GNC architecture for autonomous robotic capture of a non-cooperative target: preliminary concept design[⋆,⋆⋆]

Marko Jankovic[*], Jan Paul, Frank Kirchner

Robotics Innovation Center, DFKI GmbH and University of Bremen,
Robert-Hooke-Straße 1, 28359 Bremen, Germany

Abstract

Recent studies of the space debris population in low Earth orbit (LEO) have concluded that certain regions have already reached a critical density of objects. This will eventually lead to a cascading process called the Kessler syndrome. The time may have come to seriously consider active debris removal (ADR) missions as the only viable way of preserving the space environment for future generations. Among all objects in the current environment, the SL-8 (Kosmos 3M second stages) rocket bodies (R/Bs) are some of the most suitable targets for future robotic ADR missions. However, to date, an autonomous relative navigation to and capture of an non-cooperative target has never been performed. Therefore, there is a need for more advanced, autonomous and modular systems that can cope with uncontrolled, tumbling objects. The guidance, navigation and control (GNC) system is one of the most critical ones. The main objective of this paper is to present a preliminary concept of a modular GNC architecture that should enable a safe and fuel-efficient capture of a known but uncooperative target, such as Kosmos 3M R/B. In particular, the concept was developed having in mind the most critical part of an ADR mission, i. e. close range proximity operations, and state of the art algorithms in the field of autonomous rendezvous and docking. In the end, a brief description of the hardware in the loop (HIL)

[⋆] This document is an accepted version of the manuscript available onlilne at: https://doi.org/10.1016/j.asr.2015.05.018.

[*] Corresponding Author

Email address: marko.jankovic@dfki.de (Marko Jankovic)

testing facility is made, foreseen for the practical evaluation of the developed architecture.

Keywords: GNC; Active debris removal; Space debris; Proximity operations; Space robotics

1. Introduction

The launch of the first artificial satellite, Sputnik-1, a sphere of 58 cm in diameter and mass of 84 kg, in 1957 marked the beginning of human space exploration. However, it also marked the birth of non-functional, man-made, earth-orbiting objects denoted as *space debris*. Since then, there have been more than 4900 launches, which placed around 6600 satellites in orbit. Almost one half of them is still in orbit and the total amount of mass of intact space hardware is around 6300 t. However, those numbers do not include the fragmented objects. Considering also those, the number of objects goes even higher. Indeed, the total number of objects tracked routinely by the United States Space Surveillance Network (US SSN) is around 23,000 for objects larger than 5-10 cm in low Earth orbit (LEO[1]) and 30 cm-1 m in geostationary Earth orbit (GEO[2]) (Space Debris Office, 2013; Wormnes et al., 2013). The population of non traceable particles is estimated to be approximately 500,000 units, for particles between 1-10 cm, and more than 100 million for those smaller than 1 cm (Orbital Debris Program Office, 2012).

The origin of 66 % of the cataloged objects is due to more than 200 recorded in-orbit fragmentation events, the majority of which were in-orbit explosions. 28 % of the cataloged objects is represented by decommissioned satellites, spent upper stages and other related objects. The operational satellites represent only 6 % of the total figure (Wormnes et al., 2013; Liou, 2011a).

Two recent collision events have however contributed, on their own, to more than half of the objects in the region below 1 000 km thus raising the public awareness of the space debris issue. The first event was the Chinese anti-satellite weapon test (ASAT) on the Fengyun-1C (FY-1C) weather satellite, which occurred in 2007, at an altitude of 862 km. The second was the

[1] Considered to be the region from the beginning of the space environment (i. e. around 100 km from the ground) up to an altitude of 2000 km.

[2] Considered to be a geosynchronous orbit 35,786 km above the equator.

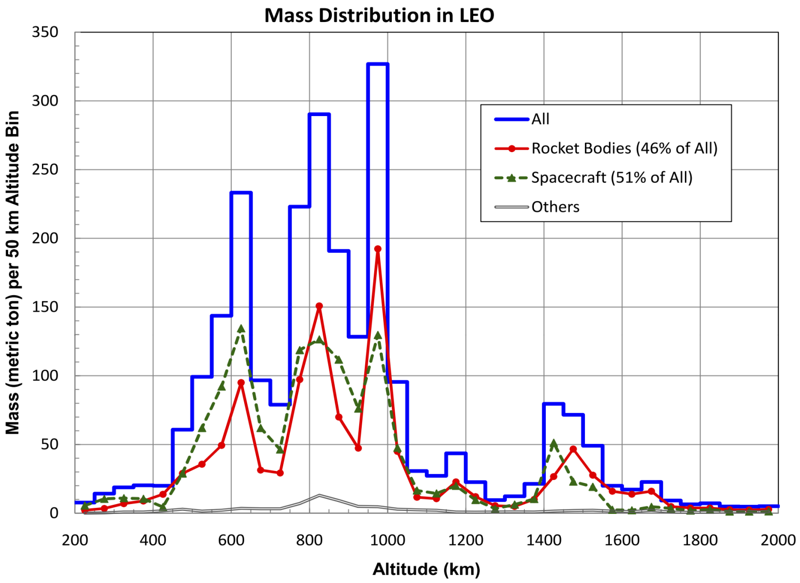

Figure 1: Mass distribution of space debris in LEO. The International Space Station is not included in the distribution (credit: Liou (2011b)-NASA Orbital Debris Program Office)

unintentional collision between the defunct Russian satellite Kosmos 2251 and the operational US satellite Iridium 33, which occurred in 2009, at an altitude of 789 km. This last event in particular has confirmed the concern of the international scientific community of onsetting, in LEO, of a self-sustaining, cascading process known as the *"Kessler syndrome"* (Liou and Johnson, 2008, 2009; Liou, 2011a). This event, first predicted by Kessler and Cour-Palais (1978), indicates a phenomenon where the number of objects is expected to increase exponentially due to mutual collisions between the objects creating a belt of debris around the Earth (Kessler and Cour-Palais, 1978). The LEO region is particularly susceptible to this phenomenon since it contains more than 40 % of the total in-orbit mass (i. e. around 2 500 t). More in detail, the majority of that mass is contained in altitudes around 600, 800 and 1000 km (see Figure 1). 97 % of that mass is represented by rocket bodies (R/Bs) and spacecrafts (S/Cs). The latter are mainly concentrated in the 600 km region, while the former are mainly in the 800 and 1000 km regions (Liou, 2011a).

To mitigate this phenomenon, various national and international organizations have issued a set of non binding space debris mitigation guidelines aimed among others at (Committee on the Peaceful Uses of Outer Space, 2014):

1. reducing the amount of space debris created during nominal operations
2. minimizing potential brake-ups and collisions
3. limiting the presence of non-operational satellites and rocket bodies

Nevertheless, recent studies have shown that those mitigation measures are not enough to stabilize the current space debris environment. In fact, Liou et al. (2010), in their study, concluded that the number of in-orbit objects bigger than 10 cm is expected to rise by 75 % in the next 200 years even with 90 % compliance to post-mission disposal measures and no future in-orbit explosions (see Figure 2). The assumed launch rate was the one of the previous years. Moreover, considering even the scenario of "no-future-launches", the population of space debris is expected to grow in LEO in the next 200 years. This means that in certain orbital regions the critical density of objects has been reached and an active removal of in-orbit mass has to be considered to stabilize the space debris environment. The active removal of only five objects per year, if started in 2020, coupled with 90 % implementation of mitigation measures, should be enough to maintain the number of objects comparable to that in 2011. In order to reduce the space debris population in LEO to the number it had prior to the two most recent brake-up events, the removal of more objects per year should be considered (Liou et al., 2010; Liou, 2011a).

The concept of active debris removal (ADR) has been around for some time, especially the one involving orbital robotics, due to its similarity to on-orbit servicing (OOS). The latter has its origins in early 1980s after the successful usage of the Space Shuttle remote manipulator in STS-2 mission (Yoshida and Wilcox, 2008). Despite this, the idea of an ADR never took off due to tremendous costs, legal and technical issues related to it. Moreover, until recently it has not been possible to quantify the real benefit of an ADR mission (Liou, 2011a).

Rendezvousing and capturing large uncooperative objects[3] is not an easy task. In fact, until today it has not been performed without humans in the loop. Naasz et al. (2010) state in a paper that: "...no spacecraft has

[3] Intended as objects that have lost the control authority and the ability to communicate their state either to the ground control or to the chaser spacecraft.

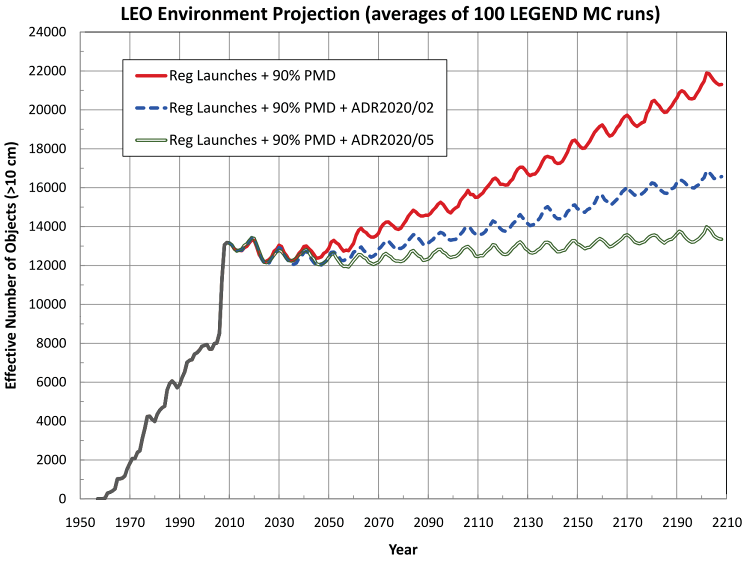

Figure 2: Benefits of using ADR to limit the LEO population (credit: Liou (2011b)-NASA Orbital Debris Program Office)

ever performed autonomous capture of a non-cooperative vehicle, and a full 6 degrees of freedom (DOF) relative navigation sensing to non-cooperative vehicle has only been shown to a limited extent."

The autonomy is requested in particular in the final phases of the approach of the chaser vehicle (chaser) to the target vehicle (target) due to the limited reaction time available to face anomalies and/or communication problems[4] that might occur (Nolet and Miller, 2007). The automated rendezvous and docking is nowadays the state of the art of the space technology (see for example (Personne et al., 2006)), but if ADR is going to be performed routinely new technological challenges need to be tackled. Most of them are related to the fact that a typical target is not sufficiently equipped for the capture. Thus, it does not have reflectors, markers or radio beacons that could ease the determination of its relative position and attitude. Moreover, no grappling features are usually available, making the capturing of the target even more complicated. In the end, the target might have some sort of tumbling

[4] Such as the transmission delay or the communication bandwidth.

motion[5] which poses strict requirements on the trajectory safety, due to the increased possibility of collision of the chaser with rotating appendages of the target.

It is worth noting that in this paper, automation and autonomy are intended as two different terms. They both indicate processes that can be executed without any human intervention. Automation involves software/hardware processes that substitute a manual routine by following a predetermined step-by-step sequences. However, they could still require human intervention to solve contingencies and unexpected behaviors. Autonomy, on the other hand, implies a more capable system that is able to perform actions and make decisions independently from the ground control. Thus, trying to emulate human processes rather than replacing them with a pre-programed sequences (Truszkowski et al., 2010).

Most of those technological challenges are somehow related to the GNC system making it one of the most critical pieces of the whole chaser spacecraft. Given its importance, not only in ADR, but in all space missions, there have been a great deal of fundamental studies in this area of research. However, it is very difficult to select one of them[6] that could readily solve all the phases related to a robotic capture of an uncooperative target. In fact, taking into account all the phases related to close range proximity operations is a difficult task. Different phases (e.g. fly-around, pose estimation, approach, manipulator deployment, grasping and stabilization of the compound) have different problems and considerations. Thus, most of the researchers tend to concentrate just on one phase or one part of the GNC system (e.g. navigation or control). A small body of work was dedicated specifically to the GNC architecture as a whole. Moreover, we have not been able to find until now a research dedicated specifically to the development of a GNC architecture for a robotic removal of upper stages.

To fill this gap and support future ADR missions, DFKI, within the initial training network (ITN) Stardust, has committed itself since November 2013 to study close range navigation and manipulation of uncooperative targets. Within that context, the following paper will present a preliminary concept design of a GNC architecture that should enable autonomous robotic cap-

[5] Intended in this paper as the target's rotation around at least one axis with an angular rate between 1 deg/s and 18 deg/s (Matsumoto et al., 2002).

[6] At least to our best knowledge.

ture of uncooperative upper stages. The novelty presented here consists of individuating the challenges and critical aspects of such an architecture, and presenting a series of state of the art algorithms that could populate that architecture. Moreover, a comprehensive description of current trends in the GNC for autonomous rendezvous and docking missions is also provided, hoping it could serve as a stepping-stone for the development of future GNC architectures for robotic ADR missions.

It is worth noting that in this paper, only the close range rendezvous phase of an ADR mission is considered. However, a quick overview of all mission phases will be illustrated for the sake of completeness. Furthermore, the target is assumed to be uncooperative although well known a priori.

The content of this paper is organized as follows. At first, a comprehensive description of the state-of-the-art in the field of autonomous rendezvous and docking/capture is presented in Section 2. Particular attention is given to the past missions, GNC architectures, and algorithms. Next, in Section 3, the envisioned ADR mission scenario is illustrated. A specific target is defined and major characteristics of the chaser are outlined. A preliminary concept of the GNC architecture is instead presented immediately after in Section 4. Various modules composing the architecture are described and the selection of algorithms that could be integrated within the individual modules is presented. A brief description of the robotics module, along with its interaction with the GNC architecture of the spacecraft is also made. The last part of this section is dedicated to a brief presentation of the hardware in the loop (HIL) testing facility intended to be used to validate the adequacy of the presented GNC architecture. The last section, i. e. Section 5, is dedicated to the conclusions and the future road map that will further improve the envisioned concept of the GNC architecture.

