IAA-AAS-DyCoSS2-14-12-08

ACTIVE VISION-BASED POSE ESTIMATION OF AN UNCOOPERATIVE TARGET

Roberto Opromolla,[*] Giancarmine Fasano,[†] Giancarlo Rufino, [‡] and Michele Grassi[§]

This paper aims at investigating the performance of a LIDAR-based system for pose determination of a known debris. A customized template matching technique is implemented for pose initial acquisition, while pose tracking is carried out by Iterative Closest Point algorithms based on different matching approaches. In order to evaluate the achievable accuracy in pose estimation, a numerical simulator is developed implementing realistic debris geometry, target/removal-system relative dynamics, and sensor operation. Results relevant to a large debris in Low Earth Orbit show that even relatively sparse point clouds allow the pose to be computed with sub-degree accuracy in attitude and sub-centimeter accuracy in the relative position.

INTRODUCTION

In the latest 50 years of space activities, the US Space Surveillance Network (SSN) has registered a continuous increase of the number of fragmentation debris in Earth orbit, mainly due to explosions of components on dead satellites or rocket bodies. Recently two catastrophic events almost doubled the amount of debris in the region below 1000 kilometers where most operating satellites are located, i.e. the low Earth orbit (LEO). These events are the anti-satellite test on the Fengyun-1C (FY-1C) weather satellite, conducted by China in 2007, and the collision between Iridium 33 and Cosmos 2251 in 2009. In particular, this last incident put into evidence that the phenomenon of debris increase in LEO will be dominated by accidental collisions between debris and operating satellites.

This possibility had been anticipated by the work of Donald Kessler in the 1970s, regarding the instability of the debris population in LEO, which showed that the environment has reached a point where collisions among orbiting objects will force the LEO population to increase even without new launches.[1] This is the reason why the overall space community (led by NASA and ESA), has acknowledged the necessity to adopt appropriate measures in order to ensure a sustainable use of space for future generations.

[*] Ph.D. student, Department of Industrial Engineering, University of Naples "Federico II", roberto.opromolla@unina.it
[†] Assistant Professor, Department of Industrial Engineering, University of Naples "Federico II", giancarmine.fasano@unina.it
[‡] Assistant Professor, Department of Industrial Engineering, University of Naples "Federico II", giancarlo.rufino@unina.it
[§] Associate Professor, Department of Industrial Engineering, University of Naples "Federico II", michele.grassi@unina.it

Numerical simulations conducted using LEGEND, a LEO-to-GEO Environment Debris model developed by NASA, have shown that the worldwide conceived mitigation measures must be accompanied by the active removal of at least five large objects per year, to stabilize the debris population preventing the triggering of the "Kessler syndrome".[2, 3]

Many studies are being conducted worldwide regarding the possibility to perform active debris removal (ADR) space missions. These studies have identified many technology challenges putting in evidence the necessity of developing technologies suitable for the rendezvous and capture of uncooperative targets as well as for target disposal.[4, 5, 6] However, up to now, the only planned demonstration missions are DEOS, by the German space agency (DLR), and ROGER, by the European space agency (ESA).[7, 8] This is a consequence of the high costs and complexity related to development, test and validation of the technologies required for ADR missions.

In this context, this paper focuses on the technologies allowing safe approach and relative navigation in close proximity of the debris target, for monitoring before capture and docking. This work is conducted within the feasibility study of a multi-removal mission.[9] Proximity operations, that must be realized autonomously and in real time, require purposely developed sensors and algorithms, capable of providing frequent, accurate and robust target pose estimates. A major issue consists in dealing with an uncooperative target which is uncontrolled, freely tumbling and not arranged to be easily approached by a removal system. In this respect, the pose determination system must rely on electro-optical (EO) sensors that can be active, e.g. Light Detection and Ranging (LIDAR) technologies, or passive, e.g. cameras operating in the visible and infrared bands.

Passive systems have lower hardware complexity. They are also less expensive than active ones and have reduced power consumption. Moreover, they can be used also for supervisory applications, like situational awareness and human-in-the-loop operations. A major disadvantage with respect to active systems is that performance is much more affected by the largely variable illumination conditions typical of the space domain, and by the presence in their field of view of other celestial bodies. On the other hand while active sensors have higher mass, cost, power consumption and poor frame rates, their high robustness to variable illumination conditions and the capability to easily discriminate the target from the background independently from the lightning conditions (easier segmentation), make LIDAR-based systems very promising for future ADR missions. Moreover, recent developments in innovative detectors (e.g. the Avalanche Photo Diode, APD), compact scanner systems (e.g. the Micro-Opto-Electro-Mechanical Systems, MOEMs), and high power and short pulse laser sources have made them more suitable for autonomous space applications.[10] Indeed, the acronym LIDAR is used to indicate a large variety of sensors that are basically able to measure distances by illuminating a target with a laser and analyzing the reflected light. Different technological solutions are suitable for LIDAR that range from scanning devices (e.g. single point, slit or pattern projection scanners) to detector arrays (e.g. flash LIDAR) or spatial light modulators. However in all these cases, the sensor output can be seen as a 3D point cloud.[11, 12, 13]

This paper focuses on performance evaluation of a LIDAR-based system for pose determination of an uncooperative target. To this end, a numerical simulation tool has been developed reproducing a realistic simulation environment in which system performance can be assessed at a preliminary level. Specifically, the relative dynamics between a large debris and a chaser platform carrying a scanning LIDAR has been reproduced according to a typical scenario for debris monitoring, as well as a realistic model of the debris target and LIDAR operation. Concerning debris simulation, the ENVISAT satellite, that has been declared inoperative by ESA in 2012, has been considered as target, since it represents a perfect example of large debris in LEO. The output

of the LIDAR has been simulated as a 3D point cloud, considering all the effects deriving from outliers and undetected regions of the target. The cloud has then been processed by the purposely developed pose estimation algorithms in order to evaluate target relative position and attitude parameters. Simulation results have been used to assess at a preliminary level the accuracy that can be achieved in pose acquisition and tracking in light of the target characteristics and the target-chaser relative dynamics. In particular, a major goal is assessing the viability of different techniques and solutions capable of operating with a sparse point cloud to reduce the computational effort while preserving pose estimation accuracy.

