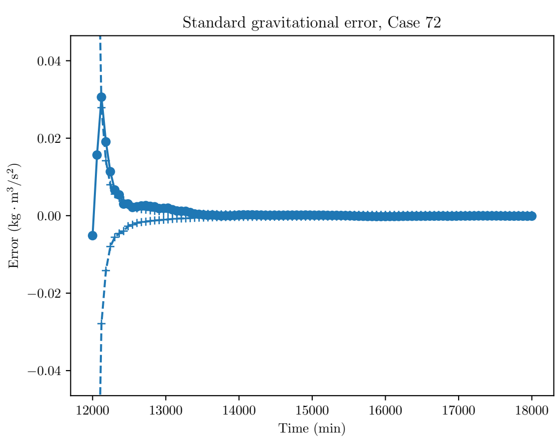
(b) RIC frame velocity errors



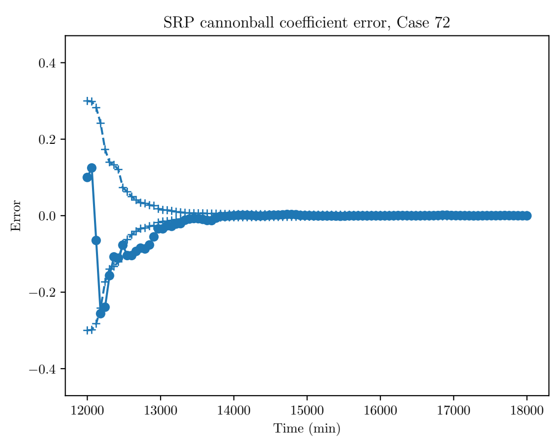
(c) Attitude MRP errors



(d) Angular velocity errors

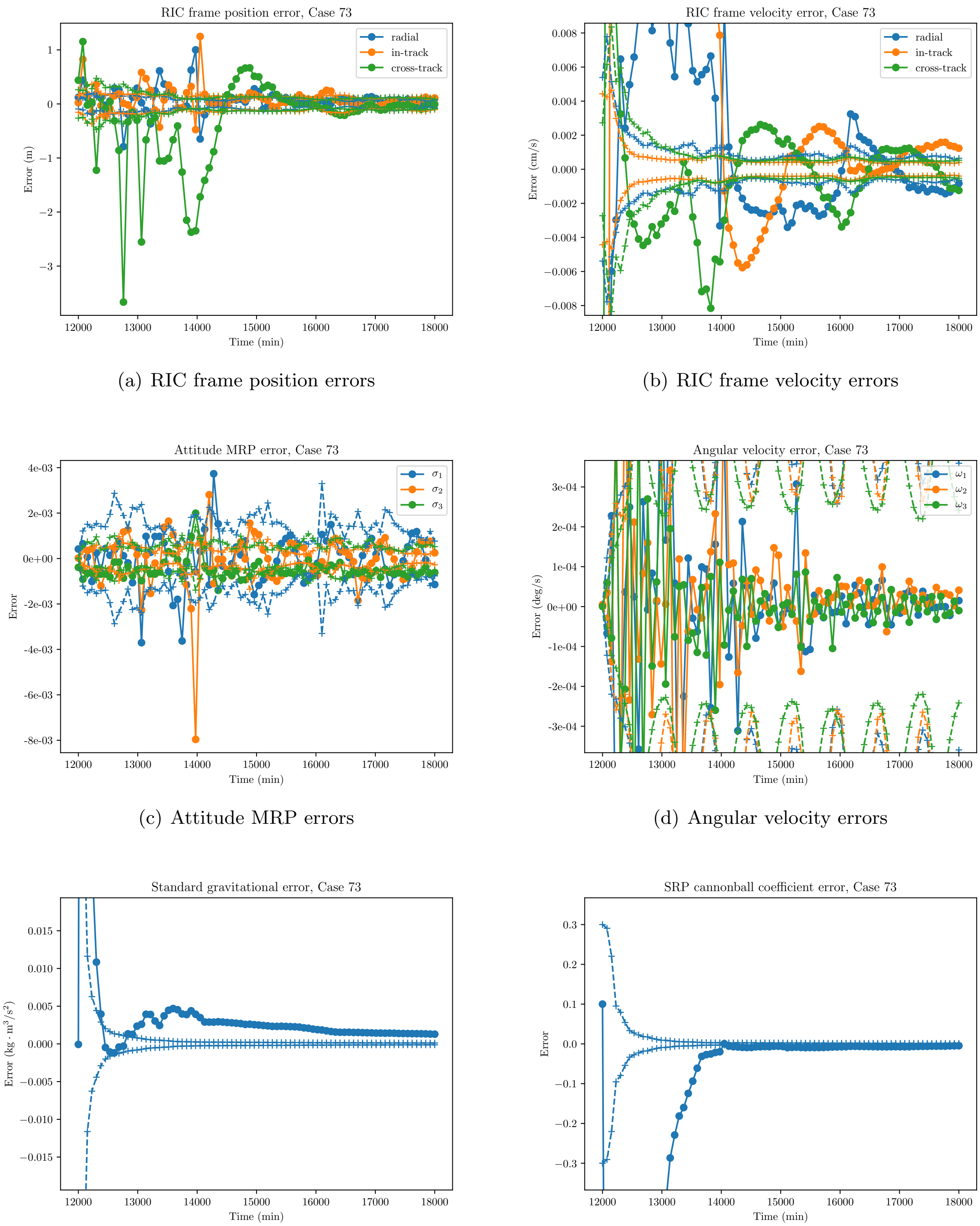


(e) Standard gravitational parameter errors



(f) SRP coefficient error

Figure 9.9: State errors and associated 3σ covariances, Case 72 (Best mean RMS)



(a) RIC frame position errors

(b) RIC frame velocity errors

(c) Attitude MRP errors

(d) Angular velocity errors

(e) Standard gravitational parameter errors

(f) SRP coefficient error

Figure 9.10: State errors and associated 3σ covariances, Case 73 (Worst mean RMS)

Table 9.12: Simulation inputs, Cases 1 to 30

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00
T_{obs} (s)	3600	3600	3600	3600	3600	4500	3600	3600	3600	3600
Input Case	1	1	1	1	4	1	4	4	7	7

	Case 11	Case 12	Case 13	Case 14	Case 15	Case 16	Case 17	Case 18	Case 19	Case 20
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10
T_{obs} (s)	3600	3600	3600	3600	3600	4500	3600	3600	3600	3600
Input Case	7	7	8	8	8	4	7	9	9	9

	Case 21	Case 22	Case 23	Case 24	Case 25	Case 26	Case 27	Case 28	Case 29	Case 30
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00
T_{obs} (s)	4500	3600	3600	3600	4500	4500	4500	4500	4500	4500
Input Case	7	10	10	10	1	1	1	1	4	4

Table 9.13: Simulation inputs, Cases 31 to 60

	Case 31	Case 32	Case 33	Case 34	Case 35	Case 36	Case 37	Case 38	Case 39	Case 40
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10
T_{obs} (s)	4500	4500	4500	4500	4500	4500	4500	4500	4500	4500
Input Case	4	4	7	7	7	7	8	8	8	8

	Case 41	Case 42	Case 43	Case 44	Case 45	Case 46	Case 47	Case 48	Case 49	Case 50
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.10	0.10
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00
T_{obs} (s)	4500	4500	4500	4500	4500	4500	4500	4500	3600	3600
Input Case	9	9	9	9	10	10	10	10	1	1

	Case 51	Case 52	Case 53	Case 54	Case 55	Case 56	Case 57	Case 58	Case 59	Case 60
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10
T_{obs} (s)	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600
Input Case	1	1	4	4	4	4	7	7	7	7

Table 9.14: Simulation inputs, Cases 61 to 90

	Case 61	Case 62	Case 63	Case 64	Case 65	Case 66	Case 67	Case 68	Case 69	Case 70
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00
T_{obs} (s)	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600
Input Case	8	8	8	8	9	9	9	9	10	10
	Case 71	Case 72	Case 73	Case 74	Case 75	Case 76	Case 77	Case 78	Case 79	Case 80
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10
T_{obs} (s)	3600	3600	4500	4500	4500	4500	4500	4500	4500	4500
Input Case	10	10	1	1	1	1	4	4	4	4
	Case 81	Case 82	Case 83	Case 84	Case 85	Case 86	Case 87	Case 88	Case 89	Case 90
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	1.00	1.00	0.10	0.10	1.00	1.00	0.10	0.10	1.00	1.00
T_{obs} (s)	4500	4500	4500	4500	4500	4500	4500	4500	4500	4500
Input Case	7	7	7	7	8	8	8	8	9	9

Table 9.15: Simulation inputs, Cases 91 to 96

	Case 91	Case 92	Case 93	Case 94	Case 95	Case 96
σ_ρ (m)	0.50	1.00	0.50	1.00	0.50	1.00
$\sigma_{\ddot{\mathbf{r}}}$ (nm/s^2)	0.10	0.10	0.10	0.10	0.10	0.10
$\sigma_{\dot{\boldsymbol{\omega}}}$ (nrad/s^2)	0.10	0.10	1.00	1.00	0.10	0.10
T_{obs} (s)	4500	4500	4500	4500	4500	4500
Input Case	9	9	10	10	10	10

Table 9.16: State error standard deviations, Cases 1 to 30

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10
$\sigma_{\mathbf{r}}$ (m)	0.548	0.492	0.567	0.536	0.278	0.470	0.201	0.208	0.432	0.527
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.282	0.292	0.279	0.279	0.165	0.292	0.160	0.164	0.275	0.289
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.144	0.108	0.115	0.097	0.074	0.091	0.060	0.079	0.111	0.096
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	174.193	180.860	60.959	54.500	138.484	237.668	27.315	33.166	164.804	146.198
σ_{μ} (cm^3/s^2)	28584.235	27373.824	27702.183	27437.148	20089.607	27784.285	20145.142	19606.211	11207.447	8792.965
σ_{C_r}	0.614	0.615	0.643	0.614	0.045	0.628	0.050	0.052	0.638	0.651

	Case 11	Case 12	Case 13	Case 14	Case 15	Case 16	Case 17	Case 18	Case 19	Case 20
$\sigma_{\mathbf{r}}$ (m)	0.413	0.451	0.718	0.753	0.770	0.273	0.432	0.451	0.472	0.458
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.270	0.277	0.318	0.318	0.314	0.177	0.275	0.164	0.170	0.168
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.102	0.101	0.136	0.165	0.136	0.081	0.111	0.084	0.104	0.090
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	55.517	54.747	164.718	165.331	64.439	36.739	164.804	172.868	61.069	57.141
σ_{μ} (cm^3/s^2)	11381.118	10756.151	9849.493	6695.053	7149.038	19022.253	11207.447	21030.529	16745.239	21111.957
σ_{C_r}	0.640	0.636	0.520	0.523	0.508	0.046	0.638	0.209	0.205	0.207

	Case 21	Case 22	Case 23	Case 24	Case 25	Case 26	Case 27	Case 28	Case 29	Case 30
$\sigma_{\mathbf{r}}$ (m)	0.320	0.440	0.432	0.421	0.485	0.470	0.532	0.536	0.374	0.359
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.265	0.158	0.152	0.152	0.289	0.292	0.280	0.289	0.184	0.181
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.097	0.108	0.099	0.109	0.084	0.091	0.085	0.095	0.220	0.103
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	167.980	85.252	35.514	38.142	212.747	237.668	48.443	54.308	219.339	169.798
σ_{μ} (cm^3/s^2)	14835.430	3036.490	4779.093	4119.614	28418.529	27784.285	30274.067	27353.314	17006.536	18851.050
σ_{C_r}	0.673	0.046	0.048	0.050	0.651	0.628	0.653	0.627	0.064	0.064