2. Autonomous rendezvous and docking: background and related work

Autonomous rendezvous and docking (ARVD) between two cooperative spacecrafts is not yet a routine operation, especially when one of the spacecrafts is non-cooperative. Nevertheless, given that it involves areas of research, such as pose[7] estimation, spacecraft control and path planning, there

[7] Pose intended as position and attitude.

has been a fair amount of work on those topics in the last few decades. Moreover, quite few missions have been able to accomplish some sort of autonomous rendezvous and proximity operations in the past and there are some of them planned for the near future in order to bridge the existing gap. Thus, this section will give a brief overview of some of the most relevant past and future missions dealing with autonomous rendezvous (ARV) as well as of some theoretical work done in the mentioned research areas. The overview does not pretend to be complete but is in our opinion quite representative of the ARV panorama.

2.1. Past and future missions

The first ever in orbit rendezvous occurred on December 15th, 1965, when astronauts Walter Schirra and Thomas Stafford aligned their spacecraft *Gemini VI* with *Gemini VII* piloted by James A. Lowell and commanded by Frank Borman. This initial achievement was quickly overrun several months later, on March 16th, 1966, when the first-ever successful orbital rendezvous and docking was performed by astronauts Neil Armstrong and Dave Scott. During it they rendezvoused and docked their *Gemini VIII* spacecraft with the *Agena* target vehicle. These successes, along with the objective to favor manned space flight, towards the goal of going to the Moon, marked heavily the automated capabilities of United States (US) spacecrafts. At least until the last two decades as it will be described further on (Woffinden and Geller, 2007).

Russians, on the other hand, pursued from the start an automated approach to the space flight, relegating the onboard crew to monitoring the operations and intervening only in cases of emergency. This has led to a first-ever automated rendezvous and docking (RVD) between two unmanned, robotic spacecrafts named *Kosmos 186* (chaser) and *Kosmos 188* (target), on October 30, 1967. The automated rendezvous system responsible for this success was the Igla radar system. The success of this first mission was then repeated multiple times and in 1968 Russia finally confirmed its path towards automation in space and the building of their space station as a stepping-stone towards deep space exploration. A more advanced Russian automated spacecraft, used even today to ferry cargo to the International Space Station (ISS), is the *Progress* vehicle. It was introduced in 1978 and is equipped with the Kurs rendezvous radar system. This system is still considered to be the current standard of automatic rendezvous systems despite its weight and power consumption (Woffinden and Geller, 2007; Nolet and Miller, 2007).

To overcome the cumbersome and aging design of previous automatic navigation systems, recent experimental missions have been performed mainly by the Japan and US authorities towards the goal of autonomous close proximity operations. The first mission in line is the Japanese *Engineering Test Satellite* (ETS)-*VII*. It was launched in November 1998 and developed by the National Space Development Agency of Japan (NASDA, currently JAXA) as a demonstration mission of some of the technologies for the H-II Transfer Vehicle (HTV), in particular of advanced ARVD techniques and unmanned orbital operations. It was the first-ever mission with an unmanned spacecraft having a robotic manipulator onboard and the first to perform an ARVD between unmanned spacecrafts. The space segment of the mission consisted of two spacecrafts, the chaser, named Hikoboshi, and the target, named Orihime (Woffinden and Geller, 2007; Nolet and Miller, 2007; Yoshida and Wilcox, 2008). To date, it can be considered as "the most complex successful technological demonstration of a service mission" (Hirzinger et al., 2009). However, the target spacecraft was cooperative and even then the mission experienced an attitude anomaly during one of ARVD maneuvers. This, forced the ground control to reconfigure the Rendezvous Flight Software (RVFS) to recover from the anomaly and accomplish the task (Nolet and Miller, 2007).

The *Experimental Satellite System-10* (XSS-10) was the first US mission to demonstrate basic autonomous proximity operations capabilities around a resident space object[8] (RSO). Particularly, the mission objectives were to perform: an autonomous navigation around an RSO on a preplanned course, semi-autonomous proximity operation maneuvers and an inspection of the RSO. The 31 kg spacecraft was developed by the US Air Force Research Laboratory (AFRL) and the chosen RSO was the Delta II stage that released the spacecraft into the orbit. The mission was performed in 2003. All primary mission objectives were met although minor problems were encountered during the mission. The most relevant was the connection dropout with the satellite during its closest approach to the RSO. This way the closest distance to it could not be measured and the close-in images of the target could not be downloaded (Davis et al., 2003; Nolet and Miller, 2007).

The *Experimental Satellite System-11* (XSS-11) was the successor of the previously mentioned spacecraft. It was developed by the Lockheed Martin Space Systems and commissioned by the US AFRL. It was launched in 2005

[8] Intended as either an active or inactive/decommissioned spacecraft/vehicle in orbit.

with the objective to verify the GNC system for a safe and autonomous rendezvous and close proximity operations with multiple space objects. The spacecraft was a microsatellite class vehicle having around 100/145 kg of dry/wet mass. It was equipped with a scanning light detection and ranging (LIDAR) sensor for relative range and angle measurements. The spacecraft was planned to perform maneuvers in complete autonomy by relying on its onboard planner. By the fall of 2005, the spacecraft had successfully performed more than 20 rendezvous maneuvers with its Minotaur 4th stage rocket body and several other close proximity operations (Woffinden and Geller, 2007). The nominal duration of the mission was stated to be 12-18 months with subsequent de-orbiting of the spacecraft, but according to the EoPortal (2007) the spacecraft was still in orbit on February 2007. To our knowledge further information about the mission was not made public.

The National Aeronautics and Space Administration (NASA) agency launched its own ARV mission, the *Demonstration of Autonomous Rendezvous Technology* (DART) just few days after the launch of XSS-11, on April 15th, 2005. The objective of the mission was to demonstrate the US capability of completely autonomous rendezvous. The mission was slated to last only 24 h, during which the DART spacecraft had to autonomously track and rendezvous, within 5 m, with the specially designed target vehicle, the *Multiple Paths Beyond-Line-of-Sight Communication* (MUBLCOM) satellite. The relative position and orientation was to be determined with advanced video guidance sensor (AVGS). Also in this case the target was cooperative. After the successful orbit insertion and first phases of the rendezvous, the mission failed about 11 h into the mission, due to navigation errors and consequent excessive usage of the fuel. The DART spacecraft eventually collided with the MUBLCOM satellite without even the spacecraft being aware of the collision, given that the AVGS sensor never came into the usage (Woffinden and Geller, 2007).

Another relevant US mission in line is the *Orbital Express* (OE) developed by the US Defense Advanced Research Projects Agency (DARPA) and launched in March 2007. The duration of the mission was 90 days during which the OE needed to demonstrate several key technologies intended to validate the capabilities of autonomous approach, rendezvous, capture and on-orbit servicing (OOS) of a target spacecraft by means of a robotic manipulator. The space segment consisted of two spacecrafts: a servicing satellite, the *Autonomous Space Transport & Robotic Operations* (ASTRO) vehicle equipped with a 3 m long manipulator and a satellite being serviced, a prototype of a modular *Next Generation serviceable Satellite* (NEXTSat). Unlike the ETS-VII mission performed 10 years before, OE had to demonstrate a higher degree of autonomy in all tasks. ASTRO was equipped with several different navigation sensors and imaging software that enabled observation of the target regardless of lighting conditions, range and background (Woffinden and Geller, 2007; Nolet and Miller, 2007; Yoshida, 2009). The mission was successful although the servicer did experience some anomalies, one of which even threatened to end the mission at the day one. The anomaly was related to the flight software that commanded the reaction wheel "backwards" thus preventing the system from achieving a safe sun-pointing attitude. The situation was promptly discovered and solved with a software update issued by the ground control. Another anomaly worth mentioning was the primary sensor computer central processing unit (CPU) fault that ASTRO encountered during the 30-meter ARV scenario. The anomaly triggered an abort command and it took the ground control 8 days to solve the problem (Defense Industry Daily, 2007; Kennedy, 2008; Wright, 2011).

Based on the mentioned missions it is therefore possible to note that almost every mission did experience some sort of malfunction that required a promptly intervention from the ground control. This underlines that autonomous rendezvous and proximity operations without humans in the loop is not yet mature enough (Pavone and Starek, 2014). Thus, much work needs still to be done to raise the technological readiness level that will eventually enable routine ARVD.

One of the future missions planning on raising this technological level is the *Deutsche Orbitale Servicing* Mission (DEOS). The mission is currently in the definition phase and it is being developed by DLR and Airbus Defence and Space (as a prime contractor). According to current information, the mission should be ready for launch in 2018 (Airbus Defence & Space, 2012). The main mission objective is the in-orbit demonstration of technologies and techniques needed for unmanned autonomous and tele-operated on-orbit servicing of an uncooperative target. In particular, the mission will demonstrate all different phases of an autonomous rendezvous and docking/capture (ARVD/C) mission with increasing complexity (Rupp et al., 2009). The servicing spacecraft will have a 3 m robotic manipulator with 7 degrees of freedom (DOF), a docking and berthing mechanism. The client spacecraft should exhibit a grappling fixture and also a docking and berthing mechanism. The client will be designed to perform different attitude maneuvers in order to simulate a behavior of a non-cooperative, tumbling client satellite (Sellmaier et al., 2010).

2.2. State of the art of RV control architectures

The standard control architecture traditionally used for automated rendezvous and docking of vehicles, such as the Automated Transfer Vehicle (ATV), HTV or Progress vehicle, has been illustrated by Fehse (2003) in his book entitled *Automated Rendezvous and Docking of Spacecraft*. The architecture is divided in several modules interconnected between them showing simply the levels of authority. Those modules are[9]: the automatic failure detection, isolation and recovery (FDIR), the automatic mission and vehicle management (MVM) and the GNC. The ground control, as expected, plays in this architecture an important part given that it only has the authority to perform collision avoidance maneuvers (CAM) and impart commands to the rendezvous control system. Nevertheless, capturing a non-cooperative, tumbling target could require some degree of autonomy which is the motivation of the presented research.

Nolet and Miller (2007) presented a control architecture developed for a nanosatellite platform SPHERES[10] to demonstrate a series of autonomous docking and formation flight experiments onboard the ISS. The presented architecture is an extended version of Fehse's. It takes into account an autonomous approach and thus grants the onboard computer the authority and capability to perform decisions and in particular to perform a CAM through the FDIR module in case of anomalies. Moreover, in this case the communication with the ground control is assumed to be intermittent or even nonexistent. However, Nolet and Miller (2007) consider that the target vehicle is able to communicate its states to the chaser while tumbling. Our assumption is that the target is not only tumbling but is also non-cooperative meaning that the chaser has to estimate on its own the relative position and attitude of the target prior to its capture. Moreover, Nolet and Miller (2007) consider only the docking scenario while we tackle the capture and manipulation of the target by means of a manipulator. Furthermore, our architecture should eventually include also some of the state-of-the-art GNC algorithms that are missing in the one developed by Nolet and Miller (2007). Nevertheless, given its proven and validated design, we have considered it as a basis for our own GNC architecture.

[9] Presented hereafter in order of hierarchy in a control system.

[10] Url: http://ssl.mit.edu/spheres/index.html.

More recently, Sommer and Ahrns (2013) presented a GNC concept for rendezvous and capture, by means of a lightweight manipulator, of a small spacecraft. The methodology of their research relies heavily on the consolidated experience of the ATV thus excluding some of the cutting edge algorithms and techniques. For example, the relative pose estimation is done by using a template matching technique, an iterative closest point algorithm (ICP) and a Kalman filter. The control of the attitude and position in close range is done through a configurable proportional-integral-derivative (PID) controller. In far/mid range the pose control is done simply by comparison of the reference and actual states of the spacecraft. No information is given regarding the guidance algorithm used. The role of the ground control is not explicitly mentioned in the research although it should be expected to be similar to the one of an ATV mission.

2.3. State of the art of GNC algorithms

The idea of an unmanned robotic spacecraft capable of capturing and servicing other malfunctioning spacecrafts dates back in early 1980s after the successful usage of the Space Shuttle remote manipulator system, in the STS-2 mission. Several manned on-orbit servicing missions followed to repair and deploy malfunctioning satellites (such as Anik-B, Intelsat 6 and Hubble telescope), but a completely autonomous, unmanned mission has yet to become reality despite the demonstration missions mentioned at the beginning of the section (Yoshida and Wilcox, 2008). Nevertheless, there has been over the years a tremendous amount of theoretical research dealing with individual areas of ARVD/C missions, especially in the context of guidance, navigation and control. Flores-Abad et al. (2014) have provided an exhaustive review of space robotics technologies for on-orbit servicing. Based on their work we present hereafter some of the state of the art research in the navigation and guidance fields. A description of some of the state-of-the-art control algorithms follows.

Starting with the algorithms for the estimation of the pose of a target, Hillenbrand and Lampariello (2005) proposed a method for estimating not only the pose and angular velocity of a free-floating target but also its center of mass and inertia tensor by using range data and a least square method. Tzschichholz et al. (2011) presented an algorithm for spacecraft pose estimation and motion prediction based on rotation- and scale-invariant features using a photonic mixer device (PMD) camera. Aghili et al. (2011) made a study of a fault-tolerant method for pose estimation of space objects using

Neptec's Laser Camera System (LCS), Kalman filter (KF) and an iterative closest point (ICP) algorithm in a closed-loop configuration.

Regarding guidance techniques for proximity operations, Flores-Abad et al. (2014) mentions the following works: Matsumoto et al. (2003) proposed two methods for safe approach to an uncontrolled rotating spacecraft: a passive fly-by and an optimized trajectory. Ma et al. (2012) optimized the approach trajectory of a chaser to a tumbling target such that the relative motion between the two is zero by minimizing the approach time and fuel. Pontryagin's Maximum Principle was used for the optimization process. A method using mixed-integer linear programing (MILP) or alternatively only linear programming (LP) for generating on-line fuel-efficient and safe trajectories was developed by Breger and How (2008).