The paper is organized as follows. In the first section the pose determination process is described in detail. The second section briefly describes the developed simulation environment with particular attention towards target and sensor modeling. Finally the simulation scenario in terms of target-chaser relative dynamics is introduced and results from MonteCarlo (MC) simulations are presented in detail.

POSE DETERMINATION PROCESS

In general terms, pose estimation is the problem to find the set of parameters that describe the rigid rotation and the translation necessary to align a sensor reference frame (SRF) to a target reference frame (TRF). In this respect, when trying to follow the evolution of the relative pose of a moving target, two main steps are required: pose initialization and tracking. Pose initialization is carried out when the first set of data is acquired by the sensor and no a-priori information about the target relative position and attitude are available. Pose tracking is the step allowing the pose parameters to be updated, on the basis of the previously estimated ones, as a new measurement is available.

Concerning pose estimation algorithms, the best way to deal with uncooperative targets is to adopt model-based techniques, which compute the pose by comparing global or local features of the target, extracted from the available data, to the same features in a 3D model stored or built on board. Basically, they can be subdivided in template matching techniques based on correlation, suitable to pose acquisition, and iterative techniques which minimize an error function, in a least square sense, suitable to pose tracking.

In the case of template matching, the 3D model is used to generate a set of views of the target which samples the pose space. Then, the pose producing the best correlation with the measured data is selected as initial pose estimate. A fundamental limitation of this technique is the high computational cost, since pose search must be carried out over the entire pose space, that has to be densely sampled in order to get sufficient accuracy. Hence, further studies are needed to overcome this limitation, starting from the possibility of organizing the pose database with hierarchical structures.[14] Indeed, a fast convergence of the pose acquisition process is advisable when operating in close proximity of an uncooperative large target.

Template matching provides the initial pose estimate needed to start pose tracking, which is performed by computing the pose that finds the best fit between the acquired 3D point cloud and the 3D target model in a least square sense.[15, 16, 17] The best example of this approach is given by the Iterative Closest Point (ICP) algorithm.[18] However, the effectiveness of such techniques can be largely affected by the initial pose estimate as well as by the chaser-target relative dynamics. Indeed, the latter one can produce sparse as well very concentrated 3D point clouds providing only partial views of a large target and a variable pose estimation accuracy.

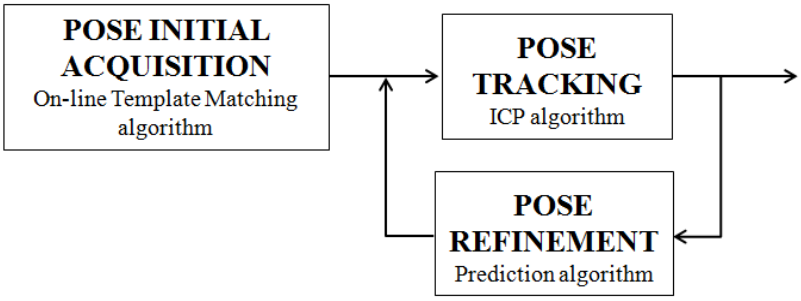

Figure 1. Flow diagram that describes the adopted pose determination process.

A flow diagram that shows the logical steps of the adopted pose determination process and indicates also the techniques adopted for each step, is presented in Figure 1. It also shows how an additional step, known as pose refinement, can be introduced in order to improve the accuracy level of the pose solution provided as input to the tracking algorithm.

Before presenting the details of the pose determination process, it is necessary to clarify how the relative pose parameters, used in the rest of the paper, are defined. The relative position is indicated as the vector $\underline{T}$, which is expressed in the SRF. The relative attitude is represented by a 321 sequence of Euler angles, i.e. yaw (γ), pitch (β) and roll (α), or equivalently by the unit quaternion q, that gives the rotation matrix, $\underline{\underline{R}}(q)$ which aligns the TRF to the SRF.

Pose acquisition

In order to solve the problem of the pose initial acquisition, an optimized template matching algorithm has been developed and implemented in this work. Recalling that the limits of template matching are the huge amount of data to be stored on board and the consequently high computational cost, our approach has been to choose as tentative solution for the relative position vector, the centroid of the acquired LIDAR point cloud. This allows building the database on-line thus getting two important advantages, in terms of data storage, with respect to the off-line approach. First, since the relative position is fixed, the database is built sampling only the attitude parameter space thus reducing significantly the number of templates to be generated and compared to the data set; second, the database is created dynamically and so the only data to store on board are the geometrical information about the target needed to generate the generic template. For each template, that is also a point cloud, the correlation parameter is computed as the mean squared distance of corresponding points between the template itself and the LIDAR point cloud. The pose solution given by the algorithm is the one that minimize the correlation function, *C*, shown in Eq. (1).

$$C(q,\underline{T}) = \frac{1}{N_t(q,\underline{T})} \sum_{i=1}^{N_t(q,\underline{T})} \left| \underline{P}^i_{SENSOR} - \underline{P}^i_{TEMPLATE}(q,\underline{T}) \right|^2 \tag{1}$$

In Eq. (1) N_t is dimension of each template, while $\underline{P}^i_{SENSOR}$ and $\underline{P}^i_{TEMPLATE}$ are respectively the ith point of the LIDAR and template point clouds.