Table 9.17: State error standard deviations, Cases 31 to 60

	Case 31	Case 32	Case 33	Case 34	Case 35	Case 36	Case 37	Case 38	Case 39	Case 40
$\sigma_{\mathbf{r}}$ (m)	0.337	0.273	0.320	0.481	0.328	0.313	0.755	0.931	0.766	0.741
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.178	0.177	0.265	0.301	0.266	0.268	0.330	0.341	0.316	0.327
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.084	0.081	0.097	0.149	0.096	0.080	0.140	0.227	0.124	0.166
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	40.851	36.739	167.980	235.930	60.298	57.343	207.983	200.366	66.720	79.724
σ_{μ} (cm^3/s^2)	19661.448	19022.253	14835.430	9531.850	13880.966	13670.705	7568.183	6093.456	5602.472	4611.941
σ_{C_r}	0.027	0.046	0.673	0.682	0.662	0.650	0.512	0.513	0.489	0.501

	Case 41	Case 42	Case 43	Case 44	Case 45	Case 46	Case 47	Case 48	Case 49	Case 50
$\sigma_{\mathbf{r}}$ (m)	0.518	0.509	0.522	0.507	0.449	0.442	0.443	0.444	0.853	0.791
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.170	0.170	0.166	0.175	0.162	0.163	0.161	0.164	0.323	0.330
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.085	0.081	0.089	0.084	0.093	0.099	0.099	0.103	0.220	0.146
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	294.663	181.297	69.378	53.666	113.981	129.127	37.369	37.333	265.297	227.976
σ_{μ} (cm^3/s^2)	13850.721	18075.066	13286.445	18394.316	3537.155	4315.665	3958.060	4742.865	28169.812	27066.963
σ_{C_r}	0.202	0.203	0.178	0.187	0.045	0.042	0.050	0.044	0.636	0.643

	Case 51	Case 52	Case 53	Case 54	Case 55	Case 56	Case 57	Case 58	Case 59	Case 60
$\sigma_{\mathbf{r}}$ (m)	0.748	0.845	0.276	0.265	0.271	0.290	0.616	0.728	0.604	0.679
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.314	0.317	0.166	0.169	0.165	0.168	0.294	0.319	0.292	0.299
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.130	0.187	0.071	0.077	0.066	0.079	0.244	0.175	0.171	0.232
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	68.240	71.982	140.804	159.407	27.572	34.522	207.439	206.412	72.913	85.643
σ_{μ} (cm^3/s^2)	27514.598	27225.588	20108.857	19438.893	20228.103	19758.522	10943.379	8641.526	11189.305	10586.309
σ_{C_r}	0.676	0.646	0.041	0.053	0.040	0.040	0.673	0.683	0.664	0.657

Table 9.18: State error standard deviations, Cases 61 to 90

	Case 61	Case 62	Case 63	Case 64	Case 65	Case 66	Case 67	Case 68	Case 69	Case 70
$\sigma_{\mathbf{r}}$ (m)	0.921	0.909	0.848	0.842	0.464	0.462	0.479	0.460	0.376	0.374
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.327	0.334	0.325	0.322	0.164	0.167	0.171	0.170	0.155	0.158
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.231	0.297	0.220	0.196	0.087	0.107	0.101	0.108	0.111	0.124
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	195.659	219.459	81.462	77.650	253.528	191.651	62.011	58.620	83.957	94.275
σ_{μ} (cm^3/s^2)	9858.983	6335.437	7118.772	4462.284	16807.964	21072.018	16763.498	21133.718	4860.901	3091.747
σ_{C_r}	0.534	0.532	0.524	0.526	0.214	0.219	0.210	0.213	0.047	0.044

	Case 71	Case 72	Case 73	Case 74	Case 75	Case 76	Case 77	Case 78	Case 79	Case 80
$\sigma_{\mathbf{r}}$ (m)	0.368	0.358	0.853	0.873	0.792	0.918	0.461	0.251	0.339	0.339
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.151	0.150	0.333	0.337	0.317	0.330	0.187	0.177	0.178	0.181
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.109	0.111	0.237	0.188	0.228	0.240	0.127	0.087	0.085	0.092
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	33.220	36.932	335.183	303.283	82.799	82.905	185.363	209.794	33.642	40.126
σ_{μ} (cm^3/s^2)	4793.805	4161.274	28116.713	27475.234	30125.342	27146.987	20104.500	18286.872	19417.558	19928.000
σ_{C_r}	0.048	0.049	0.687	0.662	0.692	0.656	0.047	0.067	0.038	0.035

	Case 81	Case 82	Case 83	Case 84	Case 85	Case 86	Case 87	Case 88	Case 89	Case 90
$\sigma_{\mathbf{r}}$ (m)	0.731	0.818	0.666	0.710	0.967	1.123	0.907	0.917	0.516	0.507
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.297	0.332	0.293	0.301	0.356	0.357	0.332	0.340	0.171	0.171
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.417	0.258	0.312	0.359	0.224	0.362	0.176	0.267	0.091	0.105
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	267.493	264.122	96.653	106.216	283.388	288.234	89.610	101.608	295.365	183.488
σ_{μ} (cm^3/s^2)	14638.648	9376.921	13690.342	13468.839	7661.193	6073.851	5594.428	4501.194	13896.274	18086.676
σ_{C_r}	0.704	0.717	0.697	0.682	0.532	0.527	0.509	0.519	0.207	0.210

Table 9.19: State error standard deviations, Cases 91 to 96

	Case 91	Case 92	Case 93	Case 94	Case 95	Case 96
$\sigma_{\mathbf{r}}$ (m)	0.530	0.521	0.380	0.384	0.378	0.392
$\sigma_{\dot{\mathbf{r}}}$ (mm/s)	0.167	0.176	0.161	0.163	0.160	0.163
$\sigma_{\boldsymbol{\sigma}}$ (deg)	0.084	0.092	0.118	0.116	0.111	0.115
$\sigma_{\boldsymbol{\omega}}$ (μdeg/s)	70.668	56.029	116.773	131.055	39.204	39.073
σ_{μ} (cm^3/s^2)	13296.246	18405.889	3591.420	4372.749	3987.701	4783.589
σ_{C_r}	0.182	0.192	0.045	0.042	0.050	0.045

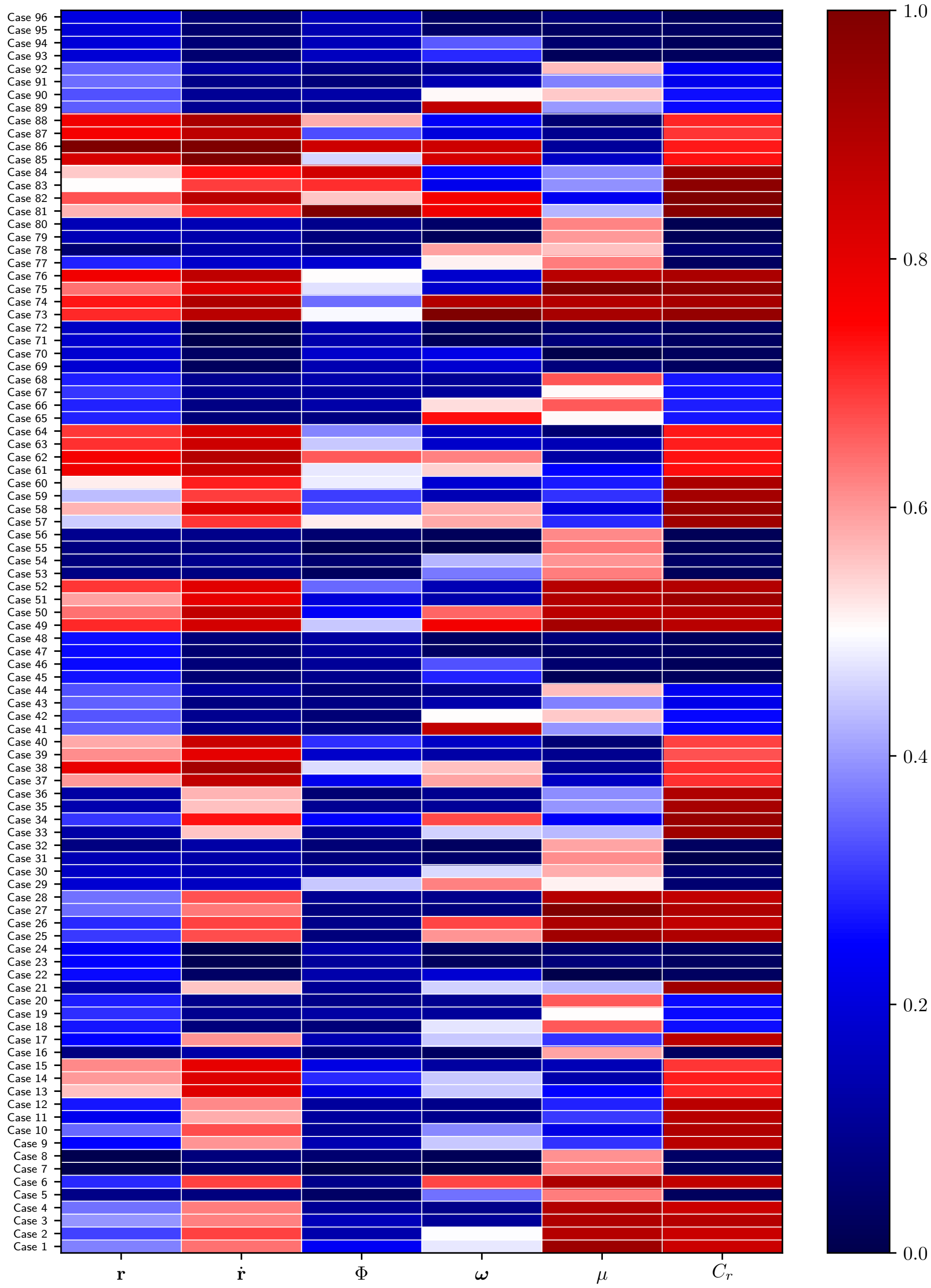


Figure 9.11: Normalized simulation RMS (State Partition vs Navigation Case). The minimum and maximum values in each state partition RMS were linearly rescaled to 0 and 1 respectively