Concerning the control algorithms, there have been a wide variety of researchers tackling both linear and nonlinear control methodologies. Luo et al. (2014) have provided in their research paper a good overview of current modern control methods based on fuzzy logic (Karr and Freeman, 1997), neural networks (Youmans and Lutze, 1998) and simulated annealing algorithms (Luo and Tang, 2005). The State-dependent Riccati equation (SDRE) approach recently seems to attract a lot of research in this field as proven by the works of Çimen (2010); Lee and Pernicka (2010); Di Mauro (2013). The Linear quadratic tracking controller (LQT) and linear quadratic regulator (LQR), based on linear systems, have been studied and proposed by Lee and Pernicka (2010) and Arantes and Martins-Filho (2014).

3. Reference mission

The reference mission selected for this research is a robotic ADR mission aiming at capturing and de-orbiting several targets, all of the same type. This approach has several advantages over the single object mission. Namely, the reduced research and development effort (R&D) for the whole mission and the overall cost.

Following is a more detailed description of the selected target object, the robotic chaser spacecraft and the overall mission profile. The description is more focused on proximity operations since our aim is just to give a context to the architecture that is described in the next section.

3.1. Target object

According to the US Space Track catalog[11] there are in orbit, at the time of writing[12], 3974 intact payloads, 1998 intact rocket bodies and 11157 tracked space debris objects. If the objective of future ADR missions is to stabilize the space debris environment by limiting the number of fragments arising from accidental collisions, targets of those missions should be mainly large intact objects from the most crowded regions. This means, to focus the ADR efforts towards targets that exhibit the highest product of collision probability and mass. The objects on highly eccentric GEO-transfer orbits should however be excluded from this metrics given their limited presence in LEO (Liou, 2011a). According to the above mentioned ranking method, (Liou, 2011a) has identified the top 500 targets that should be first tackled by any future debris removal mission in order to stabilize the space debris environment. The prograde region, and in particular the $h = 950 \pm 100\,\mathrm{km}$, $i = 82 \pm 1\,\mathrm{deg}$[13] band is especially interesting due to the fact that it is dominated mainly by several well-known RSOs (e. g. SL-3 R/Bs-Vostok second stages, SL-8 R/Bs-Kosmos 3M second stages, SL-16 R/Bs-Zenit second stages, etc.) (Liou, 2011a; DeLuca et al., 2013).

Additional issues that need to be taken into consideration during the selection of the target are the issues of legal nature. In fact, according to the international law, the launching state retains the jurisdiction of the launched object perpetually. Thus, to remove an RSO, an approval from its legal owner is needed (DeLuca et al., 2013).

In general, rocket bodies are considered to be less confidential than spacecrafts which is why their removal should pose less legal problems. Furthermore, due to their design, they are considered to be sturdier. Moreover, they generally do not posses appendages and their attitude motion in LEO can be expected to be stable, with low angular rates[14], after only few years in space (Praly et al., 2012).

All those considerations have led us to consider the Russian Kosmos 3M second stages (see Figure 3[15]) as the target objects of our research. Around 300 are currently present in orbit (DeLuca et al., 2013).

[11] Url: https://www.space-track.org.

[12] August 26th, 2014.

[13] h indicates the orbital altitude and i its inclination.

[14] Considered to be according to the literature few degrees per second.

[15] Source url: http://goo.gl/sUTltm

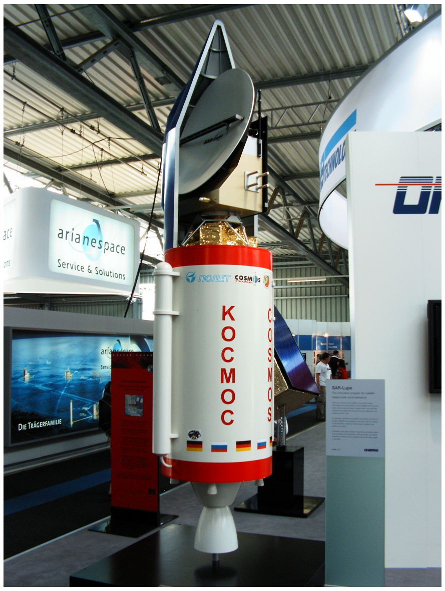

Figure 3: Mockup of a Kosmos 3M second stage with a model of a German SAR-Lupe reconnaissance satellite mounted on top of it (source: Wikimedia)

One particular R/B was chosen as the initial target of the reference mission due to its orbital characteristics and the date of the launch[16]. The selected R/B is identified in the US Space Track catalog[17] with the ID 1975-074B and its essential orbital parameters are listed hereafter[18]:

- launch date: 14.08.1975
- period: 104.66 min
- inclination: 82.9 deg
- apogee: 995 km
- perigee: 962 km
- eccentricity: 0.0022

[16] This body was chosen only to define a representative mission and may change based on future studies.

[17] Url: https://www.space-track.org.

[18] Obtained at the orbit epoch of August 23, 2014.

The physical properties of this vehicle are:

- diameter: 2.4 m
- length: 6.5 m
- dry mass: 1400 kg

The authors were not able to retrieve any data concerning the attitude motion of the chosen target. However, according to studies performed by Praly et al. (2012), it is acceptable to consider that given its age, its angular rate should be low and in any case no more than few degrees per second.

3.2. Chaser spacecraft

The chaser spacecraft is a robotic system, similar to the one described by Castronuovo (2011), carrying onboard four de-orbiting kits, a suite of sensors for ARVD/C and two robotic manipulators. One will be used for the capture of the target while the other will be used for the attachment of a de-orbiting kit to it. The complete system architecture is out of the scope of the present paper thus, the general overview of the chaser system will be given for the sake of completeness.

The choice of de-orbiting kits and in particular of hybrid propulsion modules (HPM), such as those described by DeLuca et al. (2013), was made based on the requirements of a controlled reentry and very high levels of thrust needed for de-orbiting the target. Hybrid propulsion modules were a specific choice given their compact design, high specific impulse, throttling and re-ignition capabilities. The latter two characteristics are in particular the advantage of such modules over the ones based on the solid state propellant. Their disadvantage is a lack of space experience that however should be overcome in coming years given the potential and benefits they showed over the last decades.

The high level system architecture of a chaser for an ADR mission is a difficult trade-off, given the number of variables that need to be taken into account in the optimization process. This difficulty is underlined in a paper written by Bonnal et al. (2013) focusing on recent progress and trends for the ADR. The authors stress that currently there is a confusion on how an optimal system architecture of a chaser should look like. Nevertheless, they have identified as a most promising solution a 4-5 t chaser spacecraft carrying at least five de-orbiting kits that should be launched on an Ariane 5 class launch vehicle along with four other identical spacecrafts, each targeting contemporaneously different orbital regions (Bonnal et al., 2013). Based on the above mentioned result, we assume a chaser spacecraft of the same class whose characteristics and exact configuration are to be defined in a future research. Nonetheless, the feasibility of the usage of robotic manipulators for the envisioned mission should be out of question as outlined by recent papers (Castronuovo, 2011; Bonnal et al., 2013).

The capture of the target is assumed to be performed using only one manipulator having as an end-effector a capture mechanism that employs directional (Hawkes et al., 2013) or electro adhesives (Tellez et al., 2011; DeLuca et al., 2013). The use of these capture mechanisms allows relaxing the requirements on the tracking of a capture point, since no specific feature on the target has to be grasped.

The spacecraft itself is supposed to be a three axis stabilized vehicle able to perform orbital maneuvers, by means of a cluster of bi-propellant main engines, i. e. orbit control thrusters (OCT). Other actuators of the spacecraft control system (SCS) are a cluster of reaction wheels (RWs), a three-axis magnetic torquer (for desatuartion of RWs), and bi-propellant attitude control thrusters (ACT). The latter are supposed to be ON/OFF thrusters and to have a thruster level of around 200 N. RWs are taken into account in order to reduce fuel requirements given the expected long duration of the mission and dynamical coupling that will occur between the manipulators and the base spacecraft. Note however that in order to make the GNC architecture as generic as possible, the SCS is considered as a black box capable of controlling the spacecraft in all 6 DOF. Thus, specific algorithms for controlling the SCS hardware components will be neglected at the time of writing. This restriction will be evaluated further on and eventually removed from the GNC architecture.

Similarly to the configuration defined by Sommer and Ahrns (2013) in their concept, the sensors system of the spacecraft is composed of the typical suite of sensors for an attitude determination and control system (ADCS), plus a suite of sensors needed for the relative navigation purposes. The first suite of sensors is imagined to be composed of a coarse Earth/Sun sensor, a magnetometer, star sensors, gyroscopes and GPS receivers. The suite of sensors for the ARVD/C is envisioned to be composed instead of: an infrared

(IR) camera and an optical camera for the far range phase[19] (i. e. for distances from $5 - 1\,\mathrm{km}$) of the RV, of a scanning LIDAR for the close range approach (i. e. for distances from $1\,\mathrm{km} - 50\,\mathrm{m}$) of the RV and for the pose estimation (in 3D mode and distances from $50 - 3\,\mathrm{m}$) of the target and finally of a stereo camera system for the capturing and manipulation phases.

3.3. Mission scenario

In our mission scenario a single chaser spacecraft is expected to autonomously rendezvous and capture, in sequence, four Kosmos 3M rocket bodies and deorbit them. In order to perform this task it needs to successfully accomplish several major mission phases. Those phases are (Fehse, 2003): launch, phasing, far range rendezvous or homing, close range rendezvous and mating (or more specifically capture in our case). This paper focuses on the last two phases, but, for the sake of completeness, the description of the whole mission scenario is presented in what follows.

3.3.1. Phasing with the target

After the launch, the chaser is assumed to be injected into an initial near circular orbit that is in the same orbital plane as the one of the target, but is lower in altitude, as illustrated in Figure 4.

During this initial phase, the chaser will be placed few tens of kilometers below and behind the target. This way the chaser will be in an orbit well below the sphere of uncertainty of approximately 1-2 km in diameter, assumed to be surrounding the target (DeLuca et al., 2013). At this point, after the successful initialization of the chaser spacecraft, the phasing maneuver is initiated in order to reduce the phasing angle between the two vehicles. To achieve this, the altitude of the chaser's orbit is gradually raised until the rendezvous gate, which is assumed to be around 3 km below and 5 km behind the target (see Figure 5).

These steps, visible in Figure 5, consist in a series of Hohmann transfers and drift times. This approach offers several advantages over a direct injection into the target's orbit. First, the passive collision avoidance safety is guaranteed at all time given that, even in the case of chaser's complete control inability, the spacecraft will only drift below the target indefinitely. Second, the Hohmann transfers are generally the most fuel efficient orbital transfers

[19] Various phases of the mission are described in the next subsection.

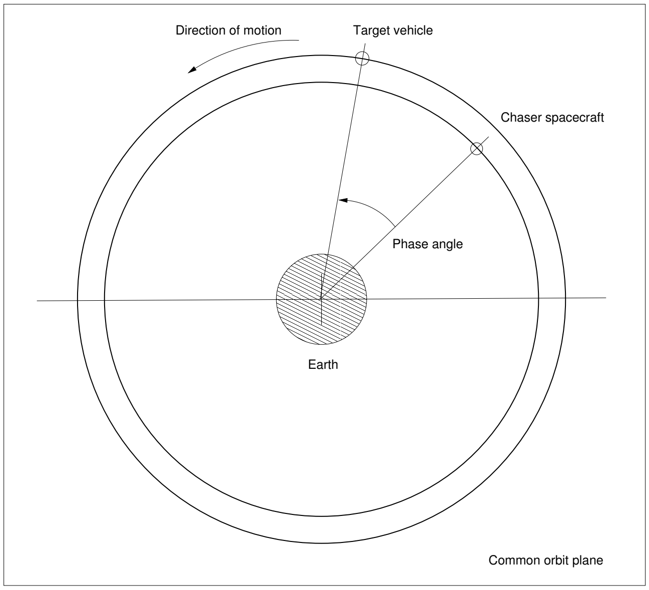

Figure 4: Initial mission scenario

in LEO, which makes this approach very fuel efficient. Third, the timing of Hohmann maneuvers and the duration of drift times can be appropriately tuned to meet specific mission requirements (Barbee et al., 2010).

Absolute navigation sensors (such as the GPS receivers) are generally used in this first rendezvous phase. Autonomy is not needed in this phase since the commands are generally sent directly from the ground control.

3.3.2. Far range rendezvous

After reaching the rendezvous gate (see Figure 5), the far range rendezvous phase is preformed to bring the chaser in the immediate vicinity of the target and create the conditions for close range rendezvous or final approach. This phase consists respectively of a homing and closing rendezvous maneuvers, as illustrated in Figure 6 and Figure 7. In these two phases, only relative navigation is performed using the onboard sensors such as an optical or IR camera and/or a LIDAR. To switch between those sensors at least one intermediate hold point (at a distance of approximately 250-100 m) is necessary.

At the start of the homing maneuver, the bearing angles (azimuth and elevation) are the most important parameters, but, as the distance between the two reduces, the relative distance and velocity gain more and more prominence (Castronuovo, 2011). It is worth noting that technologies and tech-

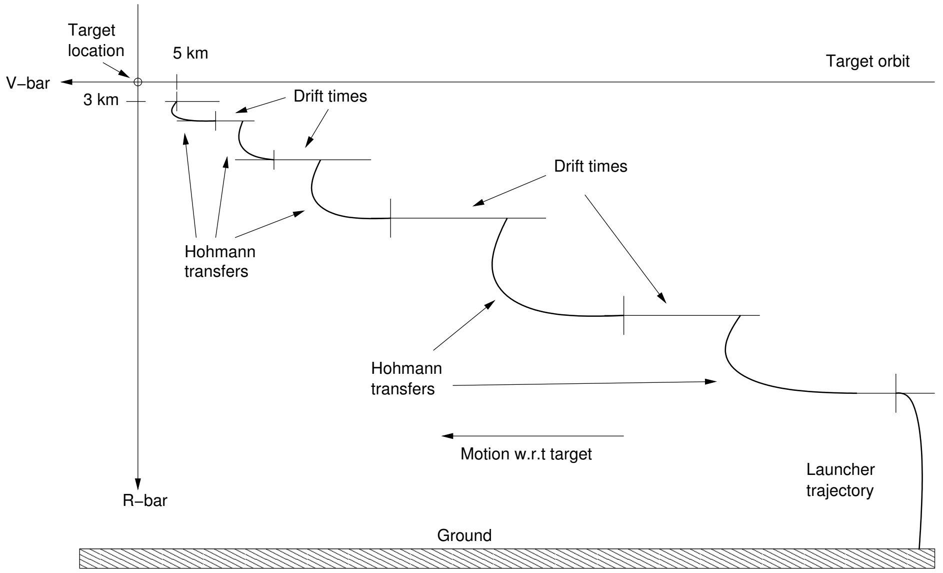

Figure 5: Phasing strategy

niques for the far range rendezvous phase of the mission are not considered critical given that they have already been proven by several past missions (see Subsection 2.1) (DeLuca et al., 2013).