Pose tracking

Regarding pose tracking, we have focused our attention on the ICP algorithm, which is used to find the optimal transformation aligning the LIDAR point cloud and the one, obtained by the knowledge of the target model that we will indicate as model point cloud. The ICP procedure comprises several different phases for each of which many variants have been derived.[19] Among them, the mandatory steps to be implemented are the matching step, the selection of the error metric function, and the selection of the minimization technique. In this work the error metric function, *f*, shown in Eq. (2), is the mean squared distance of the corresponding points between the LIDAR point cloud and model one. It is minimized using a closed form solution based on unit quaternions.[20]

$$f(q,\underline{T}) = \frac{1}{N_p}\sum_{i=1}^{N_p}\left|\underline{P}^{i}_{MODEL} - \underline{\underline{R}}(q)^{T}(\underline{P}^{i}_{SENSOR} + \underline{T})\right|^{2} \tag{2}$$

In Eq. (2) N_p is dimension of the LIDAR point cloud, while $\underline{P}^{i}_{MODEL}$ is the ith point of the model point cloud.

Once the optimal quaternion, which provides the best rotation to align the two reference frames, is obtained, the relative position vector is updated as the relative distance between the centroid of the two point clouds ($\underline{\mu}_{MODEL}$ and $\underline{\mu}_{SENSOR}$, respectively, for the model and LIDAR point clouds) expressed in the same coordinate system.

$$\underline{T} = \underline{\underline{R}}(q)\underline{\mu}_{MODEL} - \underline{\mu}_{SENSOR} \tag{3}$$

Particular attention has been paid to the ICP matching step that has the function of finding the correspondences between LIDAR and model points. It represents, in fact, one of the algorithm's stages that mostly determines its computational cost. In this context, two different solutions have been implemented which are known as Nearest Neighbor (NN) and Normal Shooting (NS). The NN associates each element in the LIDAR point cloud to the closest one in the model point cloud according to an Euclidean metric. The NS consists in finding the model point corresponding to the generic LIDAR measurement point by projecting it on the closest surface of the target according to the local normal unit vector.

Finally, a pose prediction algorithm is adopted for the pose refinement step. It consists in modifying the attitude parameters that are used to initialize the ICP on the basis of the ones computed at two or more previous time instants. More precisely the value of each Euler angle is updated by exploiting its time derivative, as shown in Eq. (4) for the roll angle.

$$\alpha_{INIT}^{t=i} = \alpha_{EST}^{t=i-1} + \left.\frac{d\alpha_{EST}}{dt}\right|_{t=i-1} \tag{4}$$

SIMULATION ENVIRONMENT

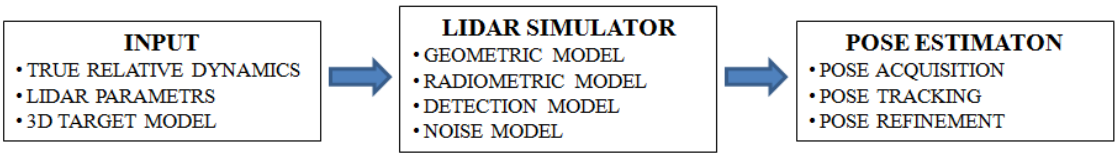

Figure 2. Simulation environment block representation.

The developed SW environment can be represented as the aggregation of three main blocks, as shown in Figure 2. In the first block all the input data necessary to perform the simulation are collected. In particular, chaser and target absolute positions are propagated numerically starting from assigned instantaneous orbit parameters, and taking into account the main harmonics of Earth potential and other perturbations such as solar radiation pressure and aerodynamic drag. Chaser orbit maneuvers can also be simulated if necessary. Chaser and target attitude can be computed by considering numerical propagation and/or controlled attitude dynamics (only for the chaser). All these pieces of information are used by the second block, i.e. the LIDAR simulator, in order to reproduce realistically the measurements of a scanning LIDAR. Finally the data set is sent to the pose estimation block in which the algorithms previously described are applied to estimate the target relative pose.

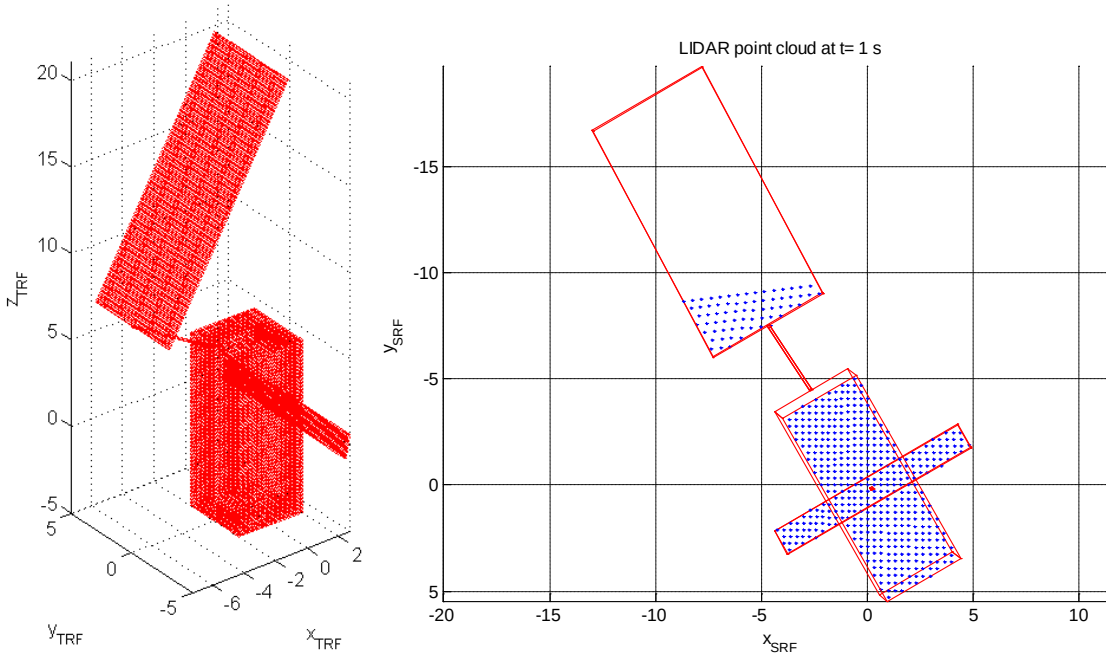

Figure 3. (Left) ENVISAT model point cloud in the TRF. (Right) LIDAR point cloud in the SRF for $\alpha = -97.5°$, $\beta = 29.6°$ and $\gamma = 83.7°$, with an angular resolution of 1°.