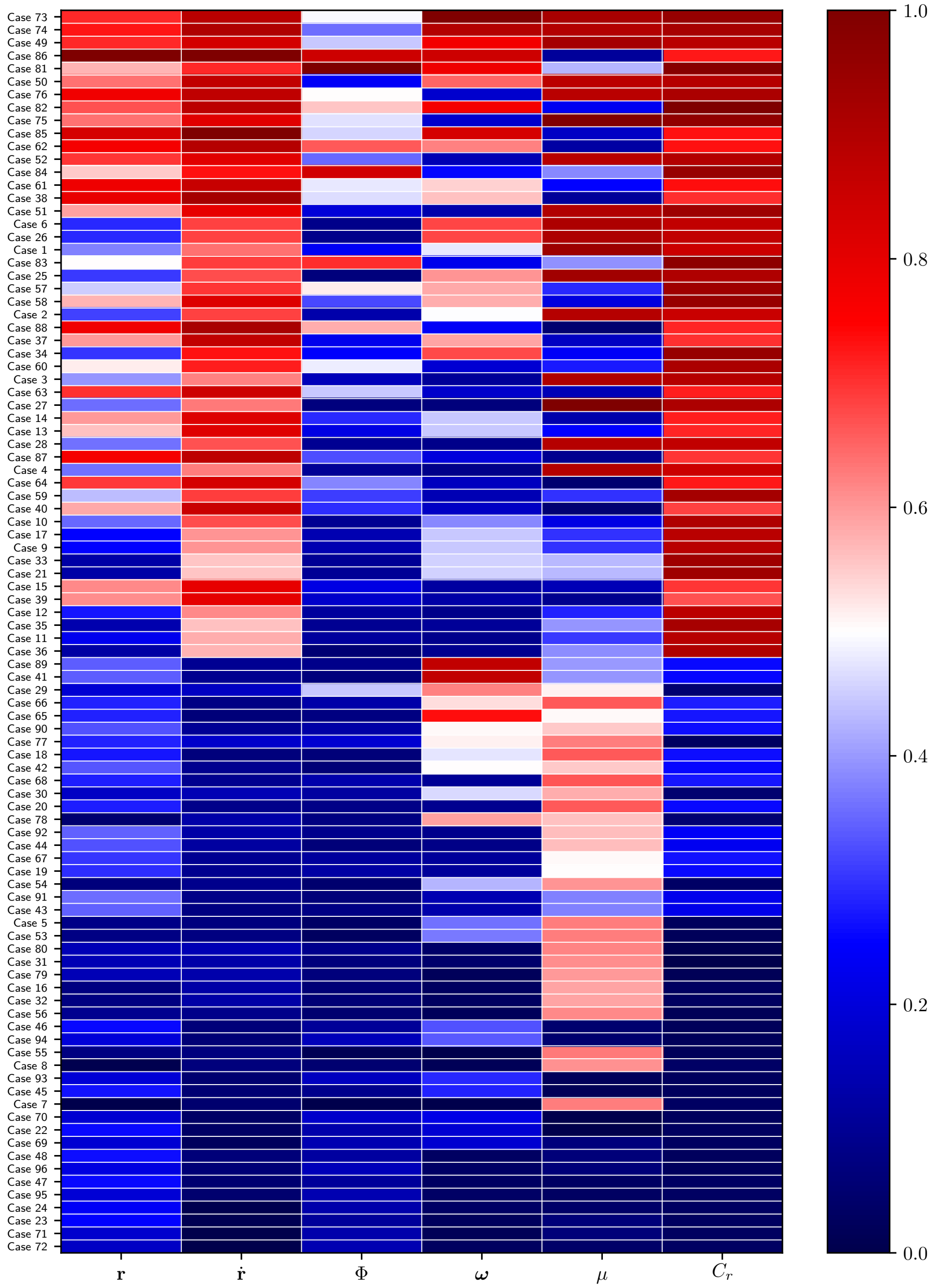


Figure 9.12: Normalized simulation RMS (State Partition vs Navigation Case), sorted from worst mean RMS (top) to best (bottom). The minimum and maximum values in each state partition RMS were linearly rescaled to 0 and 1 respectively

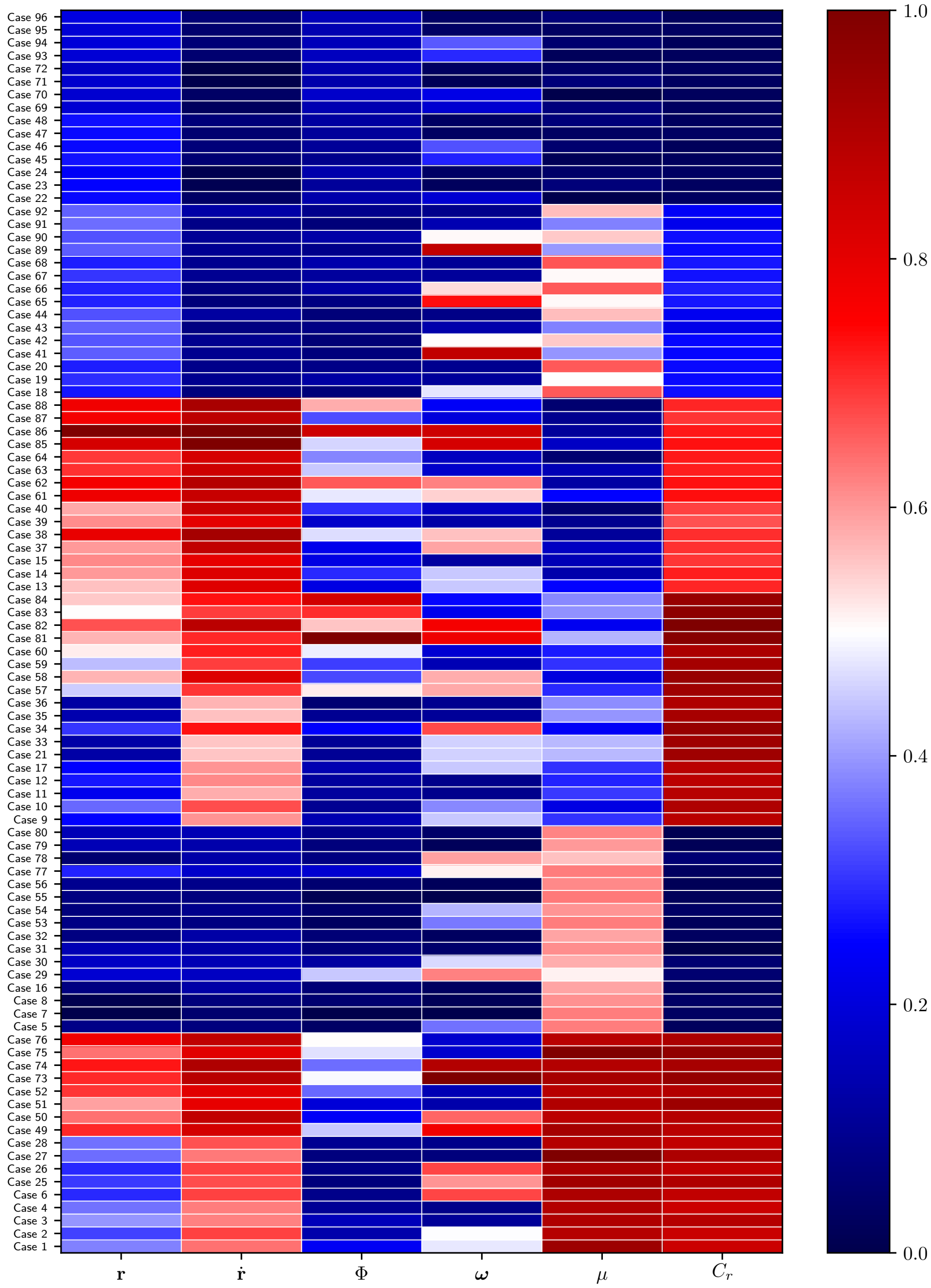


Figure 9.13: Normalized simulation RMS (State Partition vs Navigation Case), sorted by input case. The minimum and maximum values in each state partition RMS were linearly rescaled to 0 and 1 respectively

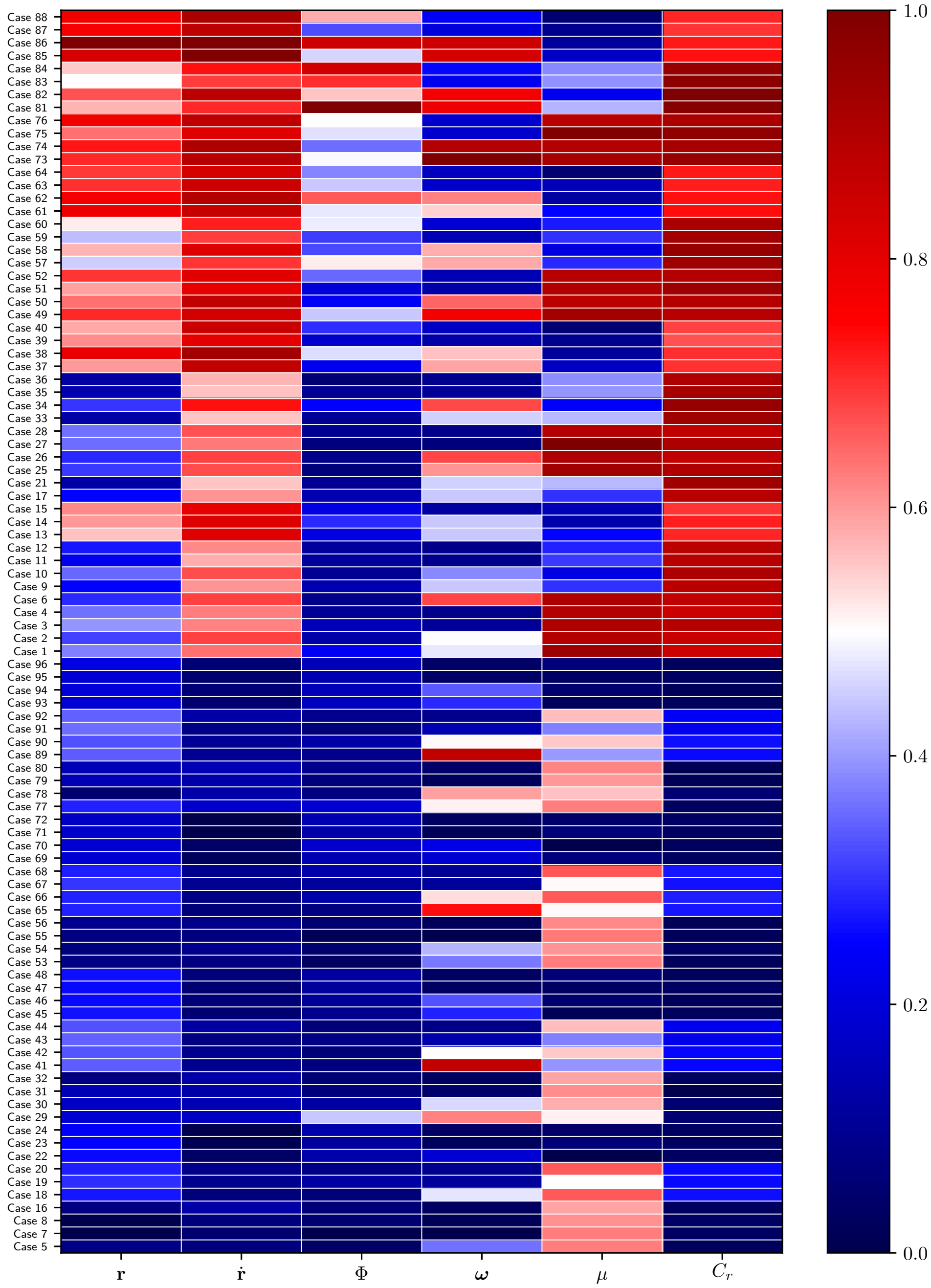


Figure 9.14: Normalized simulation RMS (State Partition vs Navigation Case), sorted by T_{obs} used in the input Survey & Mapping case. The minimum and maximum values in each state partition RMS were linearly rescaled to 0 and 1 respectively