Two possible profiles for the far range rendezvous phase have been identified in this paper as the most suitable for this kind of mission, given their inherent passive safety and, to some extent, fuel-efficiency.

The first one, illustrated in Figure 6, is similar to the approach strategy of the European ATV. It consists at first of a Hohmann transfer, to bring the chaser at the same altitude of the target, but, around 1 km behind it (P_1 in Figure 6). From this point a series of radial boost transfers, with waiting (station keeping) points, follows, to place the chaser in the immediate proximity of the target, about 50 m from it (P_2 in Figure 6). The waiting points are to be used for the switch-over of navigation sensors and re-evaluation of the relative distance between the two objects.

When the chaser is at point P_2, the first pose estimation of the target's motion is performed. A fly around maneuver[20], using a radial boost, is preformed for inspection of the target. Finally, the chaser returns to the initial

[20] Or multiple ones if necessary.

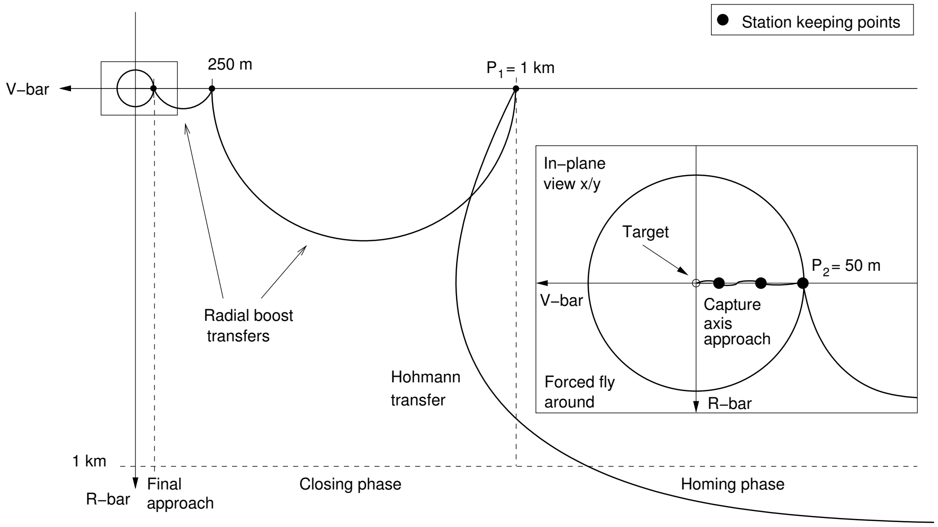

Figure 6: First far/close range approach strategy

holding point (P_2 in Figure 6) where it performs another pose estimation of the target, before starting the final approach.

The second method, illustrated in Figure 7, consists instead of two Hohmann transfers and some drift times that bring the chaser 250 m below and behind the target (P_1 in Figure 7). The advantages of this method are the same described in the phasing strategy. In this approach, the drift times are used for the switch-over of navigation sensors and and re-evaluation of the relative distance between the two objects, just as the waiting points were used in the first strategy. The schedule of the Hohmann transfers and the amount of drift times can be appropriately tuned to ensure the convergence and accuracy of the navigation filter (Barbee et al., 2010).

From point P_1, a free drift of the chaser is allowed until the in-track distance between the two is zero (P_2 in Figure 7). During this phase a final estimation of the target's position is performed from below. An inspection of the target vehicle could also be performed given the relative vicinity of the two vehicles.

Once that the position has been estimated and that the in-track distance is zero, a maneuver using Clohessy-Wiltshire targeting[21] (C-W targeting) is

[21] Trajectory control based on the initial and final relative states derived from the Clohessy-Wiltshire equations (Luo et al., 2014).

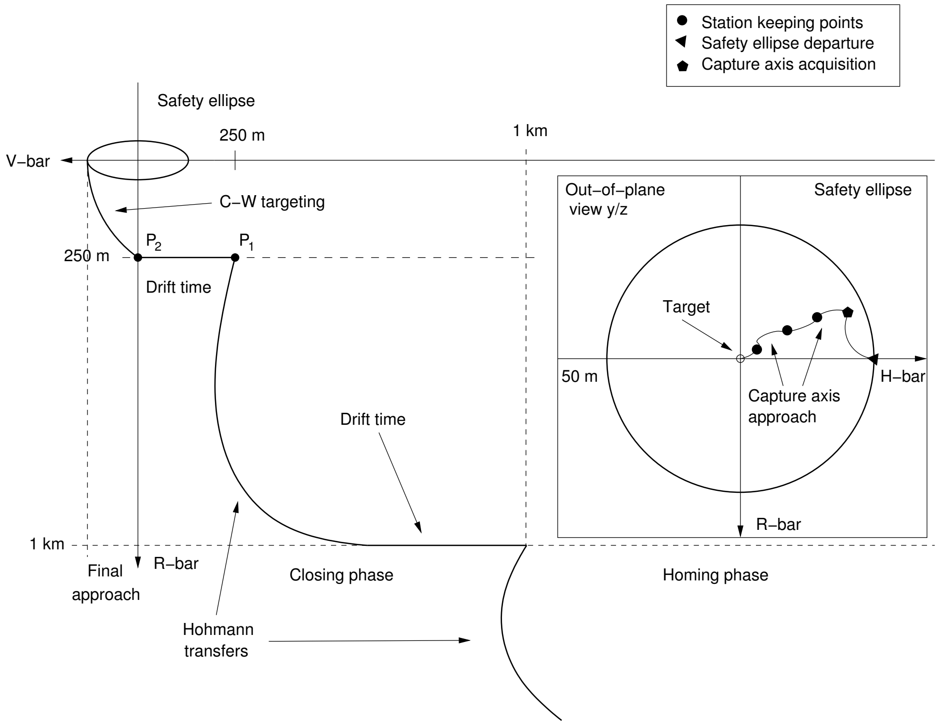

Figure 7: Second far/close range approach strategy

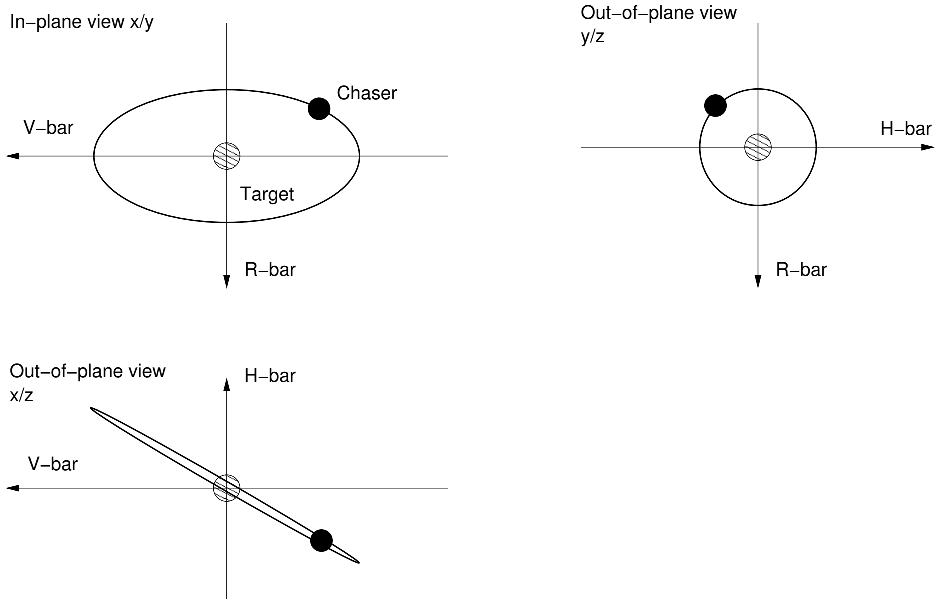

Figure 8: Safety ellipse

performed. It places the chaser into a safety ellipse (SE), ahead of and out of plane of the target. The result of this maneuver, illustrated in Figure 8, is an out of plane relative elliptical trajectory of the chaser around the target, that is is fixed with respect to the target and never crosses its V-bar[22] (see Figure 8) (Barbee et al., 2010). The projection of the ellipse on to the radial, cross-track plane (R-bar/H-bar or y/z plane visible in Figure 8) should be a circle of 50 m, thus guaranteeing the minimum distance between the two vehicles. The advantage of this approach lies in the possibility to appropriately design the SE to reach desirable illumination conditions required for the inspection, while guaranteeing at the same time the passive safety of the trajectory. The pose estimation of the target is to be performed while the chaser moves on the SE.

3.3.3. Close range rendezvous

In both previously mentioned far range rendezvous strategies, the final approach phase begins with the acquisition of the capture axis (see the il-

[22] Defined as the horizontal axis of the local-vertical/local-horizontal (LVLH) reference frame.

lustration on the right of Figure 6 and Figure 7). The approach trajectory will vary according to the closing method chosen and the requirements of the robotic capture mechanisms. However, in any case it shall guarantee passive safety and to some extent fuel efficiency. The capture axis will generally be the main axis of rotation of the target body.

Independently of the previous approach strategy, once the capture axis has been reached, the maneuver will consist either in: a) a straight line trajectory, consisting of a series of hold points and constant rate motion within a predefined corridor (illustrated in Figure 6 and Figure 7) or b) in an optimized trajectory that limits as much as possible the active safety requirement and is fuel-efficient.

The final selection of one of the two depends greatly upon the requirements of the robotic capture mechanism that will be defined in future studies. Nevertheless, in both cases the capture approach lasts until the berthing box[23] is reached or the conditions for the capture are met. In case something goes wrong a CAM is to be performed autonomously by the chaser. It is paramount that the autonomous pose estimation of the target is constantly updated during the capture phase to know at every moment the exact[24] relative position and attitude of the target.

Once the berthing box has been reached the chaser needs to actively synchronize, within the required boundaries of the capture mechanism, its attitude motion with that of the target. Moreover, the chaser needs to actively maintain its position within the moving berthing box given the natural drifts that would otherwise occur in just few minutes.

3.3.4. The capture and manipulation

Finally, the chaser deploys its robotic capturing mechanism and captures the target. After the attenuation of the shock and residual velocities, the rigid connection between the two spacecrafts is achieved. Transferred angular momentum, from the target to the chaser, is dissipated and the compound is stabilized. At this point the second manipulator will detach an HPM de-

[23] Defined essentially as a volume within which the chaser must stay in order to create conditions necessary for the capture of the target vehicle. For a more detailed description please refer to (Fehse, 2003)

[24] Within the limits of the pose estimation sensor precision.

orbiting kit[25] from the chaser and firmly attach it to the target. A preferable attachment position is the main engine of the R/B, given its mechanical properties and alignment with the center of mass of the R/B. The envisioned attachment could use either an expandable umbrella mechanism (Castronuovo, 2011) or the so called corkscrew system (DeLuca et al., 2013) or even a clamp mechanism that would rigidly secure the de-orbiting kit to the main engine of the R/B.

Afterwards, the chaser will reorient the composite system in the right direction and retreat itself to a safe location while the de-orbiting maneuver is performed. Subsequently, the chaser is free to perform the described sequence again, in order to reach and de-orbit the next object in the sequence.

It is worth noting that the description of the chaser's robotic capture mechanism is intentionally vague given that its definition will be scope of our future studies.

4. GNC concept

The guidance, navigation and control system has to: a) process the information coming from sensors, b) plan the execution of appropriate maneuvers and c) perform them. Based on the following, the GNC architecture has been defined in this paper as "an abstract description of the entities of a GNC system and the relationship between those entities" (Nolet and Miller, 2007).

Modularity is seen as one of the key feature of this architecture given that it is envisioned to be built using the open source Robot Construction Kit (ROCK)[26] software framework specifically developed for robotic systems and with modularity in mind. The ROCK provides a wide variety of tools necessary to develop and test robotic systems for many applications. Particularly, it contains a multitude of ready to use drivers and modules, and can be easily extended adding new components, facilitating the development of the GNC architecture and its implementation onto the robotic hardware, for testing purposes.

Moreover, its open source nature is seen as another advantage, since the developed architecture should be easily accessible and modifiable by the sci-

[25] The definition of the HPM de-orbiting kit is out of the scope of the current paper so for further information please refer to the research performed by DeLuca et al. (2013).

[26] Url: http://rock-robotics.org/stable/index.html.

entific community. Development of the architecture in other platforms is not excluded, given the early stage of the research, but the ROCK framework is at the time of writing the chosen platform.

Figure 9 illustrates the envisioned architecture. It consists of several software modules each responsible for a particular function within the GNC system. As in the research of Nolet and Miller (2007), each software module is a set of algorithms capable of executing a particular task. Those modules and their principal tasks are:

1. navigation module: performing the pose estimation of the target
2. guidance module: performing trajectory planning towards the capture axis and ultimately towards the target with safety and fuel-efficiency in mind
3. control module: performing the execution of maneuvers according to the guidance function and suppression of external disturbances

Their more in depth description is illustrated further on in this section. Particular attention is devoted to the individualization and characterization of algorithms chosen for populating relative modules. Their quantitative evaluation and modes of implementation within the GNC architecture is left for a future research.

For the sake of completeness, the robotics module, although not explicitly part of the GNC architecture[27] (see Figure 9) will be described only briefly in this section, due to the initial stadium of our research in this area. A future paper will be dedicated entirely to this topic and its integration with the spacecraft's GNC architecture.

The MVM and FDIR modules (visible in Figure 9) are not considered at this stage of research although we are well aware of their importance in an autonomous system like this one. This is especially true in the last few meters of the close approach, when CAM capabilities of the chaser spacecraft are usually a requirement.