The point cloud in the TRF generated for the target 3D model, and an example of LIDAR point cloud in the SRF generated by the LIDAR simulator are provided in Figure 3.

Target model

The simulations are performed considering, as target, a model of the satellite ENVISAT which has been declared inoperative by ESA since April 2012. The model takes into account the main body of ENVISAT connected to its solar array and its synthetic aperture radar (SAR) antenna by two appendixes. For modeling purpose, all these five elements have a cuboid shape, and their dimensions are consistent with information provided in the literature. The TRF is defined as a body centered reference frame with the origin in the center of mass of the ENVISAT main body (which is assumed to coincide with its geometrical center). It is important to notice that dealing with very large debris and considering a rather close relative trajectory, the target will be only partially in view by the LIDAR the most time, thus affecting algorithm performance.

It is important to underline that, if the NN technique is implemented, the 3D model of the target must be discretized in order to generate a point cloud (see Figure 3). More precisely a fixed discretization step of 5 centimeters has been set for each surface of the target. On the other hand, when adopting the NS approach, the model point cloud can be generated dynamically on the basis of very limited geometrical characteristics about the target, i.e. the number of plane surfaces with which it is modeled and the corresponding local normal unit vectors.

Measurement model

In order to evaluate the performance of the developed pose estimation algorithm, a Time-of-Flight (TOF) based scanning LIDAR has been considered as the main relative navigation sensor installed on the chaser platform for close-proximity operations. This sensor is able to scan its FOV by steering the laser beam thanks to high-speed and high-precision galvanometers. It has been assumed that the sensor has the same angular resolution (δ_{LOS}) in both azimuth and elevation. As shown in Figure 2, the LIDAR simulator is composed of four different models.

The geometric model first identifies the surfaces of the target which are seen by the sensor on the basis of the real relative pose; then, it computes the range of the intersection (ρ) of each laser beam, generated by the sensor according to a raster scan pattern by implementing a ray tracing algorithm. It consists in computing ρ by solving the system of equations composed of the parametric representation of a straight line and the Cartesian representation of a plane, as shown in Eq. (5) where all the quantities are expressed in the SRF.

$$\begin{cases} x = \lambda_x * \rho \\ y = \lambda_y * \rho \\ z = \lambda_z * \rho \\ n_x * x + n_y * y + n_z * z + d = 0 \end{cases} \Rightarrow \rho = \frac{-d}{n_x * \lambda_x + n_y * \lambda_y + n_z * \lambda_z} \tag{5}$$

If more than one plane is in sight for a certain laser shot, the ambiguity is solved by selecting the minimum range of interception.

The radiometric and the detection models are used to establish if the backscattered laser beams are detected or not. More precisely the probability of detection has been computed as a function of the probability of false alarm, the transmitted power, the range of interception, the beam angle of incidence with respect to the local normal unit vector.[21] In this approach, the radiometric as-

pects are taken into account by assuming a model of diffuse reflection for each target surface. Finally, the detected point cloud is modified to take the possible noise sources into account, in order to get a more realistic simulation of the LIDAR operation. The pointing uncertainty, σ_{LOS}, is modeled as a Gaussian white noise in both azimuth and elevation angles, for each emitted laser beam; the range error due to computed TOF uncertainty, σ_{RANGE}, is modeled as a Gaussian white noise. Furthermore, a percentage of outlier, P_O, has been considered for each target component; the outliers are selected in the detected point cloud thanks to a random extraction from a uniform distribution and are characterized by a range uncertainty that is four times σ_{RANGE}.

SIMULATION SCENARIO AND RESULTS

In this section the simulation scenario is presented in terms of the approach adopted to define the target/chaser relative dynamics, together with the achieved results.

Relative Dynamics

Circular relative trajectories are particularly suited for debris monitoring because of constant range to the target. Thus, this geometry has been adopted for the simulated scenario. Chaser orbit has been designed as follows.

ENVISAT Two Line Elements (TLE) relevant to a recent measurement (January 7th 2014) have been considered as starting point for the analysis. A relative motion model developed in mean orbit parameters and based on a time explicit formulation has been used to compute differences in mean orbit parameters that allow the chaser to fly in a circular formation with respect to the target, considering a radius of 20 m.[22] In order to achieve a stable relative motion, no differences in mean semi-major axis and inclination have been considered as a design constraint. On the other hand, slight differences in eccentricity, mean anomaly, argument of perigee, and right ascension of the ascending node, have been obtained from the required geometry. The transformation from mean to osculating orbit parameters has been carried out on the basis of Schaub-Junkins approximation.[23]

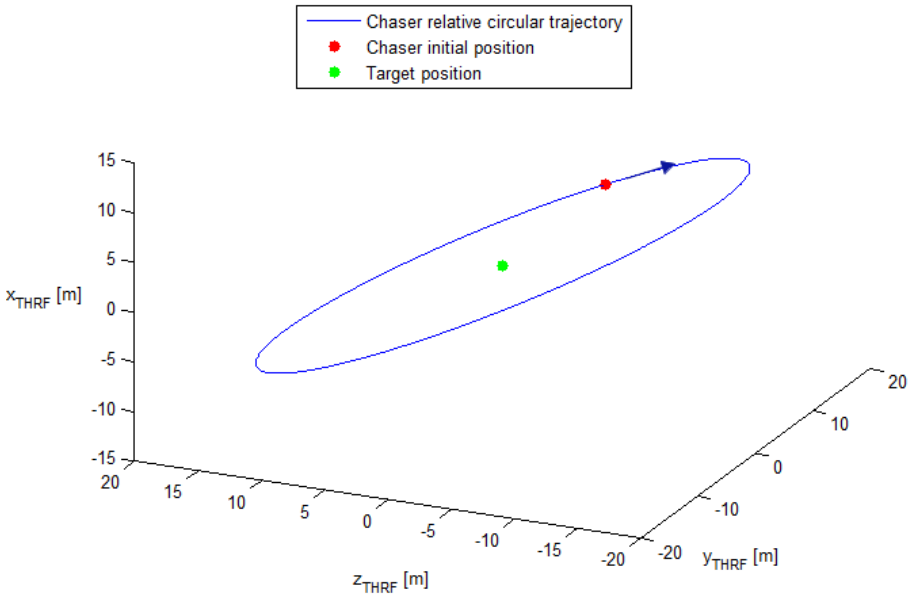

Figure 4. Chaser relative trajectory in the target Hill reference frame (THRF).