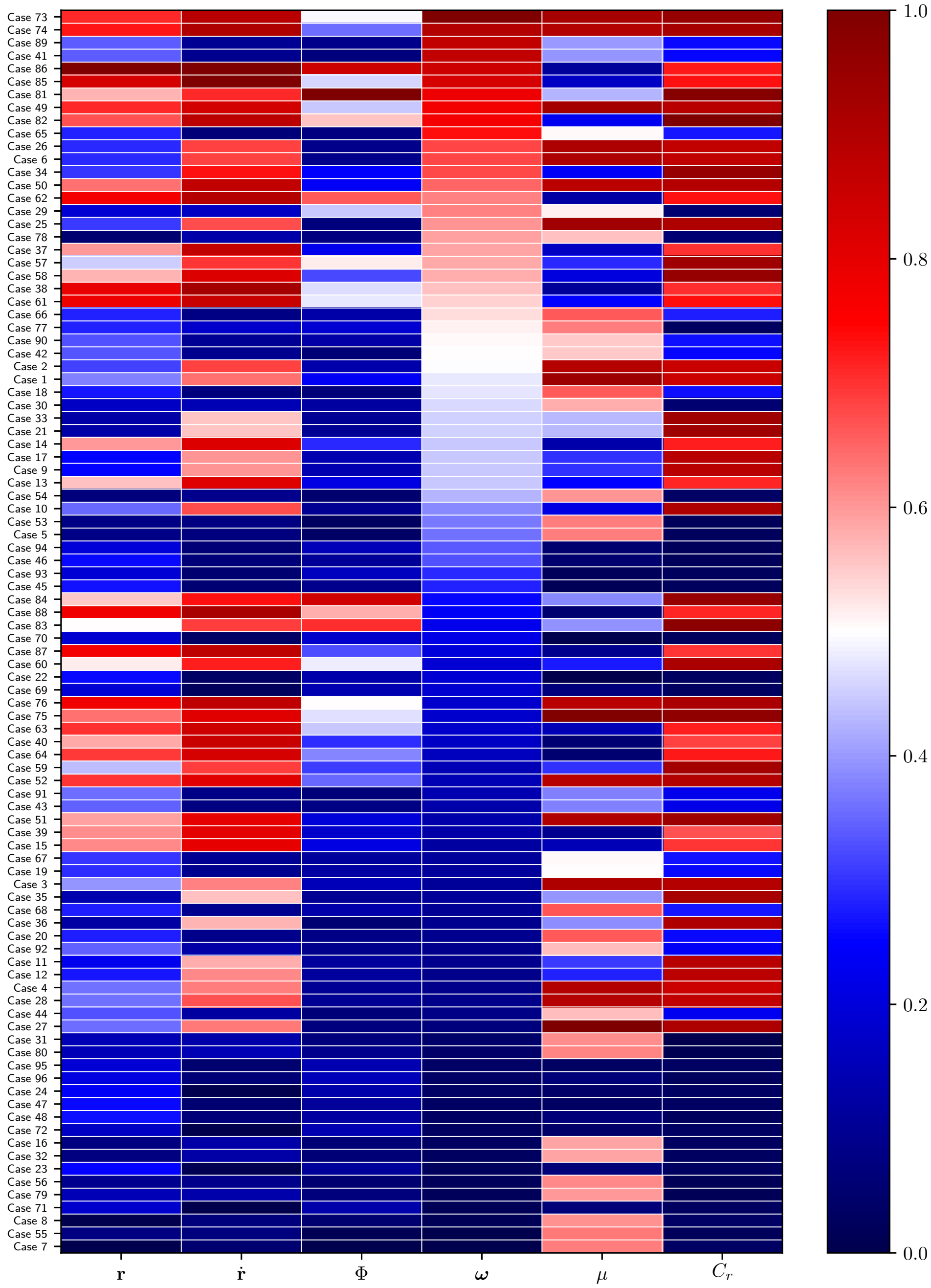


Figure 9.15: Normalized simulation RMS (State Partition vs Navigation Case), sorted by $\sigma_{\dot{\omega}}$ RMS. The minimum and maximum values in each state partition RMS were linearly rescaled to 0 and 1 respectively

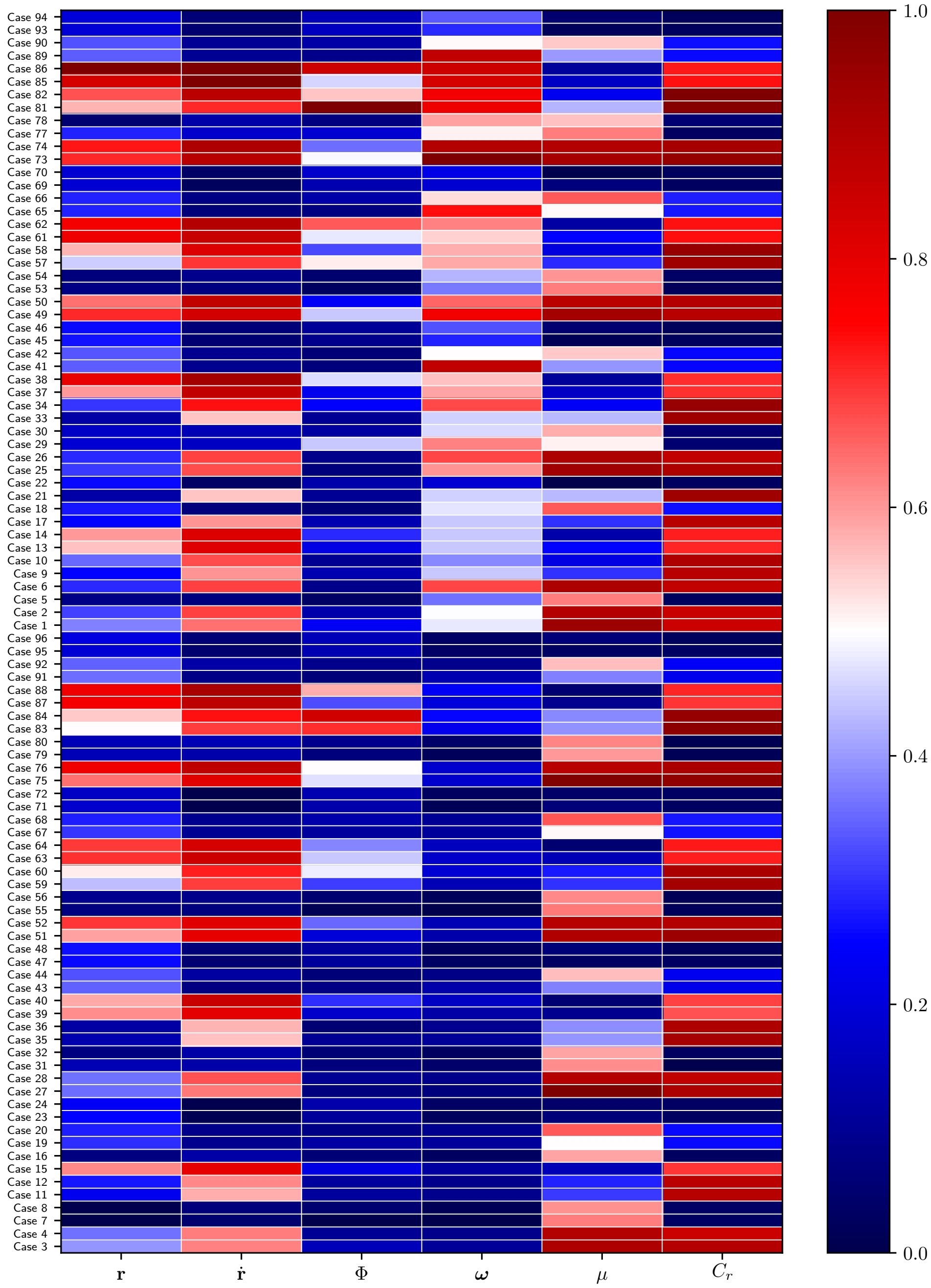


Figure 9.16: Normalized simulation RMS (State Partition vs Navigation Case), sorted by process noise on angular velocity. The minimum and maximum values in each state partition RMS were linearly rescaled to 0 and 1 respectively

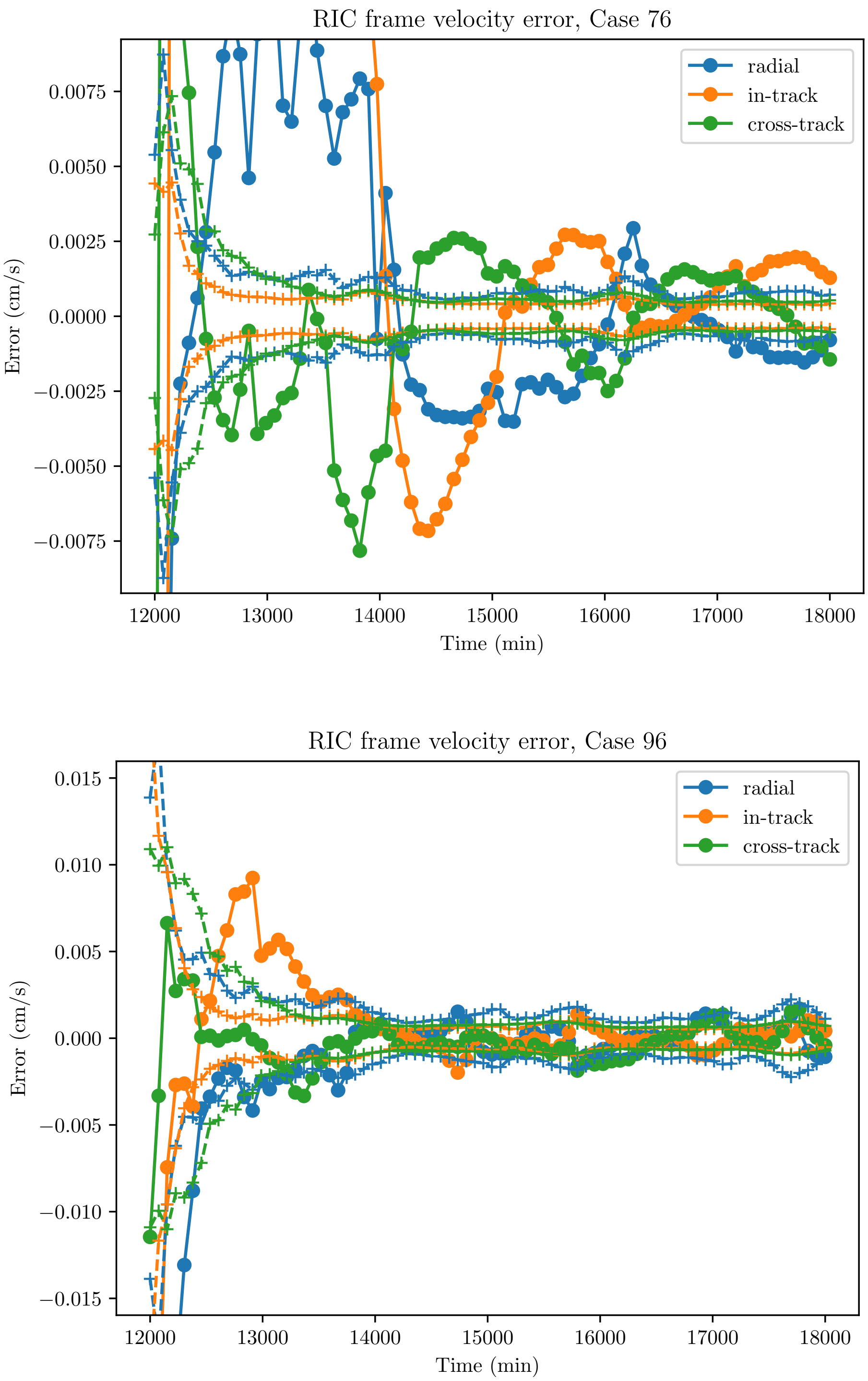
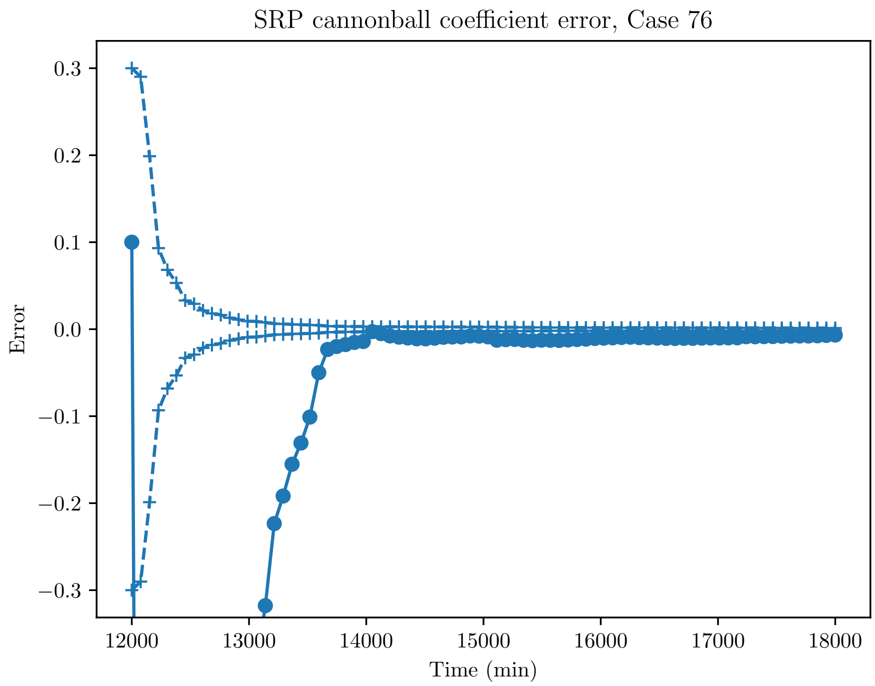