It is worth noting that the architecture is built having in mind the most critical phase of an ADR mission which is the close range rendezvous. The target of the mission, as described in previous sections, is assumed to be an uncooperative but known (in shape and approximate attitude) Kosmos 3M R/B.

[27] For an explanation of this statement please see Subsection 4.4 on page 41.

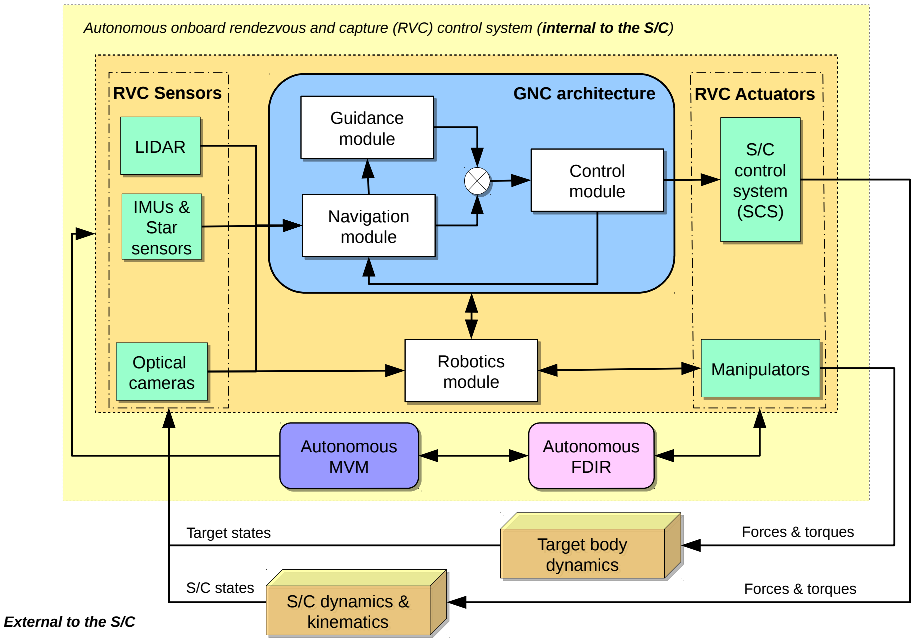

Figure 9: Autonomous onboard rendezvous and capture control system architecture

4.1. Navigation module

The task of the navigation module is to use a filter to: a) process the information about the states of the chaser and target vehicles[28] and b) propagate this information in time by using the model of the spacecrafts' dynamics and information about the imparted commands. This information is then made available to the control and guidance modules for further processing (Fehse, 2003).

The sensors used by this module are: a LIDAR (in 3D mode), to generate a 3D point cloud of the target, and inertial measurements units (IMUs) (i. e. gyros and/or magnetometers), to eliminate the ambiguity between the pure rotation and translation (Kervendal et al., 2013)[29].

The reason for choosing a LIDAR is that this active sensor has already been used successfully in space. Moreover, it is relatively insensitive to illumination conditions, and it is usable over a wide range of distances. Moreover, a working unit is present, as of time of writing, in DFKI's facilities, which means that it could be used for real testing of developed algorithms. The disadvantages of using such a sensor are the power consumption and the required minimum distance between the chaser and the target. Thus, these issues must be taken into account when defining the required onboard power and the characteristics of the robotic capture mechanism.

Regarding the filter algorithms, a thorough survey of possible nonlinear attitude estimation algorithms has been provided by Crassidis et al. (2007). Based on the literature research, we have selected for this module the extended Kalman filter (EKF) and the unscented Kalman filter (UKF).

The desirable features of such an algorithm are: fast convergence, robustness and stability in the whole state space of the mission (Nolet and Miller, 2007).

The EKF has been used quite extensively in the aerospace industry in the last few decades, given the right balance it offers between the computational requirements and performance. Moreover, it has the same mathematical scheme of the traditional KF, but, it has the advantage of being used in nonlinear systems, such as ours. All this made it a good candidate as the baseline filter technique. However, the convergence and robustness of the

[28] Coming from different sensors.

[29] Eventual use of other sensors like star trackers and/or Earth/Sun sensors should eventually be taken into account for updating the measurements of gyros due to their inevitable drift over time.

algorithm are not a priori guaranteed as in case of the KF. Additionally, the need to calculate the Jacobian functions might prove difficult, if the functions of the dynamics or measurements prove not to be differentiable (Nolet and Miller, 2007).

Hereafter, we present a generic EKF algorithm omitting some theoretical considerations. More detail on the presented algorithm can be found in (Wan and van der Merwe, 2002).

The basic idea behind the EKF is to estimate the state of a discrete-time nonlinear dynamic system that can be described with (Wan and van der Merwe, 2002; LaViola, 2003):

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{1}$$

$$\mathbf{y}_k = \mathbf{H}(\mathbf{x}_k, \mathbf{n}_k) \tag{2}$$

where $\mathbf{x}_k$ is the unobserved state of the system, $\mathbf{u}_k$ is a known control input vector, $\mathbf{y}_k$ is the observed measurement signal, $\mathbf{v}_k$ is the process noise and $\mathbf{n}_k$ is the observation noise. The system dynamic models, represented by the functions $\mathbf{F}$ and $\mathbf{H}$, are assumed to be known.

The EKF, like all Kalman filters is a recursive process which uses the dynamic model of the system to make an estimate of its current state and correct it using measurement updates. With this in mind, the explicit equations of a generic EKF algorithm are what follows (Wan and van der Merwe, 2002):

Initialize with:

$$\hat{\mathbf{x}}_0 = \mathbb{E}\left[\mathbf{x}_0\right] \tag{3}$$

$$\mathbf{P}_{\mathbf{x}_0} = \mathbb{E}\left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T\right] \tag{4}$$

For $k \in \{1, \ldots, \infty\}$, the time update equations of the extended Kalman filter are:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \bar{\mathbf{v}}) \tag{5}$$

$$\mathbf{P}_{\mathbf{x}_k}^- = \mathbf{A}_{k-1}\mathbf{P}_{\mathbf{x}_{k-1}}\mathbf{A}_{k-1}^T + \mathbf{B}_k\mathbf{R}^{\mathbf{v}}\mathbf{B}_k^T \tag{6}$$

and the measurement update equations are:

$$\mathcal{K}_k = \mathbf{P}_{\mathbf{x}_k}^{-}\mathbf{C}_k^T(\mathbf{C}_k\mathbf{P}_{\mathbf{x}_k}^{-}\mathbf{C}_k^T + \mathbf{D}_k\mathbf{R}^{\mathbf{n}}\mathbf{D}_k^T)^{-1} \tag{7}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^{-} + \mathcal{K}_k(\mathbf{y}_k - \mathbf{H}(\hat{\mathbf{x}}_k^{-}, \bar{\mathbf{n}})) \tag{8}$$

$$\mathbf{P}_{\mathbf{x}_k} = (\mathbf{I} - \mathcal{K}_k\mathbf{C}_k)\mathbf{P}_{\mathbf{x}_k}^{-} \tag{9}$$

where $\mathbf{A}_k \triangleq \frac{\partial \mathbf{F}(\mathbf{x},\mathbf{u_k},\bar{\mathbf{v}})}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_k}$, $\mathbf{B}_k \triangleq \frac{\partial \mathbf{F}(\hat{\mathbf{x}}_k^{-},\mathbf{u}_k,\mathbf{v})}{\partial \mathbf{v}} |_{\bar{\mathbf{v}}}$, $\mathbf{C}_k \triangleq \frac{\partial \mathbf{H}(\mathbf{x},\bar{\mathbf{n}})}{\partial \mathbf{x}} |_{\hat{\mathbf{x}}_k}$, $\mathbf{D}_k \triangleq \frac{\partial \mathbf{H}(\hat{\mathbf{x}}_k^{-},\mathbf{n})}{\partial \mathbf{n}} |_{\bar{\mathbf{n}}}$; $\mathbf{R}^{\mathbf{v}}$ and $\mathbf{R}^{\mathbf{n}}$ are the covariances of $\mathbf{v}_k$ and $\mathbf{n}_k$, respectively; $\mathbf{x}_0$ is the initial state of the system; $\mathbf{P}_{\mathbf{x}_0}$ is the expected initial state error; $\hat{\mathbf{x}}_k^{-}$ is the optimal prediction (i.e. prior mean) of $\mathbf{x}_k$; $\mathbf{P}_{\mathbf{x}_k}^{-}$ is the prediction of the covariance of $\mathbf{x}_k$; $\mathcal{K}_k$ represents the optimal gain term at the step k and $\mathbf{I}$ is the identity matrix. $\bar{\mathbf{n}}$ and $\bar{\mathbf{v}}$ are the values of the noise means and are equal to $\mathbb{E}[\mathbf{n}]$ and $\mathbb{E}[\mathbf{v}]$, respectively. The superscript $(-)$ indicates a value prior to a state update and $(\wedge)$ indicates an estimated value.

The UKF on the other hand has not been yet used in space, but, given that it does not require the computation of the Jacobian functions, it has the advantage of ease of implementation over the EKF. Moreover, with respect to the EKF its should present: lower error, faster convergence and higher-order expansions. Its disadvantage is that it requires as much as twice the computational load (Nolet and Miller, 2007; Crassidis et al., 2007).

The basic idea behind the UKF is to use a deterministic sampling approach to capture the mean and covariance estimates with a minimal set of points, instead of linearizing a nonlinear function using Jacobian matrices (LaViola, 2003). Hereafter, we present a generic UKF algorithm omitting some theoretical considerations, as we did in case of EKF. More detail on the presented algorithm can be found in (Wan and van der Merwe, 2002).

A generic UKF algorithm can be described by the following (Wan and van der Merwe, 2002):

Initialize with:

$$\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0] \tag{10}$$

$$\mathbf{P}_0 = \mathbb{E}\left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T\right] \tag{11}$$

$$\hat{\mathbf{x}}_0^a = \mathbb{E}[\mathbf{x}^a] = \begin{bmatrix} \hat{\mathbf{x}}_0^T & 0 & 0 \end{bmatrix}^T \tag{12}$$

$$\mathbf{P}_0^a = \mathbb{E}[(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)(\mathbf{x}_0^a - \hat{\mathbf{x}}_0^a)^T] = \begin{bmatrix} \mathbf{P}_0 & 0 & 0 \\ 0 & \mathbf{R}^{\mathbf{v}} & 0 \\ 0 & 0 & \mathbf{R}^{\mathbf{n}} \end{bmatrix} \tag{13}$$

For $k \in \{1, \ldots, \infty\}$,
Calculate sigma points:

$$\boldsymbol{X}_{k-1}^a = \left[\begin{array}{ccc} \hat{\mathbf{x}}_{k-1}^a & \hat{\mathbf{x}}_{k-1}^a + \gamma\sqrt{\mathbf{P}_{k-1}^a} & \hat{\mathbf{x}}_{k-1}^a - \gamma\sqrt{\mathbf{P}_{k-1}^a} \end{array} \right] \tag{14}$$

Time update equations are:

$$\boldsymbol{X}_{k|k-1}^x = \mathbf{F}[\boldsymbol{X}_{k-1}^x, \mathbf{u}_{k-1}, \boldsymbol{X}_{k-1}^v] \tag{15}$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{i,k|k-1}^x \tag{16}$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-][\mathcal{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_k^-]^T \tag{17}$$

$$\boldsymbol{Y}_{k|k-1} = \mathbf{H}[\boldsymbol{X}_{k|k-1}^x, \boldsymbol{X}_{k-1}^n] \tag{18}$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1} \tag{19}$$

Measurement update equations are:

$$\mathbf{P}_{\widetilde{\mathbf{y}}_k \widetilde{\mathbf{y}}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T \tag{20}$$

$$\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T \tag{21}$$

$$\mathcal{K}_k = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}_{\widetilde{\mathbf{y}}_k \widetilde{\mathbf{y}}_k}^{-1} \tag{22}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathcal{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \tag{23}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathcal{K}_k \mathbf{P}_{\widetilde{\mathbf{y}}_k \widetilde{\mathbf{y}}_k} \mathcal{K}_k^T \tag{24}$$

where $\mathbf{x}^a = \begin{bmatrix} \mathbf{x}^T & \mathbf{v}^T & \mathbf{n}^T \end{bmatrix}^T$, $\boldsymbol{X}^a = \begin{bmatrix} (\boldsymbol{X}^x)^T & (\boldsymbol{X}^v)^T & (\boldsymbol{X}^n)^T \end{bmatrix}^T$, $\gamma = \sqrt{(L+\lambda)}$; λ is a composite scaling parameter; L is a dimension augmented state; $\mathbf{R^v}$ is process noise covariance; $\mathbf{R^n}$ is measurement noise covariance; W_i are weights as expressed in (Wan and van der Merwe, 2002); $\tilde{\mathbf{y}}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k^-$, $\boldsymbol{X}$ is a matrix of $2L+1$ sigma vectors $\mathcal{X}_i$ as defined in (Wan and van der Merwe, 2002) and $\mathcal{Y}_i = f(\mathcal{X}_i)$ $i = 0, \ldots, 2L$.

In order to use the mentioned algorithms, the single pose estimation of the target needs to be calculated. This is done in our case by using the open source Point Cloud Library (PCL)[30]. The C++ library is already integrated into the ROCK framework and contains all the state of the art algorithms for 3D point cloud processing. The only hurdle that we think could appear is the significant amount of resources that such kind of library generally requires given its terrestrial nature[31]. Thus, a quantitative evaluation of the required resources of the library for our purposes is a next logical step to asses its usability on a robotic spacecraft such as ours. However, this approach if viable, would significantly speed up the implementation of the navigation module and ultimately of the whole architecture.

4.2. Guidance module

The task of the guidance module is to define, in time, a set of nominal values that will be used by the control module as a reference for the required maneuvers (Fehse, 2003). More specifically, the guidance function has to perform, based on the mission phase, the following actions (Fehse, 2003):

- calculate the execution and the duration time of boost maneuvers
- calculate the position and velocity profiles for closed loop controlled trajectories and hold points
- calculate the attitude and angular rate profiles
- calculate the instantaneous position of the center of mass of the spacecraft in the vehicle body frame based on the consumption of the propellant during the mission

[30] Url: http://pointclouds.org/.