The resulting relative trajectory lies on a plane that intersects the target cross-track/along-track plane along the along-track axis and is inclined 30 deg to that plane, as shown in Figure 4.

Regarding attitude, it is assumed that ENVISAT dynamics is mostly due to gravity gradient effects. Specifically, it has been simulated considering a constant rotation rate along the yaw axis, and small amplitude zero offset oscillations at orbital frequency of pitch and roll angle. Finally, chaser attitude has been simulated assuming that it is three-axis stabilized on the circular relative trajectory and the LIDAR boresight axis is always pointed towards the target center of mass.

Results

A series of MonteCarlo (MC) simulations have been performed considering four different approaches for pose determination after the initial acquisition: these four variants are obtained considering the NN or the NS techniques for the matching step in the ICP algorithm, and presence or not of the prediction technique.

Table 1. LIDAR parameters used in the simulations.

LIDAR parameters	
FOV (°)	40
Update rate (Hz)	1
δ_{LOS} (°)	1
σ_{RANGE} (mm)	25
σ_{LOS} (°)	0.0007
P_O	• 5 %, solar array and SAR antenna • 7 %, main body and appendixes

The characteristics of the scanning LIDAR used to simulate the measured point clouds are in Table 1. This data are relevant to a worst-case scenario since the sensors used in space applications have typically better noise characteristics.[24]

Table 2. Template matching results averaged on 100 MC simulations.

	True initial pose	Estimated initial pose	Range of variation of the pose parameters			Number of templates
			Min	Max	Step	
α (°)	-97.48	-100.00	-180	180	20	3610
β (°)	29.59	50.00	-90	90	20	
γ (°)	83.67	80.00	-180	180	20	
T_X (m)	0	0.63	$\underline{\mu}_{SENSOR}$ - fixed for each template			
T_Y (m)	0	0.98				
T_Z (m)	-20.00	-18.76				

Results for pose initialization with the on-line template matching algorithm are collected in Table 2. To keep the time required to perform the template matching within few seconds, the angular step used to sample the range of variation of the Euler angles is selected to be quite large (i.e. 20°); this is possible because the only purpose of this technique is to find an initial solution for the pose parameters that falls in the field of convergence of the ICP.

Analyzing these results we can state that the algorithm is robust with respect to the sensor noise. The estimation error is, in fact, mainly due to the decision of generating each template using as a tentative solution for the relative position vector the LIDAR point cloud centroid.

In general, the performance of the different ICP techniques is evaluated in terms of estimation accuracy in translation vector and Euler angles.

For the sake of brevity, it can be also computed in terms of purposely defined equivalent parameters for the relative pose. The relative attitude estimation error is represented by φ_{ERR}, which is the equivalent Euler angle corresponding to the error quaternion (q_{ERR}) between the true and estimated ones (respectively q_{TRUE} and q_{EST}), defined as in Eq. (6).

$$q_{TRUE} = q_{ERR} * q_{EST} \tag{6}$$

The relative position estimation error is represented by $\left|\underline{T}\right|_{ERR}$, which is the difference between the Euclidean norms of the true and estimated relative position vectors.

Table 3. Statistical parameters regarding the ICP estimation error on the relative attitude and position evaluated considering the entire orbit averaged on 100 MC simulations.

Tracking Technique	φ_{ERR} [°]			$\left\|\underline{T}\right\|_{ERR}$ [m]		
	mean	std	rms	mean	std	rms
NN	$1.08\cdot10^{-1}$	$6.1\cdot10^{-2}$	$1.24\cdot10^{-1}$	$-3.99\cdot10^{-3}$	$4.16\cdot10^{-3}$	$5.76\cdot10^{-3}$
NN + prediction	$1.30\cdot10^{-1}$	$1.03\cdot10^{-1}$	$1.66\cdot10^{-1}$	$-4.24\cdot10^{-3}$	$2.10\cdot10^{-3}$	$4.73\cdot10^{-3}$
NS	$1.14\cdot10^{-1}$	$9.7\cdot10^{-2}$	$1.50\cdot10^{-1}$	$-4.31\cdot10^{-3}$	$8.09\cdot10^{-3}$	$9.17\cdot10^{-3}$
NS + prediction	$1.15\cdot10^{-1}$	$1.31\cdot10^{-1}$	$1.75\cdot10^{-1}$	$-4.79\cdot10^{-3}$	$2.45\cdot10^{-3}$	$5.38\cdot10^{-3}$

Table 3 reports statistical results relevant to these two parameters. From these results, it is evident that all the developed techniques allow obtaining an average accuracy along the orbit of 0.1° in attitude and less than 1 cm in the relative position.

It also appears that the NN approach provides slightly better performance for both the relative attitude and position parameters. Moreover the use of the pose prediction appears not to impact at a significant level algorithm performance for both the NN and NS methods.

To clarify these results, it is necessary to consider that the ICP performance can be negatively affected by the fact that the target lies partially out of LIDAR FOV. This means that the estimation error must be related to the true pose that determines the shape of the LIDAR point cloud. Thus, let us now focus our attention on the temporal behavior of the estimation error of the pose parameters, e.g. referring to the case of the classic NN technique not aided with the refinement algorithm.