Figure 9.17: Velocity errors and 3-σ covariances. Top: Navigation Case 76. Bottom: Navigation Case 96



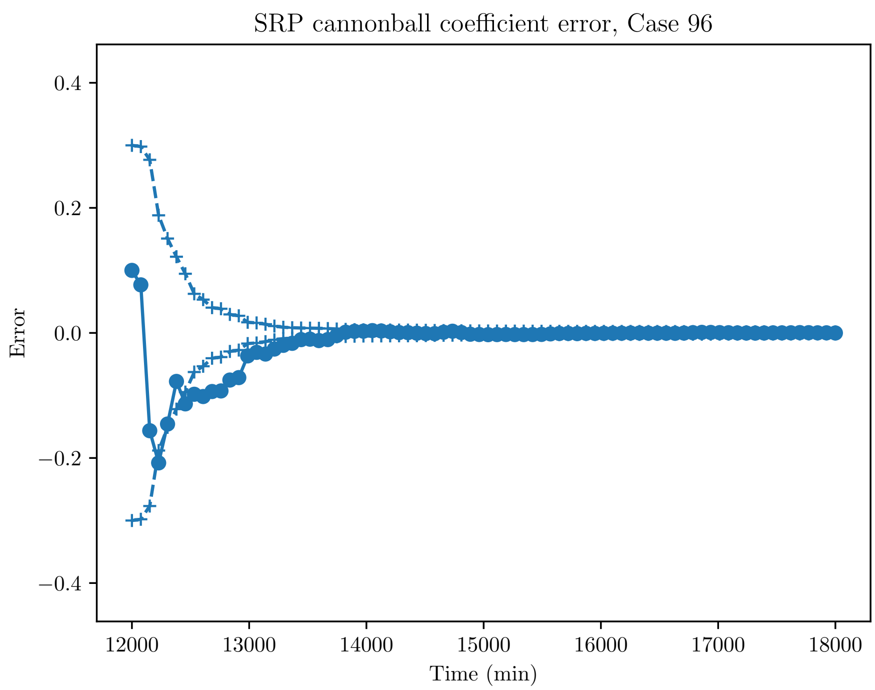
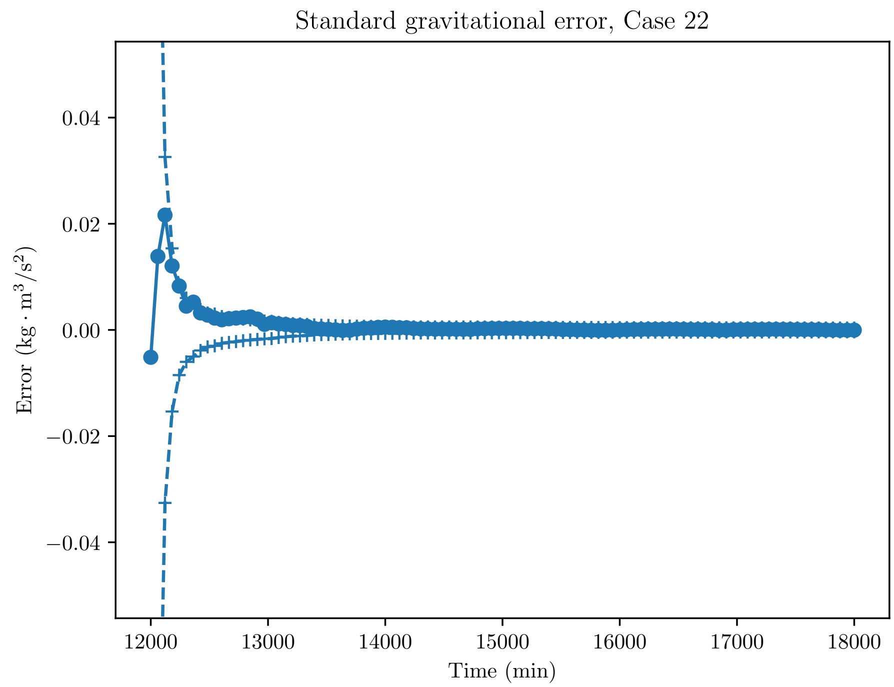


Figure 9.18: C_r errors and 3-σ covariances. Top: Navigation Case 76. Bottom: Navigation Case 96



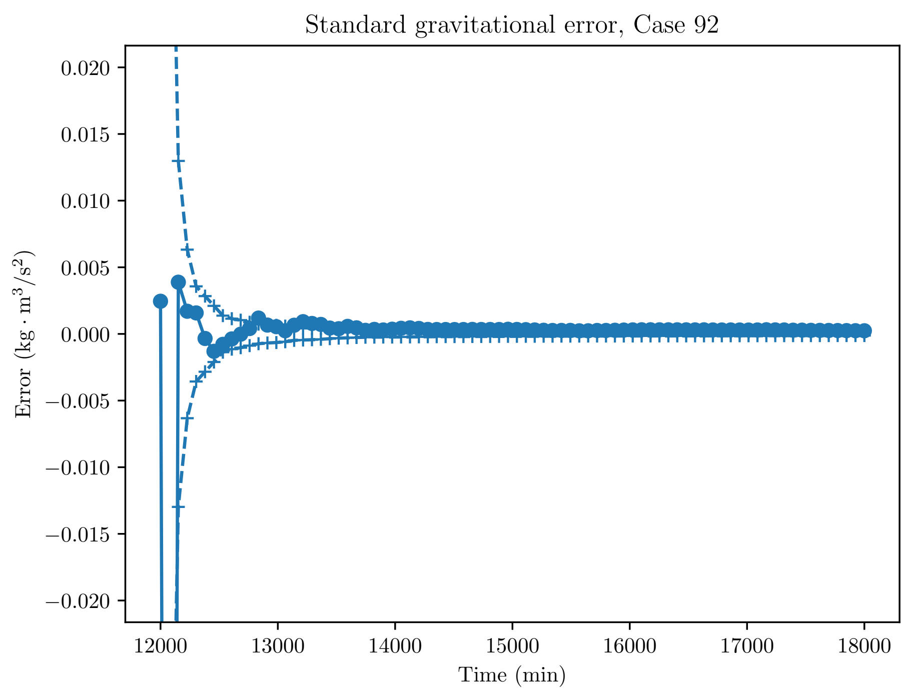


Figure 9.19: μ errors and 3-σ covariances. Top: Navigation Case 22. Bottom: Navigation Case 92

Chapter 10

Future work

This chapter goes over possible extensions of the research presented in this thesis that would help improving upon the current framework's runtime, robustness and performance, as well as providing more insight into the uncertainties in the gravity field of an uncertainty small body shape.

10.1 Uncertainty quantification about an unknown small body

10.1.1 Uncertainty in shape from different measurement type

The Lidar-based shape uncertainty model proposed in chapter 5 allows to recover the surface element covariances best explaining the shape fitting residuals in each element. This maximum-likelihood approach should be extended to other measurement types (namely, lightcurves and radar observations) so as to make the methods proposed in this thesis trully comprehensive, to the benefit of radar and optical astronomers.

10.1.2 Uncertainties in gravity arising from unknown, non-uniform density distribution

The polyhedron gravity model from which an analytical uncertainty model was extracted was based on the assumption that the small body density was a constant, uniform, known value. Allowing uncertainty in the uniform density is actually trivial, as pointed out in Chapter 6, since doing so leaves the shape uncertainty and the density uncertainty completely uncorrelated. This is no longer the case if the density becomes non-uniform. The polyhedron gravity model has already been extended to bodies with linearly varying density profiles [124], and more ambitious multi-layered versions of the original approach are currently being investigated, with the potential of being suited to arbitrary density distributions [125]. If manipulating these models is possible to attain a similar uncertainty model as in the uniform density model, one should nonetheless be wary of the substantial increase in the complexity of the uncertainty model.

10.1.3 Uncertainty in exterior spherical harmonics expansion arising from an uncertain shape

Besides the polyhedron gravity model, the other workhorse of small body gravity models is the exterior spherical harmonics expansion of the gravity field created by a constant density shape, in the form given in Chapter 2. In a similar fashion to what was achieved for the polyhedron gravity model, the closed-form expressions of the coefficients derived by Werner could be differentiated so as to yield the partial derivatives in the coefficients with respect to the shape coordinates. Backtracking the variance and covariance in the potential and accelerations should then be relatively straightforward, since these two quantities are linear combinations of the spherical harmonics coefficients.

10.1.4 Uncertainty propagation

The polyhedron gravity model derived in this thesis would be made most helpful to small-body mission designers when combined with a suitable uncertainty propagation scheme. That is, given the a-priori uncertainty in the spacecraft state at the epoch (possibly in the form of an-priori covariance matrix), this next step would consist in tying in the evaluation of the polyhedron gravity model uncertainty into the propagation of the spacecraft uncertainty itself. Beyond classical State Transition Matrix-based methods, State Transition Tensors [126], Polynomial Chaos Expansion [127] and Automatic Taylor Differentiation [128] could provide the necessary framework to embed the proposed uncertainty model into a more complete uncertainty propagation pipeline.