[31] Or in other words, the library was build with terrestrial applications in mind.

In case of an ADR mission the most critical task of the guidance module is the path planning of the final approach trajectory. The criticality of this trajectory is principally given by the stringent safety requirement (in particular the passive safety requirement) that such phase of the RV involves. Nevertheless, the safety is not the only desirable feature that such trajectory should have. Other features that should be considered by the relative path planing algorithm are the propellant consumption, the robustness to perturbations, the plume impingement and the line of sight. Moreover, low computational capabilities of space qualified computers limit quite heavily the number of algorithms that could be practically applied. Thus, this additional requirement should be also taken into consideration during the research for the best possible path planing algorithm (Nolet and Miller, 2007).

Numerous methods have been developed during the years to solve this optimization problem, as it has been mentioned in Subsection 2.3. The ones we selected for this concept based on the literature research are the inbound decelerating glideslope algorithm and the MILP-based path planing algorithm developed by Breger and How (2008).

The glideslope algorithm, similarly to the EKF described in the previous subsection, has been extensively used in space for real time trajectory planning. The reason behind this lies in its simplicity, robustness and low computational requirements (Nolet and Miller, 2007; Hablani et al., 2002). Moreover, it has been successfully used for autonomous docking of SPHERES microsatellites (Nolet and Miller, 2007). Thus, the inbound decelerating glideslope algorithm was the clear choice for a baseline of our guidance module. The algorithm is especially suited for straight line approaches given that it calculates the velocity profile in the phase plane, using a finite number of thruster commands. This makes it a hybrid algorithm incorporating also a velocity control, that, if paired with another control algorithm, could be used directly for planning and executing the approach along and transverse to the capture axis. In addition, it does incorporate some sort of plume impingement feature, given that it reduces the amount of thrust towards the end of the trajectory. Nonetheless, it does not account for the propellant consumption or passive safety of the trajectory (Nolet and Miller, 2007), which, was the motivation for the selection of another more advanced algorithm that could ultimately solve the above mentioned optimization problem.

The generic mathematical expression of the inbound decelerating glideslope algorithm can be described with the following equation (Nolet and Miller, 2007):

$$\dot{\rho} = a\rho + \dot{\rho}_T \tag{25}$$

where ρ is the linear distance between the chaser and target, $\dot{\rho}$ is the approach velocity, $\dot{\rho}_T$ is the desired arrival velocity and a is the glideslope (< 0). The solution of the previous differential equations is (Nolet and Miller, 2007):

$$\rho(t) = \rho_0 e^{at} + \frac{\dot{\rho}_T}{a}(e^{at} - 1) \tag{26}$$

The total time of the maneuver that the spacecraft employs to go from a ρ_0 to 0 can be calculated with the following expression (Nolet and Miller, 2007):

$$\mathcal{T} = \frac{1}{a}\ln(\frac{\dot{\rho}_T}{\dot{\rho}_0}) \tag{27}$$

More detail on the presented algorithm can be found in (Hablani et al., 2002).

The MILP-based path planing algorithm selected as the advanced algorithm for the guidance module is the one developed by Breger and How (2008). The following is its generic formulation omitting some theoretical considerations. For more information please refer to (Breger and How, 2008).

A linearized dynamics of a chaser spacecraft being in a state $\mathbf{x}_k$, at time k can be written as (Breger and How, 2008):

$$\mathbf{x}_{k+1} = A_d\mathbf{x}_k + B_d\mathbf{u}_k \tag{28}$$

where A_d and B_d are the state transition matrix and the discrete input matrix for a single time step, and $\mathbf{u}_k$ is the input vector at the step k.

The state of the spacecraft at any future step k can be described by (Breger and How, 2008):

$$\begin{aligned}\mathbf{x}_k = A_d^k\mathbf{x}_0 + \begin{bmatrix} A_d^{k-1}B_d & A_d^{k-2}B_d & \dots & A_dB_d & B_d \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_{k-1} \end{bmatrix} \\ = B^2 + 2\left(r_g^2 + 2x_0x - x_0^2 - r_k^2\right) + \frac{\left(r_g^2 + 2x_0x - x_0^2 - r_k^2\right)^2}{B^2}\end{aligned} \tag{29}$$

where Γ_k is the discrete convolution matrix.

To solve the optimization problem the cost function that penalizes exclusively the fuel usage is used (Breger and How, 2008):

$$J = \sum_{i=0}^{N-1} \|\mathbf{u}_i\|_1 \tag{30}$$

where the 1-norm cost is used to take into account the fuel expenditure. This way the optimization problem can be formed to optimize the control input command and, at the same time, constrain the states of the system (Breger and How, 2008).

With this in mind the selected MILP algorithm consists of the optimization of the following (Breger and How, 2008):

$$J* = \min_{\mathbf{u}_0,\dots,\mathbf{u}_{N-1}} \sum_{i=0}^{N-1} \|\mathbf{u}_i\|_1 \tag{31}$$

$$\mathbf{u}_{min_k} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \; \forall \, k \in \{0, \dots, N-1\} \tag{32}$$

$$A_{LOS_k}\mathbf{x}_k \leq b_{LOS_k} \; \forall \, k \in \{1, \dots, N\} \tag{33}$$

$$A_{Term_N}\mathbf{x}_N \leq b_{Term_N} \tag{34}$$

$$\mathbf{x}_{FT_k} \notin \mathcal{T}_k \; \forall \, k \in \{T+1, \dots, N+S\}, \forall \, \mathcal{T} \in \mathcal{F} \tag{35}$$

where Equation 32 constrains directly the input at each time step between the vector bounds $\mathbf{u}_{min_k}$ and $\mathbf{u}_{max}$; Equation 33 describes the requirements of a line-of-sight (LOS) (with respect to the target satellite); A_{LOS_k} and b_{LOS_k} describe the states within the LOS cone at step k; Equation 34 describes, through the state terms A_{Term_N} and b_{Term_N}, the terminal constraint at the end of the planning horizon, that, the spacecraft must achieve for a safe docking; and finally Equation 35 defines the safety horizon, i. e. the period of time after a failure during which both spacecrafts are guaranteed not to collide. In the latter equation $\mathbf{x}_{FT_k}$ is a chaser state at some step $k < N$, in the planning horizon, after an occurred failure at a step T and is defied by Breger and How (2008) as:

$$\mathbf{x}_{FT_k} = A_d^{k-N}\mathbf{x}_{FT_N} \text{ for } k \geq N \tag{36}$$

In Equation 35 $\mathcal{T}_k$ defines the set of position states occupied by the target, S is a number of steps the safety horizon lasts, after the end of the nominal trajectory and $\mathcal{F}$ is the set of every potential failure time at which the system must guarantee collision avoidance even during the GNC system shutdown (Breger and How, 2008).

Starting from this algorithm it is possible to expand it even further to guarantee a longer safety horizon and at the same time prevent failure trajectories from drifting away from the target. To achieve this, invariant formulation of the algorithm must be made. This is done by constraining the state of the chaser in the failure trajectory at some step k to be the same one full orbit after k. Mathematically this is done by adding another constraint to the algorithm described by Equations 31-35 (Breger and How, 2008):

$$\mathbf{x}_{FT_k} = A_d^{N_0}\mathbf{x}_{FT_N} \text{ for } k \geq T \tag{37}$$

where N_0 is the number of steps in an orbit. The state of the chaser is propagated forward using a linear state transition matrix.

With this formulation all failure orbits are guaranteed to be invariant with respect to the target, which means that if the invariance constraints are properly imposed, all failure trajectories will result in circular trajectories relative to the target at no fuel expenditure.

When compared with a strict V-bar straight line approach, the fuel savings, that Breger and How (2008) were able to obtain in their case study with the invariant formulation of the algorithm, were significant, around 9 times. However, solving this type of algorithm requires quite intensive calculations which makes its real-time implementation very difficult. A solution to this problem could be to use a linear programing (LP) formulation that allows the reduction of the computational load 150 times, while reducing the fuel optimality by only two times. This was done by Breger and How (2008) by using the convex safety formulation, as opposed to the non-convex safety formulation mentioned until now. The latter requires the chaser to remain outside a collision avoidance region while the former constrained the failure trajectories to a region known not to contain the target (Breger and How, 2008). Mathematically this is achieved by adding to the algorithm described until this point (see Equations 31-37) the convex safety constraints (Breger and How, 2008):

$$H_y \mathbf{x}_{FT_k} \geq y_{min} \; \forall \, k \in \{T+1, \ldots, N+S\} \tag{38}$$

where y_{min} is the maximum in-track position of the spacecraft and H_y is a row vector that extracts the scalar in-track component.

4.3. Control module

The task of the control function is to generate appropriate commands (i. e. control forces and torques) to achieve the nominal attitude and trajectory, according to the discrepancies of the actual state vector from the desired one. Additionally, it has to ensure the stability of the vehicle (Fehse, 2003). In homing and closing phases this is done by controlling separately the attitude and the trajectory by using open loop maneuvers based on the initial and final relative states (Luo et al., 2014; Fehse, 2003). As the distance between the two objects reduces the accuracy requirements become more stringent and the closed loop control must be employed (Fehse, 2003). This is particularly true for the final approach phase, where the maximum relative distance is only 50 m in our mission scenario.

A single-input-single-output (SISO) control system can be used to control separately the translation and the rotation of the chaser until very few meters from the target, given that the coupling between the two is relatively small. In close proximity however, the mentioned motions are coupled and a multiple-input-multiple-output (MIMO) control should be considered. This requirement is nevertheless less stringent in a case such as ours where the chaser has to acquire a berthing box and not a particular docking port (Fehse, 2003).

These considerations indicate that for the last few meters, the control module requires an advanced multi-variable controller. Other desirable features of the controller are the stability, robustness and fuel efficiency (Nolet and Miller, 2007). For this purpose a great deal of research has been preformed, as mentioned in Subsection 2.3. Based on those researches we have selected in particular two controllers: the proportional-integral-derivative (PID) and the LQR. The PID was chosen to represent a baseline for the control module, given its proven usage in space, low computational requirements and general robustness (Nolet and Miller, 2007). The LQR, on the other hand, was selected as a more advanced controller capable of dealing with the optimization process (Nolet and Miller, 2007).

The PID is a well known, commonly used controller which has a proven space heritage. It does not solve the optimization problem, but, in comparison to the others, is easier to implement and could potentially deliver a higher degree of accuracy (Nolet and Miller, 2007). The "textbook" version of the continous-time PID controller can be represented by (Haugen, 2010):

$$u(t) = u_0 + K_p e(t) + \frac{K_p}{T_i}\int_0^t e d\tau + K_p T_d \dot{e}(t) \tag{39}$$

where u_0 is the control bias or manual control value to be tuned accordingly, u is the control command output, e is the controller error defined as:

$$e(t) = r(t) - y(t) \tag{40}$$

where r is the reference and y is the measured process.

However, given the discrete nature of the GNC architecture, this standard form is not suitable to be implemented in it. For this we need a descrete-time exspression of Equation 39. Following the discretization implemented in (Haugen, 2010), the expression of the discrete-time PID controller is what follows (Haugen, 2010):

$$\begin{aligned} u(t_k) = u(t_{k-1}) + [u_0(t_k) - u_0(t_{k-1})] + \\ + K_p[e(t_k) - e(t_{k-1})] + \\ + \frac{K_p T_s}{T_i} e(t_k) + \\ + \frac{K_p T_s}{T_s}[e(t_k) - 2e(t_{k-1}) + e(t_{k-2})] \end{aligned} \tag{41}$$

where $u(t_k)$ is a control command at the step t_k and T_s is a time step or sampling interval (i.e. typically 0.1 s in commercial controllers).

LQRs, on the other hand, have never been used in space despite they are a class of well established algorithms in the control community. However, they have been used in more than one occasion for theoretical studies of close range proximity operations given their design flexibility and inherent ability to optimize cost functions. They have the same computational requirements as the PID, but are more difficult to implement (Nolet and Miller, 2007). Hereafter, a generic form of a typical LQR algorithm is presented omitting some theoretical considerations. More detail on the presented algorithm can be found in (Stengel, 1986; Nolet and Miller, 2007).

Let us first define a discrete time-variant controllable system as what follows (Nolet and Miller, 2007) :

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k \tag{42}$$

where $\mathbf{x}_k$ is a state vector at time t_k, $\mathbf{A}_k$ and $\mathbf{B}_k$ are dynamic and control matrices, respectively, and $\mathbf{u}_k$ is a control input vector. With that in mind, the optimization problem consist in finding the control gain matrix $\mathbf{K}_k$ that minimizes the quadratic cost function, J, associated with the state and control inputs, over a finite horizon of steps, N (Nolet and Miller, 2007):

$$J = \mathbf{x}_N^T\boldsymbol{Q}_f\mathbf{x}_N + \sum_{i=0}^{S-1}(\mathbf{x}_i^T\boldsymbol{Q}\mathbf{x}_i + \mathbf{u}_i^T\boldsymbol{R}\mathbf{u}_i) \tag{43}$$

where

$$\boldsymbol{Q} = \boldsymbol{Q}^T \geq 0 \tag{44}$$

$$\boldsymbol{Q}_f = \boldsymbol{Q}_f^T \geq 0 \tag{45}$$

$$\boldsymbol{R} = \boldsymbol{R}^T > 0 \tag{46}$$

The optimal control input has the following form (Nolet and Miller, 2007):

$$\mathbf{u}_k = \mathbf{K}_k\mathbf{x}_k \tag{47}$$

where the optimal gain matrix, which solves the problem is calculated using:

$$\mathbf{K}_k = -(\boldsymbol{R} + \mathbf{B}_k^T\mathbf{P}_{k+1}\mathbf{B}_k)^{-1}\mathbf{B}_k^T\mathbf{P}_{k+1}\mathbf{A}_k \tag{48}$$

and $\mathbf{P}_k$ satisfies the algebraic Riccati equation

$$\mathbf{P}_{k-1} = \boldsymbol{Q} + \mathbf{A}_k^T\mathbf{P}_k\mathbf{A}_k - \mathbf{A}_k^T\mathbf{P}_k\mathbf{B}_k(\boldsymbol{R} + \mathbf{B}_k^T\mathbf{P}_k\mathbf{B}_k)^{-1}\mathbf{B}_k^T\mathbf{P}_k\mathbf{A}_k \tag{49}$$

The recursion process for $\mathbf{P}$ is initiated with the following equation (Nolet and Miller, 2007):

$$\mathbf{P}_N = \boldsymbol{Q}_f \tag{50}$$

and is solved backward in time (i.e. from $k = N, \dots, 1$).