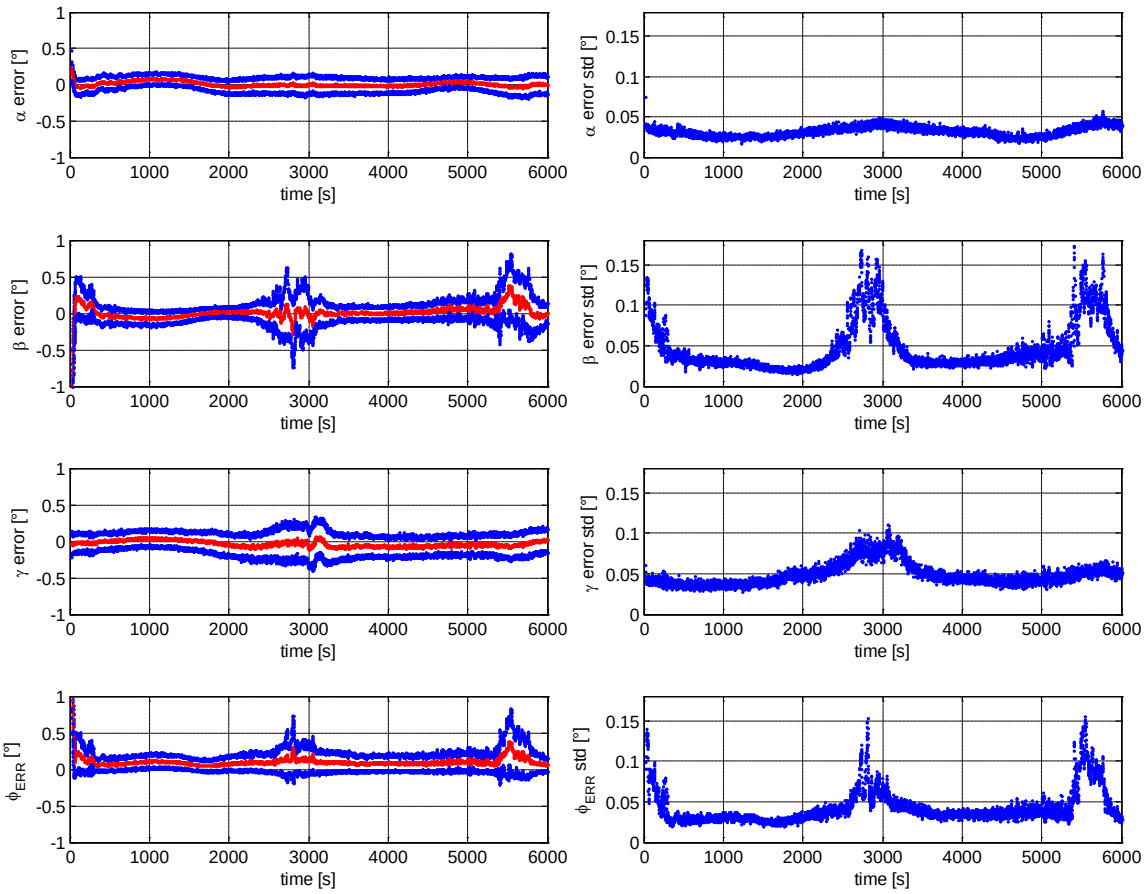

Figure 5. (Left) Mean error, in red, mean $\pm$ three times the standard deviation error, in blue, for relative attitude estimation. (Right) Standard deviation error for relative attitude estimation. These results are relevant to the NN technique not aided by the prediction algorithm.

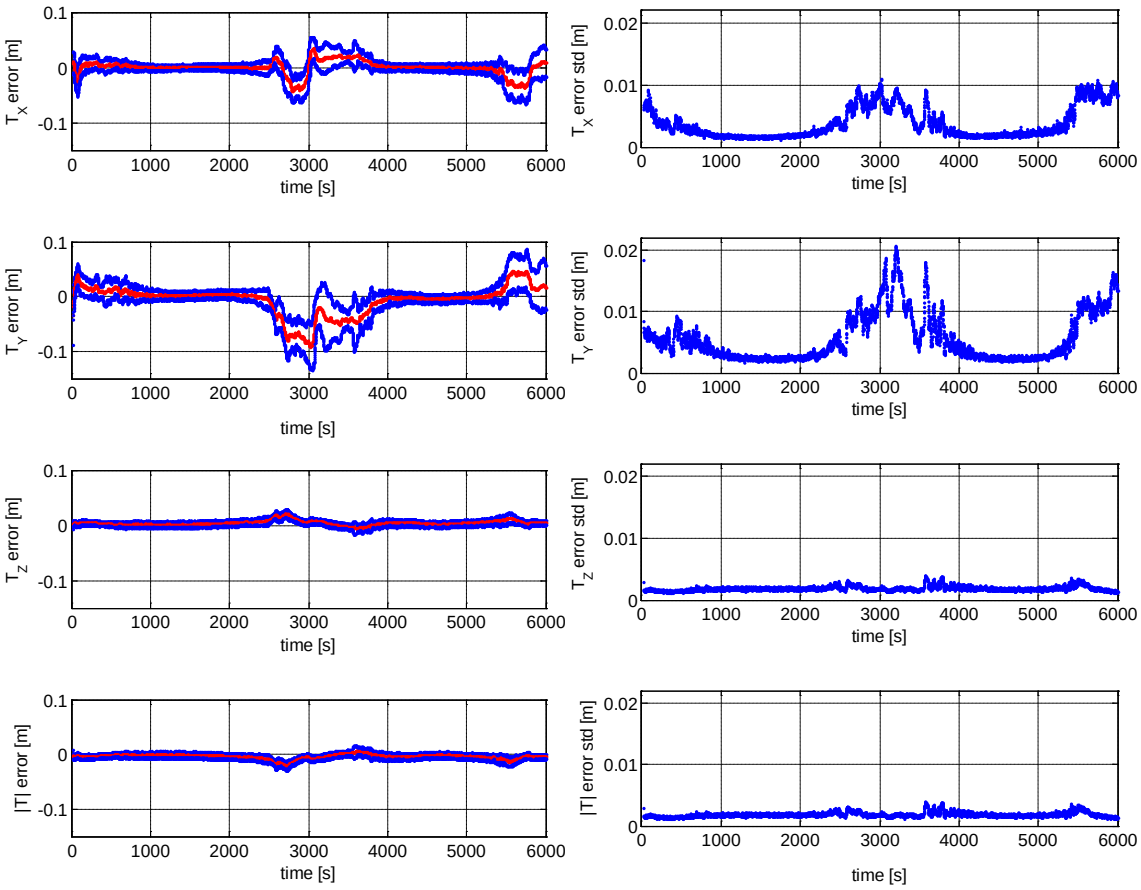

Figure 6. (Left) Mean error, in red, mean $\pm$ three times the standard deviation error, in blue, for relative position estimation. (Right) Standard deviation error for relative position estimation. These results are relevant to the NN technique not aided by the prediction algorithm.