10.2 Survey & Mapping framework

10.2.1 SLAM performance

The bundle-adjuster and overlap graph implemented in the frame of this thesis are by no means a nec-plus-ultra, since the robotics community has been devoting tremendous effort into proposing more robust and efficient optical camera-based SLAM frameworks [129, 130], which have started to be considered by the aerospace community as well [131].

10.2.2 Observations planning

The current implementation of the Survey & Mapping phase relies on Lidar observations collected at a fixed observation rate. The immediate consequence for such a mode of operation is that the frequency at which observations are collected is never optimal: it is either too high or too low, with respect to the rate at which the relative pose evolves. The former case implies the unnecessary retrieval of point-cloud data, that serves no purpose besides increasing the spatial resolution of the reconstructed scene. The latter would put a greater burden on the bundle adjuster itself, possibly at the cost of discarding overlaps in the connectivity graph if the bundle adjuster does not succeed at correcting the relative alignment of the corresponding point-clouds. Moving away from this rigid functioning and let a managing entity schedule observations according to some optimality criterion would be a major improvement over the current framework, as the observation rate could then be continuously adjusted to reflect the current geometry and kinematics. Partially-Observable Markov Decision Processes (POMDPs) could be a possible path to follow in order to design such an observation planner, since they provide the necessary mathematical framework to determine the actions an agent must take in order to maximize some reward [132]. In the present case, the reward could be formulated as the percentage of the shape that has been covered by the observations. POMDPs have already been applied to autonomous spacecraft operations [133, 134] and could thus offer a way forward towards more decisional autonomy, for the benefit of performance and robustness.

Bibliography

[1] Simone D'Amico, J.-S. Ardaens, G. Gaias, H. Benninghoff, B. Schlepp, and J. L. Jørgensen. Noncooperative Rendezvous Using Angles-Only Optical Navigation: System Design and Flight Results. Journal of Guidance, Control, and Dynamics, 36(6):1576–1595, 2013.

[2] John A. Christian, Shane B. Robinson, Christopher N. D'Souza, and Jose P. Ruiz. Cooperative Relative Navigation of Spacecraft Using Flash Light Detection and Ranging Sensors. Journal of Guidance, Control, and Dynamics, 37(2):452–465, 2014.

[3] Brent E. Tweddle, Alvar Saenz-Otero, John J. Leonard, and David W. Miller. Factor Graph Modeling of Rigid-body Dynamics for Localization, Mapping, and Parameter Estimation of a Spinning Object in Space. Journal of Field Robotics, 32(6):897–933, 2015.

[4] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson Surface Reconstruction. Proceedings of the Symposium on Geometry Processing, pages 61–70, 2006.

[5] Robert A. Werner and Daniel J. Scheeres. Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia. Celestial Mechanics and Dynamical Astronomy, 65(3):313–344, 1997.

[6] Karl Heinz Glassmeier, Hermann Boehnhardt, Detlef Koschny, Ekkehard Kührt, and Ingo Richter. The Rosetta mission: Flying towards the origin of the solar system. Space Science Reviews, 128(1-4):1–21, 2007.

[7] F E Demeo and B Carry. Solar System evolution from compositional mapping of the asteroid belt. Nature, 505(7485):629–634, 2014.

[8] Jean-marc Petit, Alessandro Morbidelli, and John Chambers. The Primordial Excitation and Clearing of the Asteroid Belt. Icarus, 153(2):338–347, 2001.

[9] C T Russell, C A Raymond, A Coradini, H Y Mcsween, M T Zuber, A Nathues, M C De Sanctis, R Jaumann, A S Konopliv, F Preusker, S W Asmar, R S Park, R Gaskell, H U Keller, S Mottola, T Roatsch, J E C Scully, D E Smith, P Tricarico, M J Toplis, U R Christensen, W C Feldman, D J Lawrence, and T J Mccoy. Dawn at Vesta : Testing the Protoplanetary Paradigm. Nature, 336(May), 2012.

[10] Larry A. Lebofsky. Asteroid 1 Ceres: evidence for water of hydration. Monthly Notices of the Royal Astronomical Society, 182(1), 1978.

[11] Vladimir Zakharov, Seungwon Lee, Paul Von Allmen, Laurence O Rourke, Dominique Bockele, David Teyssier, Anthony Marston, Thomas Mu, Jacques Crovisier, and M Antonietta Barucci. Localized sources of water vapour on the dwarf planet (1) Ceres. Nature, 505:525–527, 2014.

[12] C T Russell, C A Raymond, E Ammannito, D L Buczkowski, M C De Sanctis, H Hiesinger, R Jaumann, A S Konopliv, H Y Mcsween, A Nathues, R S Park, C M Pieters, T H Prettyman, T B Mccord, L A Mcfadden, S Mottola, M T Zuber, S P Joy, C Polanskey, M D Rayman, P J Chi, J P Combe, A Ermakov, R R Fu, M Hoffmann, Y D Jia, S D King, D J Lawrence, S Marchi, F Preusker, T Roatsch, O Ruesch, P Schenk, M N Villarreal, and N Yamashita. Dawn arrives at Ceres: Exploration of a small, volatile-rich world. Science, 353(6303):2–5, 2016.

[13] D S Lauretta, S S Balram-Knutson, E Beshore, W V Boynton, C Drouet d'Aubigny, D N DellaGiustina, H L Enos, D R Golish, C W Hergenrother, E S Howell, C A Bennett, E T Morton, M C Nolan, B Rizk, H L Roper, A E Bartels, B J Bos, J P Dworkin, D E Highsmith, D A Lorenz, L F Lim, R Mink, M C Moreau, J A Nuth, D C Reuter, A A Simon, E B Bierhaus, B H Bryan, R Ballouz, O S Barnouin, R P Binzel, W F Bottke, V E Hamilton, K J Walsh, S R Chesley, P R Christensen, B E Clark, H C Connolly, M K Crombie, M G Daly, J P Emery, T J McCoy, J W McMahon, Daniel J. Scheeres, S Messenger, K Nakamura-Messenger, K Righter, and S A Sandford. OSIRIS-REx: Sample Return from Asteroid (101955) Bennu. Space Science Reviews, 212(1):925–984, oct 2017.

[14] Yuichi Tsuda, Makoto Yoshikawa, Takanao Saiki, Satoru Nakazawa, and Sei-ichiro Watanabe. Acta Astronautica Hayabusa2 Sample return and kinetic impact mission to near-earth asteroid Ryugu. Acta Astronautica, (January):1–7, 2018.

[15] Luis W Alvarez. Experimental evidence that an asteroid impact led to the extinction of many species 65 million years ago. Proceedings of the National Academy of Sciences of the United States of America, 80(January):627–642, 1983.

[16] Christopher Chyba, Paul Thomas, and Kevin Zahnle. The 1908 Tunguska explosion: atmospheric disruption of a stony asteroid. Nature, 361(1), 1993.

[17] Marco Micheli, Richard J Wainscoat, and Larry Denneau. Detectability of Chelyabinsk-like impactors with Pan-STARRS. Icarus, 303:265–272, 2018.

[18] Rob Landis and Lindley Johnson. Advances in planetary defense in the United States (in press). Acta Astronautica, 2018.

[19] William F Bottke, David Vokrouhlick, David P Rubincam, and David Nesvorn. The Yarkovsky and YORP Effects : Implications for Asteroid Dynamics. Annual Review of Earth and Planetary Sciences, 34, 2006.

[20] Dana G Andrews, K D Bonner, A W Butterworth, H R Calvert, B R H Dagang, K J Dimond, L G Eckenroth, J M Erickson, B A Gilbertson, N R Gompertz, O J Igbinosun, T J Ip, B H Khan, S L Marquez, N M Neilson, C O Parker, E H Ransom, B W Reeve, T L Robinson, M Rogers, P M Schuh, C J Tom, S E Wall, N Watanabe, and C J Yoo. Defining a successful commercial asteroid mining program. Acta Astronautica, 108:106–118, 2015.

[21] Rudolf Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(Series D):35–45, 1960.

[22] Leonard Mcgee and Stanley Schmidt. Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry. Technical Report November, 1985.

[23] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman Filtering for Spacecraft Attitude Estimation. Journal of Guidance, Control, and Dynamics, 5(5):417–429, 1982.

[24] J. W. Zielenbach, J. W. O'Neil, J. F. Jordan, S. K. Wong, R. T. Mitchell, W. A. Webb, and P. E. Koskela. Mariner 9 Navigation, NASA Technical Report 32-1586. Technical report, 1973.

[25] S. P. Synnott, W. M. Owen, J. E. Riedel, J. A. Stuve, and R. M. Vaughan. Optical navigation during the Voyager Neptune encounter. In Astrodynamics Conference, Guidance, Navigation, and Control and Co-located Conferences, 1990.

[26] Kevin Criddle, Julie Bellerose, Dylan Boone, William Owen, and Duane Roth. Optical Navigation During Cassini's Solstice Mission. In AAS/AIAA Astrodynamics Specialist Conference, Columbia River Gorge, Stevenson, WA, 2017.

[27] Francesco Castellini, David Antal-Wokes, Ramon Pardo de Santayana, and Klaas Vantournhout. Far Approach Optical Navigation and Comet Photometry for The Rosetta Mission. In Proceedings of the 25th ISSFD, number 1, pages 1–19, 2015.

[28] J. E. Riedel, W. M. Owen, J. Stuve, S. P. Synnott, and R. M. Vaughan. Optical Navigation for the Galileo Gaspra Encounter.

[29] B. G. Williams, P. G. Antreasian, J. J. Bordi, E. Carranza, S. R. Chesley, C. E. Helfrich, J.K. Miller, W. M. Owen, and Wang. T. C. Navigation for NEAR Shoemaker: the First Spacecraft to Orbit an Asteroid. In AAS/AIAA Astrodynamics Specialist Conference, 2001.

[30] W. M. Owen and Wang. T. C. NEAR Optical Navigation at Eros. In AAS/AIAA Astrodynamics Specialist Conference, Quebec City, Canada, 2001.

[31] Takashi Kubota, Tatsuaki Hashimoto, JuN'Ichiro Kawaguchi, Masashi Uo, and KeN'Ichi Shirakawa. Guidance and navigation of hayabusa spacecraft for asteroid exploration and sample return mission. 2006 SICE-ICASE International Joint Conference, pages 2793–2796, 2006.

[32] Francesco Castellini, Ramon Pardo De Santayana, Klaas Vantournhout, and Mathias Lauer. Operational Experience and Assessment of the Implementation of the Maplet Technique for Rosetta's Optical Navigation. In AAS/AIAA Astrodynamics Specialist Conference, pages 1–20, 2017.

[33] R W Gaskell, O S Barnouin-Jha, Daniel J. Scheeres, a S Konopliv, T Mukai, S Abe, J Saito, M Ishiguro, T Kubota, T Hashimoto, J Kawaguchi, M Yoshikawa, K Shirakawa, T Kominato, N Hirata, and H Demura. Characterizing and navigating small bodies with imaging data. Meteoritics and Planetary Science, 43(6):1049–1061, 2008.

[34] Coralie D Jackman, Derek S Nelson, Leilah K Mccarthy, Tiffany J Finley, Andrew J Liounis, Kenneth M Getzandanner, and Peter G Antreasian. Optical Navigation Concept of Operations for the Osiris-Rex Mission. In AAS/AIAA Space Flight Mechanics Meeting, number Code 595, pages 1–18, San Antonio, TX, 2018.