It is worth noting that, the weight matrices $\boldsymbol{Q}$, $\boldsymbol{Q}_f$ and $\boldsymbol{R}$ are to be determined and tuned appropriately by the user to meet the required behavior of the controller (Nolet and Miller, 2007).

Their nonlinear counterparts, the SDRE controllers, appear even more attractive, given their ability to account for perturbations and nonlinear relative dynamics of an ADR mission. The only disadvantage is that they generally require significant computational power. One approach to solve this was developed by Di Mauro (2013), but given the novelty of the approach we have to preform further in depth research and quantitative analysis to rule in or out this intriguing nonlinear technique.

4.4. Robotics module

The tasks of the robotics module are essentially to:

1. control the capture of a tumbling target, by means of a robotic manipulator
2. stabilize the compound (i. e. chaser plus target), while limiting the transfer of the angular momentum from the target to the chaser

These tasks are readily solved on ground, however, in space, the control problem arises from the fact that any motion of the manipulator exerts reaction effects on the mounting spacecraft. This leads to a series of constraints that a control architecture of a free-floating[32] robotic spacecraft must take into consideration during the operation of its manipulator. The most prominent are (Ellery, 2004):

1. generalized Jacobian is required to derive the orientation of the spacecraft
2. robot kinematics are affected by dynamic properties of both the spacecraft and the manipulator
3. dynamic singularities, function of both robot kinematics and dynamic properties of the manipulator and the spacecraft, occur in the workspace of the robot

[32] Free-floating operational mode of a robotic spacecraft implies that the motion of the spacecraft in reaction to the movement of its manipulator is allowed, while in the case of a free-flying mode the spacecraft's pose will be actively maintained in the operational space by its attitude control system.

4. joint angle configuration is path dependent due to the non-holonomic redundancy

Moreover, due to the very small magnitude of the existing dissipative forces in orbit, the control architecture of the robotic spacecraft must limit the impact forces and torques transmitted to the target body during the contact phase. At the same time, the control architecture must optimize the configuration of the robotic spacecraft during the pre-impact phase to counteract the angular momentum of the target spacecraft once the latter is safely grasped (i.e. during the post-impact phase).

Up until now, there has been a vast amount of literature covering various phases of the robotic capture of an uncooperative target, but, just as in case of a GNC, it is difficult to choose one method which could readily solve the entire problem. Moreover, most of the studies concentrate on individual operations (Yoshida et al., 2006) without considering the whole control problem of the capture process. Furthermore, most of the proposed methods are developed having in mind only the limited resources of an on-board computer. Thus, they frequently do not guarantee a feasibility of a planned trajectory and do not exploit the nonlinear nature of the robot kinematics to optimize the grasping of a tumbling target (Lampariello and Hirzinger, 2013).

The proposed control architecture, illustrated in Figure 10, is divided into two modules: an onboard (i. e. on-line) and on ground (i. e. off-line). The latter consists of target motion simulation and prediction module along with a motion planner based on learning algorithms. The former instead resides within the robotics module, outside the GNC architecture (see Figure 9), in order to enhance the computational efficiency of the onboard computer (Ellery, 2004). It uses the calculated off-line solution as an initial guess for the trajectory generation and control of the robotic arm in real time. The reason behind this division lies in the computational requirements of the motion planner, that can not be performed in a reasonable time with the computational power of nowadays onboard computers. Moreover, it is worth noting that this computationally intensive task has to be performed just once given the dynamic properties of the robotic chaser spacecraft and the target's geometry. Thus, it makes more sense to do it on ground and upload it to the spacecraft before the capture maneuver.

The described control architecture uses coordinated manipulator/spacecraft motion control, known in terrestrial robotics as full body control, to optimize the whole configuration of the robotic spacecraft, during the grasping task,

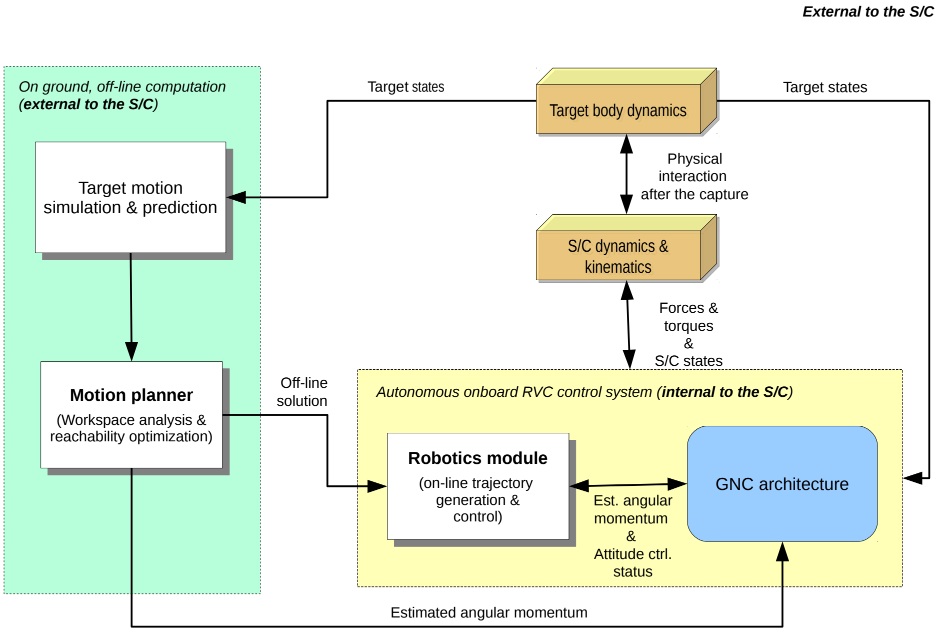
External to the S/C
On ground, off-line computation
(external to the S/C)
Target motion simulation & prediction
Motion planner
(Workspace analysis & reachability optimization)
Target states
Target body dynamics
Target states
Physical interaction after the capture
S/C dynamics & kinematics
Forces & torques & S/C states
Off-line solution
Autonomous onboard RVC control system (internal to the S/C)
Robotics module
(on-line trajectory generation & control)
Est. angular momentum & Attitude ctrl. status
GNC architecture
Estimated angular momentum

Figure 10: Concept of robotic control architecture

and to limit the transmitted angular momentum from the target body to the base spacecraft, after the grasping task.

The off-line motion planner will be based on machine learning[33] and will be dedicated to the one time, off-line identification of: a) a best suitable grasping point, b) workspace analysis and c) reachability optimization. Two possible methods were identified for this calculation: a) the black box approach and b) the parametrization. In the first case, the learning algorithm selects and evolves on its own the most suitable outputs for the optimization process, starting from current states of the chaser and target. However, it should be noted that this approach is only feasible for more or less simple optimization problems or in case the search space is well defined. Outside these boundaries the algorithm could simply fail to perform the optimiza-

[33] The individuation of a particular learning algorithm is still work in progress thus at the time of being no specific learning algorithm is mentioned in this paper.

tion process. The second approach instead relies on an operator to select the outputs to be optimized and parametrize them so that they can be then evolved by a learning algorithm. This approach assures the desired performance of the optimized configuration, although, the first method appears the most interesting one since it could give birth to new and unexpected configurations. Nevertheless, the expertise and knowledge of an operator cannot be replaced by the first method. Thus, the best solution would be to use the black box approach as a starting point of a further optimization process based on parametrization of the selected outputs.

The expected advantages of the proposed control architecture over the existing methods would be:

1. manipulator and base spacecraft motions would be constrained
2. dynamic singularities would be avoided
3. dynamic coupling would be used to facilitate the capture maneuver
4. angular momentum of the whole system would be significantly limited

4.5. Validation

The development of a GNC architecture is however just one piece of the puzzle given that the developed architecture will need to be appropriately tested. However, difficulties in computer rendering of the space scene, as observed by chaser's navigation sensors (e.g. LIDAR, IR and optical cameras), as well as modeling of the dynamics of the coupled system, might prove and arduous task. Thus, an initial software testing must be followed by a hardware in the loop (HIL) testing in order to assess the adequacy of the developed architecture. A system capable of performing such a task is the HIL simulation system for orbital rendezvous maneuvers at DFKI-RIC, that was developed in the INVERITAS project (Paul et al., 2014). The facility is located in a 24 m long, 12 m wide and 10 m high hall. It uses a cable robot system able to move a chaser platform, of up to 150 kg, in three dimensions, with one rotational axis. One industrial robotic arm is used for the movements of a target vehicle. Both systems move the chaser and the target according to a real-time software simulation of orbital dynamics, so that the relative movement of both objects inside the facility matches the movements that would occur in orbit. The system can simulate an approach of up to 16.5 m inside the available operational space. The lighting system simulates the in-orbit illumination conditions eliminating the need of computer rendering of the space scene.

5. Conclusion and future work

Up until recently the space debris issue was seen as something straight out of the science fiction. Today, thanks to two recent unfortunate collision events (one of which was intentional) and in depth studies, the space debris issue has gained more visibility. Nevertheless, the problem remains and if we do not act quickly the access to space, as we know it today, could be just a thing of the past. Thus, an active removal of intact hardware has to be preformed routinely in the next few hundred years if we are going to stabilize the space debris environment. In order to do this, space technologies need to make a significant leap forward. Most of those technologies are related to the GNC system and to the ability of a chaser spacecraft to autonomously detect, approach and capture a target.

Within this context, this paper presents a preliminary design of a GNC architecture envisioned specifically to tackle the ADR problem by means of a robotic system. Current state of the-art architectures are either envisioned for automatic systems, with humans in the loop, or they lack the ability to deal with the robotic capture phase. The GNC architecture presented here should fill that gap by including state of the art algorithms and a robotics module. Moreover, its modular structure based on the open source ROCK framework should enable the scientific community to quickly and easily modify the architecture to its own needs, once completed. Given the preliminary status of the concept, further quantitative evaluation of the selected algorithms will be performed in order to define the final structure of the concept in the near future. Moreover, the possibility of incorporating the desirable MVM and FDIR capabilities into the architecture will also be evaluated. The development of the robotics module is already one of our research goals and will be illustrated in more depth in a future paper.

Acknowledgments

The work of this research was supported by the Research Executive Agency (REA), acting under powers delegated by the European Commission, within the Stardust project of the Seventh Framework Program of the European Union, grant agreement № [317185]. Thus, the authors would like to thank the REA and the European Commission for their support and funding. Moreover, we would like to thank our reviewers for their invaluable advises and suggestions.

References

Aghili, F., Kuryllo, M., Okouneva, G., English, C., 2011. Fault-Tolerant Position/Attitude Estimation of Free-Floating Space Objects Using a Laser Range Sensor. Sensors Journal, IEEE 11, 176–185. URL: http://goo.gl/gnCEdA, doi:10.1109/JSEN.2010.2056365.

Airbus Defence & Space, 2012. Astrium wins DEOS contract to demonstrate in-orbit servicing. Online. URL: http://goo.gl/tZL3AE.

Arantes, G., Martins-Filho, L.S., 2014. Guidance and Control of Position and Attitude for Rendezvous and Dock/Berthing with a Noncooperative/Target Spacecraft. Mathematical Problems in Engineering 2014, 1–8. URL: http://goo.gl/GQ2Hjt, doi:10.1155/2014/508516. article ID 508516.

Barbee, B.W., Carpenter, J.R., Heatwole, S., Markley, F.L., Moreau, M., Naasz, B.J., VanEepoel, J., 2010. Guidance and Navigation for Rendezvous and Proximity Operations with a Non-Cooperative Spacecraft at Geosynchronous Orbit, in: George H. Born Symposium, NASA Goddard Space Flight Center, Boulder, CO, USA. pp. 1–21. URL: http://goo.gl/WfgJT5. document ID:20100019266.

Bonnal, C., Ruault, J.M., Desjean, M.C., 2013. Active debris removal: Recent progress and current trends. Acta Astronautica 85, 51–60. URL: http://goo.gl/ytEMDI, doi:10.1016/j.actaastro.2012.11.009.

Breger, L.S., How, J.P., 2008. Safe Trajectories for Autonomous Rendezvous of Spacecraft. Journal of Guidance, Control, and Dynamics 31, 1478–1489. URL: http://goo.gl/BgDNeg, doi:10.2514/1.29590.

Castronuovo, M.M., 2011. Active space debris removal-A preliminary mission analysis and design. Acta Astronautica 69, 848–859. URL: http://goo.gl/4BMTpo, doi:http://dx.doi.org/10.1016/j.actaastro.2011.04.017.

Çimen, T., 2010. Systematic and effective design of nonlinear feedback controllers via the state-dependent Riccati equation (SDRE) method. Annual Reviews in Control 34, 32–51. URL: http://goo.gl/G6NiQo, doi:10.1016/j.arcontrol.2010.03.001.

Committee on the Peaceful Uses of Outer Space, 2014. Compendium of space debris mitigation standards adopted by States and international organizations. Technical Report A/AC.105/2014/CRP.13. United Nations Office for Outer Space Affairs. Vienna, Austria. URL: http://goo.gl/LulOIv.

Crassidis, J.L., Markley, F.L., Cheng, Y., 2007. Survey of Nonlinear Attitude Estimation Methods. Journal of Guidance, Control, and Dynamics 30, 12–28. URL: http://goo.gl/ucjMVY, doi:10.2514/1.22452.

Davis, T., Baker, M., Belchak, T., Larsen, W., 2003. XSS-10 micro-satellite flight demonstration program, in: 17th Annual AIAA/USU Conference on Small Satellites, AIAA/USU, Logan, Utha, USA. pp. 1–16. URL: http://goo.gl/HYuDkJ.

Defense Industry Daily, 2007. Orbital Express: Testing On-Orbit Servicing. Online. URL: http://goo.gl/1eAWhS.