Figures 5 and 6 depict error mean and standard deviation relevant to attitude and translation parameters. They show that the ICP is able to immediately recover the error in pose initial acquisition. However, during the relative orbit, there are three time intervals during which the algorithm performance gets worse, identified by three spikes.

The first spike, at the beginning of the relative orbit, is simply a consequence of the necessity to cope with the initial acquisition error. The other two spikes, respectively in the middle and at the end of the relative trajectory, correspond to not favorable viewing conditions of the target. More precisely, during those time intervals the target visibility in the LIDAR FOV is strictly limited to a single flat surface of the main body, also corresponding to a lower number of 3D measurements in a relatively small FOV portion. In this case since the target appears to have a more regular shape, the problem of pose determination becomes more easily affected by measurement-model matching ambiguities.

Table 4. Statistical parameter regarding the estimation error on the relative attitude and position evaluated considering the average orbit in the time interval between 1000 and 2000 seconds.

Tracking Technique	φ_{ERR} [°]			$\left\|\underline{T}\right\|_{ERR}$ [m]		
	mean	std	rms	mean	std	rms
NN	$8.34 \cdot 10^{-2}$	$2.29 \cdot 10^{-2}$	$8.65 \cdot 10^{-2}$	$-1.87 \cdot 10^{-3}$	$9.15 \cdot 10^{-4}$	$2.08 \cdot 10^{-3}$
NN + prediction	$7.53 \cdot 10^{-2}$	$4.50 \cdot 10^{-3}$	$7.55 \cdot 10^{-2}$	$-3.66 \cdot 10^{-3}$	$5.39 \cdot 10^{-4}$	$3.70 \cdot 10^{-3}$
NS	$8.38 \cdot 10^{-2}$	$3.28 \cdot 10^{-2}$	$9.00 \cdot 10^{-2}$	$-1.72 \cdot 10^{-3}$	$1.00 \cdot 10^{-3}$	$1.99 \cdot 10^{-3}$
NS + prediction	$5.36 \cdot 10^{-2}$	$4.46 \cdot 10^{-3}$	$5.38 \cdot 10^{-2}$	$-4.06 \cdot 10^{-3}$	$3.79 \cdot 10^{-4}$	$4.08 \cdot 10^{-3}$

The results in Table 4 show that, far from the spikes, the pose estimation accuracy has become better than the one shown in Table 4 by 50% on average. In particular we can notice that for the relative attitude the refinement algorithm allows slightly improving the performance of the ICP. In general, it can be foreseen that prediction will impact pose estimation accuracy on the basis of LIDAR measurement noise as well as the ratio between target rotation rate and sensor update rate.

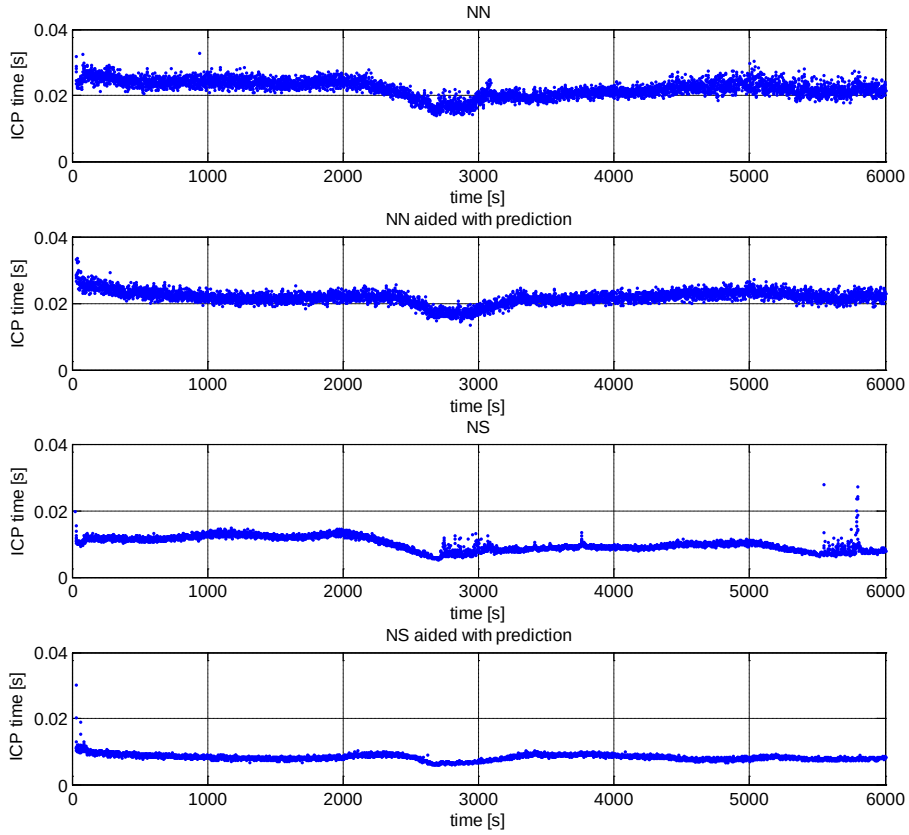

Figure 7.Temporal variation of the ICP computational time, comparison among the four analyzed ICP variants.