[35] Bradley J Clement and Mark D Johnston. Design of a Deep Space Network Scheduling Application. In Proceedings of the International Workshop on Planning and Scheduling for Space, 2006.

[36] Raymond B Frauenholz, Ramachandra S Bhat, Steven R Chesley, Nickolaos Mastrodemos, William M Owen, and Mark S Ryne. Deep Impact Navigation System Performance. Journal of Spacecraft and Rockets, 45(1):39–56, 2008.

[37] Joseph Starek, Marco Pavone, Issa A. Nesnas, and Behçet Açikmese. Spacecraft Autonomy Challenges for Next Generation Space Missions. In Lecture Notes in Control and Information Sciences, volume 460, pages 1–34. 2016.

[38] Marshall H Kaplan, Bradley Boone, Robert Brown, Thomas B Criss, and Edward W Tunstel. Engineering Issues for All Major Modes of In Situ Space Debris Debris Capture. AIAA Space, (September):1–20, 2010.

[39] Gregory A Neumann. Some Aspects of Processing Extraterrestrial Lidar Data: Clementine, NEAR, MOLA. International Archives of Photogrammetry and Remote Sensing, 34(3):22–24, 2001.

[40] Katsuhiko Tsuno, Eisuke Okumura, Yoshihiko Katsuyama, Takahide Mizuno, Tatsukaki Hashimoto, Michio Nakayama, and Hiroshi Yuasa. Lidar on Board Asteroid Explorer Hayabusa. Manufacturing Engineering, 2006(June):27–30, 2006.

[41] T Mizuno, T Kase, T Shiina, M Mita, N Namiki, H Senshu, R Yamada, H Noda, H Kunimori, N Hirata, F Terui, and Y Mimasu. Development of the Laser Altimeter (LIDAR) for Hayabusa2. Space Science Reviews, 208(1-4):33–47, 2017.

[42] M. G. Daly, O. S. Barnouin, C. Dickinson, J. Seabrook, C. L. Johnson, G. Cunningham, T. Haltigin, D. Gaudreau, C. Brunet, I. Aslam, A. Taylor, E. B. Bierhaus, W. Boynton, M. Nolan, and Dante Lauretta. The OSIRIS-REx Laser Altimeter (OLA) Investigation and Instrument. Space Science Reviews, 212(1-2):899—-924, 2017.

[43] John A. Christian and Scott Cryan. A Survey of LIDAR Technology and its Use in Spacecraft Relative Navigation. AIAA Guidance, Navigation, and Control (GNC) Conference, pages 1–7, 2013.

[44] John A. Christian, Mogi Patangan, and Heather Hinkel. Comparison of Orion vision navigation sensor performance from STS-134 and the Space Operations Simulation Center. AIAA Guidance, Navigation, and Control Conference, pages 1–18, 2012.

[45] Advanced Scientific Concepts. GoldenEye 3D Flash LIDAR Space Camera.

[46] Farhad Aghili, Marcin Kuryllo, Galina Okouneva, and Chad English. Fault-Tolerant Position/Attitude Estimation of Free-Floating Space Objects Using a Laser Range Sensor. IEEE Sensors Journal, 11(1):176–185, 2011.

[47] Benjamin B Reed, Robert C Smith, and Bo Naasz. The Restore - L Servicing Mission. In AIAA Space, number September, pages 1–8, 2016.

[48] M.D. Lichter and S. Dubowsky. State, shape, and parameter estimation of space objects from range images. IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, 3(April):2974–2979, 2004.

[49] John J Leonard, Independence Way, and Hugh F Durrant-whyte. Simultaneous Map Building and Localization for an Autonomous Mobile Robot. In IEEE/RSJ International Workshop on Intelligent Robots and Systems, number 91, 1991.

[50] Hans-andrea Loeliger. An introduction to factor graphs. IEEE Signal Processing Magazine, 21(January):28–41, 2004.

[51] Nima Keivan, Alonso Patron-Perez, and Gabe Sibley. Asynchronous adaptive conditioning for visual-inertial SLAM. Springer Tracts in Advanced Robotics, 109:309–321, 2016.

[52] Nicholas Carlevaris-Bianco, Michael Kaess, and Ryan M. Eustice. Generic node removal for factor-graph SLAM. IEEE Transactions on Robotics, 30(6):1371–1385, 2014.

[53] Ann Dietrich and Jay W McMahon. Orbit Determination with Least-Squares and Flash Lidar Measurements in Proximity to Small Bodies. In AAS/AIAA Space Flight Mechanics Meeting, number 17-276 in AAS, feb 2017.

[54] Brian Coltin, Jesse Fusco, Zack Moratto, Oleg Alexandrov, and Robert Nakamura. Localization from Visual Landmarks on a Free-flying Robot. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4377–4382, 2016.

[55] S Izadi, Kim, O Hilliges, D Molyneaux, R Newcombe, P Kohli, J Shotton, S Hodges, D Freeman, A Davison, and A Fitzgibbon. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th annual ACM User Interface Software and Technology Symposium - UIST '11, pages 559–568, 2011.

[56] Steven J Ostro, R. Scott Hudson, Lance A.M. Benner, Jon D. Giorgini, Christopher Magri, Jean-Luc Margot, and Michael C. Nolan. Asteroid radar astronomy. Asteroids III, pages 151–168, 2002.

[57] Daniel J. Scheeres, E. G. Fahnestock, Steven J Ostro, Jean-Luc Margot, L. A. M. Benner, Stephen Broschart, Julie Bellerose, Jon D Giorgini, Michael C Nolan, Christopher Magri, Petr Pravec, P. Scheirich, R. Rose, R. F. Jurgens, E M De Jong, S. Suzuki, and E. M. De Jong. Dynamical Configuration of Binary Near-Earth Asteroid (66391) 1999 KW4. Science, 314(November):1280–1283, 2006.

[58] Michael C. Nolan, Christopher Magri, Ellen S. Howell, Lance A.M. Benner, Jon D. Giorgini, Carl W. Hergenrother, R. Scott Hudson, Dante Lauretta, Jean Luc Margot, Steven J. Ostro, and Daniel J. Scheeres. Shape model and surface properties of the OSIRIS-REx target Asteroid (101955) Bennu from radar and lightcurve observations. Icarus, 226(1):629–640, 2013.

[59] Daniel J. Scheeres, S G Hesar, S Tardivel, M Hirabayashi, D Farnocchia, Jay W. McMahon, S R Chesley, O Barnouin, R P Binzel, W F Bottke, M G Daly, J P Emery, C W Hergenrother,

Dante Lauretta, J R Marshall, P Michel, M C Nolan, and K J Walsh. The geophysical environment of Bennu. Icarus, 276:116–140, 2016.

[60] Michael W. Busch, Steven J. Ostro, Lance A.M. Benner, Jon D. Giorgini, Raymond F. Jurgens, Randy Rose, Christopher Magri, Petr Pravec, Daniel J. Scheeres, and Stephen B. Broschart. Radar and optical observations and physical modeling of near-Earth Asteroid 10115 (1992 SK). Icarus, 181(1):145–155, 2006.

[61] Johanna Torppa, Mikko Kaasalainen, Tadeusz Michałowski, Tomasz Kwiatkowski, Agnieszka Kryszczyńska, Peter Denchev, and Richard Kowalski. Shapes and rotational properties of thirty asteroids from photometric data. Icarus, 164(2):346–383, 2003.

[62] J.K. Miller, A.S. Konopliv, P.G. Antreasian, J.J. Bordi, S. Chesley, C.E. Helfrich, W.M. Owen, T.C. Wang, B.G. Williams, D. K. Yeomans, and Daniel J. Scheeres. Determination of Shape, Gravity, and Rotational State of Asteroid 433 Eros. Icarus, 155(1):3–17, 2002.

[63] K Muinonen. Introducing the Gaussian shape hypothesis for asteroids and comets. Astronomy and Astrophysics, 332:1087–1098, 1998.

[64] J. Torppa, V. P. Hentunen, P. Pääkkönen, P. Kehusmaa, and K. Muinonen. Asteroid shape and spin statistics from convex models. Icarus, 198(1):91–107, 2008.

[65] Yuan Ren and Jinjun Shan. Reliability-Based Soft Landing Trajectory Optimization near Asteroid with Uncertain Gravitational Field. Journal of Guidance, Control, and Dynamics, 38(9), 2015.

[66] J C P Melman, E Mooij, and R Noomen. State propagation in an uncertain asteroid gravity field. Acta Astronautica, 91:8–19, 2013.

[67] C. Ma, E. F. Arias, T. Mm. Eubanks, A. L. Fey, A-M. Gontier, C. S. Jacobs, O. J. Sovers, B. A. Archinal, and P. Charlot. The International Celestial Reference Frame as Realized by Very Long Baseline Interferometry. The Astronomical Journal, 116(1), 1998.

[68] Hanspeter Schaub and John L Junkins. Analytical mechanics of space systems. AIAA Education Series, 2003.

[69] David A. Vallado. Fundamental of Astrodynamics and Applications. 2013.

[70] Benjamin Bercovici and Jay McMahon. SBGAT: The Small Body Geophysical Analysis Tool, 2019.

[71] Jon Louis Bentley. Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM, 18(9):509–517, 1975.

[72] P J Besl, N D McKay, and Anonymous. a Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14:239–256, 1992.

[73] Andrew Rhodes, Eric Kim, John A. Christian, and Thomas Evans. LIDAR-based Relative Navigation of Non-Cooperative Objects Using Point Cloud Descriptors. AIAA/AAS Astrodynamics Specialist Conference, (September):1–12, 2016.

[74] Szymon Rusinkiewicz and Mark Levoy. Efficient variants of the ICP algorithm. Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM, 2001-Janua:145–152, 2001.

[75] Andrew Mastin, Jeremy Kepner, and John Fisher. Automatic registration of LIDAR and optical images of urban scenes. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009, pages 2639–2646, 2009.

[76] TW Lim and AJ Toombs. Pose Estimation Using a Flash Lidar. AIAA SciTech: AIAA Guidance, Navigation, and Control Conference, (January):1–16, 2014.

[77] James Diebel. Representing Attitude : Euler Angles , Unit Quaternions , and Rotation Vectors. Matrix, 58(15-16):1–35, 2006.

[78] Puneet Singla, Daniele Mortari, and John L Junkins. How to avoid singularity when using Euler angles? Advances in the Astronautical Sciences, 119(SUPPL.):1409–1426, 2005.

[79] Panagiotis Tsiotras. Stabilization and optimality results for the attitude control problem. Journal of Guidance, Control, and Dynamics, 19(4):772–779, 1996.

[80] S. R. Marandi and V. J. Modi. A preferred coordinate system and the associated orientation representation in attitude dynamics. Acta Astronautica, 15(11):833–843, 1987.

[81] John Crassidis and Floyd Landis Markley. Attitude Estimation Using Modified Rodrigues Parameters. Nasa Conference Publication, pages 71–84, 1996.

[82] Stephen A. O'Keefe and Hanspeter Schaub. Shadow set considerations for modified rodrigues parameter attitude filtering. Advances in the Astronautical Sciences, 150(6):2777–2786, 2014.

[83] Hanspeter Schaub, Panagiotis Tsiotras, and John L Junkins. Principal Rotation Representation of Proper NxN Orthogonal Matrices. International Journal of Engineering Science, 33(15):2277–2295, 1995.