DeLuca, L., Bernelli, F., Maggi, F., Tadini, P., Pardini, C., Anselmo, L., Grassi, M., Pavarin, D., Francesconi, A., Branz, F., Chiesa, S., Viola, N., Bonnal, C., Trushlyakov, V., Belokonov, I., 2013. Active space debris removal by a hybrid propulsion module. Acta Astronautica 91, 20–33. URL: http://goo.gl/5yk58L, doi:10.1016/j.actaastro.2013.04.025.

Di Mauro, G., 2013. Theory and Experiments on Nonlinear Control for Space Proximity Maneuvers. Phd thesis. Politecnico di Milano, Milan, Italy. Milano, Italy. URL: http://goo.gl/DFEUiM.

Ellery, A., 2004. An engineering approach to the dynamic control of space robotic on-orbit servicers. Proc. Inst. Mech. Eng. Part G: Journal of Aerospace Engineering 218, 79–98. URL: http://goo.gl/788sgF, doi:10.1243/0954410041321998.

EoPortal, 2007. XSS (Experimental Spacecraft System), XSS-10 & XSS-11 Missions. Online. URL: http://goo.gl/tzYaF4.

Fehse, W., 2003. Automated Rendezvous and Docking of Spacecraft. Number 16 in Cambridge Aerospace Series. 2009 ed., Cambridge University Press, New York, USA. URL: http://goo.gl/Qm4RZs, doi:10.1017/CBO9780511543388.

Flores-Abad, A., Ma, O., Pham, K., Ulrich, S., 2014. A review of space robotics technologies for on-orbit servicing. Progress in Aerospace Sciences 68, 1–26. URL: http://goo.gl/XLmiZJ, doi:10.1016/j.paerosci.2014.03.002.

Hablani, H.B., Tapper, M.L., Dana-Bashian, D.J., 2002. Guidance and Relative Navigation for Autonomous Rendezvous in a Circular Orbit. Journal of Guidance, Control, and Dynamics 25, 553–562. URL: http://goo.gl/enRmt7, doi:10.2514/2.4916.

Haugen, F., 2010. Discretization of simulator, filter, and PID controller. Online. URL: http://goo.gl/1oKp0c.

Hawkes, E.W., Christensen, D.L., Eason, E.V., Estrada, M.A., Heverly, M., Hilgemann, E., Jiang, H., Pope, M.T., Parness, A., Cutkosky, M.R., 2013. Dynamic surface grasping with directional adhesion, in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE, Tokio, Japan. pp. 5487–5493. URL: http://goo.gl/Zfqw6T, doi:10.1109/IROS.2013.6697151.

Hillenbrand, U., Lampariello, R., 2005. Motion and Parameter Estimation of a Free-Floating Space Object from Range Data for Motion Prediction, in: Battrick, B. (Ed.), Proceedings of the 'i-SAIRAS 2005' - 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, ESA Publications Division-ESTEC, Munich, Germany. pp. 1–10. URL: http://goo.gl/YPAbvR.

Hirzinger, G., Landzettel, K., Kaiser, C., 2009. New Technologies and Robotics. 2009 ed.. John Wiley & Sons, Ltd, Chichester, UK. chapter 7. pp. 642–643. URL: http://goo.gl/AfGKiz, doi:10.1002/9780470742433.ch7.

Karr, C.L., Freeman, L., 1997. Genetic-algorithm-based fuzzy control of spacecraft autonomous rendezvous. Engineering Applications of Artificial Intelligence 10, 293–300. URL: http://goo.gl/8UId3E, doi:10.1016/S0952-1976(97)00008-0.

Kennedy, F., 2008. Orbital Express. Online.

Kervendal, E., Chabot, T., Kanani, K., 2013. GNC Challenges and Navigation Solutions for Active Debris Removal Mission, in: Chu, Q., Mulder, B., Choukroun, D., van Kampen, E.J., de Visser, C., Looye, G. (Eds.), Advances in Aerospace Guidance, Navigation and Control. Springer Berlin Heidelberg, pp. 761–779. URL: http://goo.gl/ptIzMB, doi:10.1007/978-3-642-38253-6_43.

Kessler, D.J., Cour-Palais, B.G., 1978. Collision frequency of artificial satellites: The creation of a debris belt. Journal of Geophysical Research 83, 2637–2646. URL: http://goo.gl/POFVrp, doi:10.1029/JA083iA06p02637.

Lampariello, R., Hirzinger, G., 2013. Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time, in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE, Tokyo, Japan. pp. 5652–5659. URL: http://goo.gl/4jCqDD, doi:10.1109/IROS.2013.6697175.

LaViola, J., 2003. A comparison of unscented and extended Kalman filtering for estimating quaternion motion, in: American Control Conference, 2003. Proceedings of the 2003, IEEE. pp. 2435–2440. URL: http://goo.gl/TaMijW, doi:10.1109/ACC.2003.1243440.

Lee, D., Pernicka, H., 2010. Optimal Control for Proximity Operations and Docking. International Journal of Aeronautical and Space Sciences 11, 206–220. URL: http://goo.gl/UycI7s, doi:10.5139/IJASS.2010.11.3.206.

Liou, J.C., 2011a. An active debris removal parametric study for LEO environment remediation. Advances in Space Research 47, 1865–1876. URL: http://goo.gl/cF3HlX, doi:10.1016/j.asr.2011.02.003.

Liou, J.C., 2011b. Project Review: An Update on LEO Environment Remediation with Active Debris Removal. NASA Orbital Debris Quarterly News 15, 4–6.

Liou, J.C., Johnson, N.L., 2008. Instability of the present LEO satellite populations. Advances in Space Research 41, 1046–1053. URL: http://goo.gl/nHTuCH, doi:10.1016/j.asr.2007.04.081.

Liou, J.C., Johnson, N.L., 2009. A sensitivity study of the effectiveness of active debris removal in LEO. Acta Astronautica 64, 236–243. doi:10.1016/j.actaastro.2008.07.009.

Liou, J.C., Johnson, N.L., Hill, N.M., 2010. Controlling the growth of future LEO debris populations with active debris removal. Acta Astronautica 66, 648–653. URL: http://goo.gl/e7Gjk0, doi:10.1016/j.actaastro.2009.08.005.

Luo, Y., Zhang, J., Tang, G., 2014. Survey of orbital dynamics and control of space rendezvous. Chinese Journal of Aeronautics 27, 1–11. URL: http://goo.gl/1lRWKq, doi:10.1016/j.cja.2013.07.042.

Luo, Y.z., Tang, G.j., 2005. Spacecraft optimal rendezvous controller design using simulated annealing. Aerospace Science and Technology 9, 732–737. URL: http://goo.gl/0uINTk, doi:10.1016/j.ast.2005.07.010.

Ma, Z., Ma, O., Shashikanth, B.N., 2012. Optimal approach to and alignment with a rotating rigid body for capture. The Journal of the Astronautical Sciences 55, 407–419. doi:10.1007/BF03256532.

Matsumoto, S., Jacobsen, S., Dubowsky, S., Ohkami, Y., 2003. Approach Planning and Guidance for Uncontrolled Rotating Satellite Capture Considering Collision Avoidance, in: Proceeding of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS 2003, MIT, NARA, Japan. pp. 1–8. URL: http://goo.gl/09jJy1.

Matsumoto, S., Ohkami, Y., Wakabayashi, Y., Oda, M., Ueno, H., 2002. Satellite capturing strategy using agile Orbital Servicing Vehicle, Hyper-OSV, in: Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on, IEEE, Washington, DC, USA. pp. 2309–2314. URL: http://goo.gl/h1E941, doi:10.1109/ROBOT.2002.1013576.

Naasz, B., Zimpfer, D., Barrington, R., Mulder, T., 2010. Flight Dynamics and GN&C for Spacecraft Servicing Missions, in: AIAA Guidance, Navigation, and Control Conference, AIAA, Toronto, Ontario, Canada. pp. 1–16. URL: http://goo.gl/eplEsA, doi:10.2514/6.2010-8445.

Nolet, S., Miller, D.W., 2007. Development of a Guidance , Navigation and Control Architecture and Validation Process Enabling Autonomous Docking to a Tumbling Satellite. Phd thesis. Massachusetts Institute of Technology. Cambridge, MA, USA. URL: http://goo.gl/kRwGlk.

Orbital Debris Program Office, N., 2012. Orbital Debris Frequently Asked Questions. Online. URL: http://goo.gl/SO3xXS.

Paul, J., Kirchner, F., Ahrns, I., Sommer, J., 2014. Robotics Rendezvous & Capture Test Facility "Inveritas", in: Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2014), European Space Agency, Montreal, Canada. URL: http://goo.gl/4m3gxX.

Pavone, M., Starek, J., 2014. Spacecraft Autonomy Challenges for Next Generation Space Missions. Online, to appear in Lecture Notes in Control and Information Systems (2013). URL: http://goo.gl/nU8xG0.

Personne, G., Lopez-Y-Diaz, A., Delpy, P., 2006. ATV GNC Synthesis: Overall Design, Operations and Main Performances, in: Danesy, D. (Ed.), 6th International ESA Conference on Guidance, Navigation and Control Systems, held 17-20 October 2005 in Loutraki, Greece, European Space Agency. European Space Agency, ESA, ESTEC, Noordwijk, The Netherlands. pp. 1–39. URL: http://goo.gl/hLyEXQ. bibliographic Code: 2006ESASP.606E..39P.

Praly, N., Hillion, M., Bonnal, C., Laurent-Varin, J., Petit, N., 2012. Study on the eddy current damping of the spin dynamics of space debris from the Ariane launcher upper stages. Acta Astronautica 76, 145–153. URL: http://goo.gl/nYMtw0, doi:10.1016/j.actaastro.2012.03.004.

Rupp, T., Boge, T., Kiehling, R., Sellmaier, F., 2009. Flight Dynamics Challenges of the German On-Orbit Servicing Mission DEOS, in: 21st International Symposium on Space Flight Dynamics, German Aerospace Agency, Toulouse, France. URL: http://goo.gl/RXy5o9.

Sellmaier, F., Boge, T., Spurmann, J., Gully, S., Rupp, T., Huber, F., 2010. On-Orbit Servicing Missions: Challenges and Solutions for Spacecraft Operations, in: SpaceOps 2010 Conference, AIAA, Huntsville, Alabama, USA. pp. 1–11. URL: http://goo.gl/K2b1q2.

Sommer, J., Ahrns, I., 2013. GNC for a Rendezvous in Space with an Uncooperative Target, in: Proceedings of the 5th International Conference on Spacecraft Formation Flying Missions and Technologies, German

Aerospace Agency, Munich, Germany. pp. 1–15. URL: http://goo.gl/xDzv6O.

Space Debris Office, E., 2013. About Space Debris. Online. URL: http://goo.gl/FSTkoA.

Stengel, R.F., 1986. Optimal Control and Estimation. Dover edition ed., Dover Publications, Inc., 31 East 2nd Street, Mineola, New York, 11501, NY, USA. URL: http://goo.gl/12LUm4.

Tellez, J.P.D., Krahn, J., Menon, C., 2011. Characterization of electroadhesives for robotic applications, in: Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on, IEEE, Karon Beach, Phuket, Thailand. pp. 1867–1872. URL: http://goo.gl/j3LbN9, doi:10.1109/ROBIO.2011.6181562.

Truszkowski, W., Hallock, H., Rouff, C., Karlin, J., Rash, J., Hinchey, M., Sterritt, R., 2010. Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems. 1. 2010 ed., Springer-Verlag London, London, UK. URL: http://goo.gl/hcgCKl, doi:10.1007/b105417.

Tzschichholz, T., Ma, L., Schilling, K., 2011. Model-based spacecraft pose estimation and motion prediction using a photonic mixer device camera. Acta Astronautica 68, 1156–1167. URL: http://goo.gl/Lwrisn, doi:10.1016/j.actaastro.2010.10.003.

Wan, E.A., van der Merwe, R., 2002. The Unscented Kalman Filter. haykin, s. ed.. John Wiley & Sons, Inc., New York, NY, USA. chapter 7. Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control, pp. 221–280. URL: http://goo.gl/vdWr5R, doi:10.1002/0471221546.ch7.

Woffinden, D.C., Geller, D.K., 2007. Navigating the Road to Autonomous Orbital Rendezvous. Journal of Spacecraft and Rockets 44, 898–909. URL: http://goo.gl/EZGXuk, doi:10.2514/1.30734.

Wormnes, K., Le Letty, R., Summerer, L., Krag, H., Schonenborg, R., Dubois-Matra, O., Luraschi, E., Delaval, J., Cropp, A., 2013. ESA technologies for space debris remediation, in: 6th European Conference on Space Debris, ESA, Darmstadt, Germany. pp. 1–2. URL: http://goo.gl/x32d9K.

Wright, M., 2011. Orbital Express: Overview and Lessons Learned. Technical Report November. Defence Advanced Reseacrh Projects Agency-DARPA. Mountain View, California, USA. URL: http://goo.gl/Ke3pcr.

Yoshida, K., 2009. Achievements in space robotics. Robotics & Automation Magazine, IEEE 16, 20–28. URL: http://goo.gl/RJLvnT, doi:10.1109/MRA.2009.934818.

Yoshida, K., Dimitrov, D., Nakanishi, H., 2006. On the Capture of Tumbling Satellite by a Space Robot, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE, Beijing, China. pp. 4127–4132. URL: http://goo.gl/1HA7Vk, doi:10.1109/IROS.2006.281900.

Yoshida, K., Wilcox, B., 2008. Space robots and systems, in: Siciliano, B., Khatib, O. (Eds.), Springer Handbook of Robotics. Springer Berlin Heidelberg, Berlin, Heidelberg, Germany. chapter 45, pp. 1031–1063. URL: http://goo.gl/KvgXDe, doi:10.1007/978-3-540-30301-5_46.

Youmans, E.A., Lutze, F.H., 1998. Neural Network Control of Space Vehicle Intercept and Rendezvous Maneuvers. Journal of Guidance, Control, and Dynamics 21, 116–121. URL: http://goo.gl/aotJ0m, doi:10.2514/2.4206.