[84] Benjamin Bercovici and Jay W. McMahon. Point-Cloud Processing Using Modified Rodrigues Parameters for Relative Navigation. Journal of Guidance, Control, and Dynamics, 40(12), 2017.

[85] Szymon Rusinkiewicz. Derivation of Point-to-Plane Minimization. pages 1–4, 2013.

[86] Helmut Pottmann, Stefan Leopoldseder, and Michael Hofer. Registration without ICP. Computer Vision and Image Understanding, 95(1):54–71, 2004.

[87] Byron Tapley, Bob Schutz, and George Born. Preface. In Byron D Tapley, Bob E Schutz, and George H Born, editors, Statistical Orbit Determination, pages xi – xv. Academic Press, Burlington, 2004.

[88] F. Landis Markley. Multiplicative vs . Additive Filtering for Spacecraft Attitude Determination Quaternion estimation. Journal of Guidance, Control, and Dynamics, 26(2):311–317, 2003.

[89] Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas. Registration of point cloud data from a geometric optimization perspective. Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing - SGP '04, page 22, 2004.

[90] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustmenta modern synthesis. In International workshop on vision algorithms, volume 1, pages 298–372. Springer, 1999.

[91] E Mouragnon, M Lhuillier, M Dhome, F Dekeyser, P Sayd, and Pascal Cnrs. Monocular Vision Based SLAM for Mobile Robots. In 18th International Conference on Pattern Recognition, 2006.

[92] Hauke Strasdat, J M M Montiel, and Andrew J Davison. Real-time Monocular SLAM : Why Filter ? In 2010 IEEE International Conference on Robotics and Automation, pages 2657–2664. IEEE, 2010.

[93] Michael I Jordan and Lei Xu. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. Neural Computation, 8(1):129–151, 1996.

[94] Conrad Sanderson. Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments. Technical Report, NICTA:1–16, 2010.

[95] Pierre Bezier. The Mathematical Basis of the UNIURF CAD System. Butterworth-Heinemann, Oxford, United Kingdom, 2014.

[96] Yang Yu and He-Xi Baoyin. Routing the asteroid surface vehicle with detailed mechanics. Acta Mechanica Sinica, 30:301–309, 2014.

[97] Yang Liu, Helmut Pottmann, and Wenping Wang. Constrained 3D shape reconstruction using a combination of surface fitting and registration. Computer-Aided Design, 38(6):572–583, 2006.

[98] William E Lorensen and Harvey E Cline. Marching Cubes: A High Resolution 3D Surface construction Algorithm. Computer Graphics, 21(4):163–169, 1987.

[99] Michael Kazhdan and Hugues Hoppe. Screened Poisson Surface Reconstruction. ACM Transactions on Graphics, 32(3):1–13, 2013.

[100] Efi Fogel, Monique Teillaud, The Computational, Geometry Algorithms, and Library Cgal. The Computational Geometry Algorithms Library CGAL. Association for Computing Machinery, 47(185), 2013.

[101] Mauro Pontani and Bruce a. Conway. Particle Swarm Optimization Applied to Space Trajectories. Journal of Guidance, Control, and Dynamics, 33(5):1429–1441, 2010.

[102] Jing J. Liang, A. K. Qin, Ponnuthurai Nagaratnam Suganthan, and S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 10(3):281–295, 2006.

[103] Anthony R. Dobrovolskis. Inertia of Any Polyhedron. Icarus, 124:698–704, 1996.

[104] R. Gaskell, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, N. Hirata, S. Abe, O. Barnouin-Jha, and Daniel J. Scheeres. Gaskell Itokawa Shape, Model V1.0. HAY-A-AMICA-5-ITOKAWASHAPE-V1.0. NASA Planetary Data System, 2008.

[105] F. Preusker, F. Scholten, K.-D. Matz, T. Roatsch, K. Willner, S. F. Hviid, J. Knollenberg, L. Jorda, P. J. Gutiérrez, E. Kührt, S. Mottola, M. F. A'Hearn, N. Thomas, H. Sierks, C. Barbieri, P. Lamy, R. Rodrigo, D. Koschny, H. Rickman, H. U. Keller, J. Agarwal, M. A. Barucci, J.-L. Bertaux, I. Bertini, G. Cremonese, V. Da Deppo, B. Davidsson, S. Debei, M. De Cecco, S. Fornasier, M. Fulle, O. Groussin, C. Güttler, W.-H. Ip, J. R. Kramm, M. Küppers, L. M. Lara, M. Lazzarin, J. J. Lopez Moreno, F. Marzari, H. Michalik, G. Naletto, N. Oklay, C. Tubiana, and J.-B. Vincent. Shape model, reference system definition, and cartographic mapping standards for comet 67P / Churyumov-Gerasimenko Stereo-photogrammetric analysis of Rosetta / OSIRIS image data. Astronomy and Astrophysics, 583:A33, 2015.

[106] Benjamin Bercovici and Jay W. McMahon. Autonomous Shape Determination Using Flash-Lidar Observations and Bezier patches. In Proceedings of the 41st Annual Guidance and Control Conference, 2018.

[107] Daniel J. Scheeres. Orbital Motion in Strongly Perturbed Environments. Springer Berlin Heidelberg, 2012, 2012.

[108] Robert a. Werner. Spherical harmonic coefficients for the potential of a constant-density polyhedron. Computers & Geosciences, 23(10):1071–1077, 1997.

[109] Dimitrios Tsoulis and Sveto Petrovi. On the singularities of the gravity field of a homogeneous polyhedral body. Geophysics, 66(2):535–539, 2001.

[110] D Scheeres, R Gaskell, Shinsuke Abe, O Barnouin, T Hashimoto, J Kawaguchi, Takashi Kubota, J Saito, Masayuki Yoshikawa, N Hirata, T Mukai, M Ishiguro, T Kominato, K Shirakawa, and M Uo. The Actual Dynamical Environment About Itokawa. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, number August, 2006.

[111] Benjamin Bercovici and Jay W. Mcmahon. Inertia Parameter Statistics of An Uncertain Small Body Shape. Icarus, 328(8):32–44, 2019.

[112] Salomon Kullback and R. A. Leibler. On Information and Sufficiency. The Annals of Mathematical Statistics, 22(1):79–86, 1951.

[113] Edward F Tedesco, Paul V Noah, Meg Noah, and Stephan D Price. The Supplemental IRAS Minor Planet Survey. The Astronomical Journal, 123(2):1056–1085, 2002.

[114] Michael K Shepard, James Richardson, Patrick A Taylor, Linda A Rodriguez-ford, Al Conrad, Imke De Pater, Mate Adamkovics, Katherine De Kleer, Jared R Males, Katie M Morzinski, Laird M Close, Mikko Kaasalainen, Matti Viikinkoski, Bradley Timerson, Vishnu Reddy, Christopher Magri, Michael C Nolan, Ellen S Howell, Lance A M Benner, Jon D Giorgini, Brian D Warner, and Alan W Harris. Radar observations and shape model of asteroid 16 Psyche. Icarus, 281:388–403, 2017.

[115] David Y Oh, Steve Collins, Dan Goebel, Bill Hart, Gregory Lantoine, Steve Snyder, Greg Whiffen, Linda Elkins-tanton, Peter Lord, Zack Pirkl, and Lee Rotlisburger. Development of the Psyche Mission for NASA's Discovery Program. In 35th International Electric Propulsion Conference, pages 1–19, 2017.

[116] Kenshiro Oguri, Gregory Lantoine, Bill Hart, and Jay W Mcmahon. Science Orbit Design with Frozen Beta angle: Theory and Application to Psyche mission. In AAS/AIAA Space Flight Mechanics Meeting, pages 1–15, Ka'anapali, HI, 2019.

[117] Michael W Busch, Steven J Ostro, Lance A M Benner, Marina Brozovic, Jon D Giorgini, Joseph S Jao, Daniel J Scheeres, Christopher Magri, Michael C Nolan, Ellen S Howell, Patrick A Taylor, Jean-luc Margot, and Walter Brisken. Radar observations and the shape of near-Earth asteroid 2008 EV5. Icarus, 212(2):649–660, 2011.

[118] Byron D Tapley, Bob E Schutz, and George H Born. Chapter 4 - Fundamentals of Orbit Determination. In Byron D Tapley, Bob E Schutz, and George H Born, editors, Statistical Orbit Determination, pages 159–284. Academic Press, Burlington, 2004.

[119] Ann Dietrich and Jay W. McMahon. Orbit Determination Using Flash Lidar Around Small Bodies. Journal of Guidance, Control, and Dynamics, 40(3):1–16, 2016.

[120] Christopher D. Karlgaard and Hanspeter Schaub. Nonsingular attitude filtering using Modified Rodrigues Parameters. The Journal of the Astronautical Sciences, 57(4):777–791, 2009.

[121] Ann Dietrich and Jay W McMahon. Robust Orbit Determination with Flash Lidar Around Small Bodies. Journal of Guidance, Control, and Dynamics, 41(10):2163–2184, jul 2018.

[122] Byron D Tapley, Bob E Schutz, and George H Born. Chapter 6 - Consider Covariance Analysis. In Byron D Tapley, Bob E Schutz, and George H Born, editors, Statistical Orbit Determination, pages 387–438. Academic Press, Burlington, 2004.

[123] Bobby G Williams. Technical Challenges and Results for Navigation of NEAR Shoemaker. Johns Hopkins APL technical digest, 23(1):34—-45, 2002.

[124] M G D Urso. Gravity effects of polyhedral bodies with linearly varying density. Celestial Mechanics and Dynamical Astronomy, 120(4):349–372, 2014.

[125] William Ledbetter, Rohan Sood, and James Keane. Smallsat Swarm Gravimetry : Revealing The Interior Structure Of Asteroids And Comets. In AAS Astrodynamics Specialists Conference, Snowbird, UT, pages 1–19, 2018.

[126] K Fujimoto and D J Scheeres. Analytical Nonlinear Propagation of Uncertainty in the Two-Body Problem. Journal of Guidance, Control, and Dynamics, 35(2), 2012.

[127] Brandon A Jones, Alireza Doostan, and George H Born. Nonlinear Propagation of Orbit Uncertainty Using Non-Intrusive Polynomial Chaos. Journal of Guidance, Control, and Dynamics, 36(2), 2013.

[128] Roberto Armellin and Pierluigi Di. High-order expansion of the solution of preliminary orbit determination problem. Celestial Mechanics and Dynamical Astronomy, 112(3):331–352, 2012.

[129] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM : A Versatile and Accurate Monocular. IEEE transactions on robotics, 31(5):1147–1163, 2015.

[130] Tim Caselitz. Monocular Camera Localization in 3D LiDAR Maps. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1926–1931. IEEE, 2016.

[131] Mehregan Dor and Panagiotis Tsiotras. ORB-SLAM Applied to Spacecraft Non-Cooperative Rendezvous. In AIAA SciTech Forum, number January, 2018.

[132] Hanna Kurniawati, David Hsu, and Wee Sun Lee. SARSOP : Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In Robotics: Science and systems, number June, 2008.

[133] Vincenzo Pesce, Ali-akbar Agha-mohammadi, and Michèle Lavagna. Autonomous Navigation & Mapping of Small Bodies. In 2018 IEEE Aerospace Conference, 2018.

[134] Andrew Harris, Thibaud Teil, and Hanspeter Schaub. Spacecraft Decision-Making Autonomy Using Deep Reinforcement Learning. In AAS Astrodynamics Specialists Conference, Maui, HI, pages 1–19, 2019.