Figure 7 depicts the computational time of the different ICP variants, as a function of time along the orbit. It shows that the fundamental advantage of the NS technique is the significantly reduced time required by the algorithm to reach convergence. This is due to the fact that the time needed to perform the matching step in the ICP algorithm is much lower. Moreover, the adoption of the refinement technique allows further reducing the computational load by reducing the number of ICP iterations.

CONCLUSION

In this paper the performance of LIDAR-based algorithms for pose estimation of an uncooperative target has been preliminary evaluated. A simulation environment comprising target/chaser relative dynamics consistent with a debris monitoring scenario, and the operation of a scanning time-of-flight LIDAR, has been used to this aim. Specifically, the on-line template matching algorithm adopted for pose initialization has the capability of finding a pose solution which falls in the field of convergence of the ICP algorithm, and allows limiting the amount of target data to be stored on board. Concerning the tracking step, all the analyzed variants of the ICP algorithm have comparable performance in terms of pose estimation accuracy. In particular, using a sparse point cloud with 1° level resolution, the normal shooting approach aided with the prediction algorithm provides a rms error on the whole orbit of less than 0.1° in attitude and less than 1 cm in position. Finally, simulation results demonstrate that, regarding the matching step of ICP technique, normal shooting has a much lighter computational load compared with nearest neighbor.

REFERENCES

[1] D.J. Kessler and B.G. Cour-Palais, "Collision frequency of artificial satellites: Creation of a debris belt." *Journal of Geophysical Research*. Vol. 83, 1978, pp. 2637-2646.

[2] J.C. Liou, D.T. Hall, P.H. Krisko and J.N. Opiela, "LEGEND–A three-dimensional LEO-to-GEO debris evolutionary model." *Advances in Space Research*. Vol. 34, No. 5, 2004, pp. 981-986.

[3] J.L. Liou, "An active debris removal parametric study for LEO environment remediation." *Advances in Space Research*. Vol. 47, No. 11, 2011, pp. 1865-1876.

[4] S. Cook, B. Hughes and M. Graves, "FASTSAT Orbital Debris Removal Mission - An Affordable, Scalable and Responsive Flight Demonstration." *Proceeding of the 63rd International Astronautical Congress*. Naples, Italy, October 2012.

[5] C. Bonnal, J.M. Ruault, and M.C. Desjean, "Active debris removal: Recent progress and current trends." *Acta Astronautica*. Vol. 85, 2013, pp. 51-60.

[6] X. Clerc and I. Retat, "Astrium Vision on Space Debris Removal." *Proceeding of the 63rd International Astronautical Congress*. Naples, Italy, October 2012.

[7] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber, "On-Orbit Servicing Missions: Challenges and Solutions for Spacecraft Operations". *AIAA SpaceOps 2010 Conference*. Huntsville, Alabama (USA), April 2010.

[8] D.A Smith, C. Martin, M. Kassebom, H.Petersen, A. Shaw, B. Skidmore, D. Smith, H. Stokes and A. Willig, "A mission to preserve the geostationary region." *Advances in Space Research* Vol. 34, No. 5, 2004, pp. 1214-1218.

[9] L.T. De Luca, F. Bernelli, F. Maggi, P. Tadini, C. Pardini, L. Anselmo, M. Grassi, D. Pavarin, A. Francesconi, F. Branz, S. Chiesa, N. Viola, C. Bonnal, V. Trushlyakov, I. Belokonov, "Active space debris removal by a hybrid propulsion module." *Acta Astronautica*. Vol. 91, 2013, pp. 20-33.

[10] J. Pereira do Carmo, B. Moebius, M. Pfennigbauer, R. Bond, I. Bakalski, M. Foster, S. Bellis , M. Humphries, R. Fisackerly and B. Houdou, "Imaging LIDARs for Space Applications." *Proceedings of SPIE Novel Optical Systems Design and Optimization XI*. Vol. 7061, San Diego, California (USA), August 2008.

[11] F. Blais, “Review of 20 Years of Range Sensor Development.” *Journal of Electronic Imaging*. Vol. 13, 2004, pp. 231-240.

[12] F. Crosby and S.H. Kang, "Object Identification in 3D Flash Lidar Images." *Journal of Pattern Recognition Research*. Vol. 6, No. 2, 2011, pp. 193-200.

[13] J.A. Christian and S. Cryan, “A Survey of LIDAR Technology and its Use in Spacecraft relative Navigation.” *AIAA Guidance, Navigation, and Control (GNC) Conference*. Vol. 7061, San Diego, California (USA), August 2008.

[14] C. Reinbacher, M. Ruther and H, Bischof, "Pose Estimation of Known Objects by Efficient Silhouette Matching." *IEEE 20th International Conference on Pattern Recognition*. pp. 1080-1083, Istanbul, Turkey, August 2010.

[15] D.G. Lowe, "Object Recognition from Local Scale-Invariant Features." *Proceedings of the 7th IEEE international conference on Computer Vision*. Vol. 2, pp. 1150-1157, Kerkyra, Greece, September 1999.

[16] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 24, No. 7, pp. 932-946.

[17] A.I. Comport, E. Marchand, M. Pressigout and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework." *IEEE Transactions on Visualization and Computer Graphics*. Vol. 12, No. 4, pp. 615-628.

[18] P.J. Besl and N.D. McKay, "Method for registration of 3-D shapes." *Proceeding of SPIE Sensor Fusion IV: Control Paradigms and Data Structures*. Vol. 1611, pp. 586-606, Boston, Massachusetts (USA), April 1992.

[19] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm." *Proceeding of the Third IEEE International Conference on 3-D Digital Imaging and Modeling*. pp. 145-152, Quebec City, Canada, June 2001.

[20] B.K. Horn, ”Closed-form solution of absolute orientation using unit quaternions.” *Journal of the Optical Society of America*. Vol. 4, No. 4, 1987, pp. 629-642.

[21] G. Euliss, A. Christiansen, and R. Athale, "Analysis of laser-ranging technology for sense and avoid operation of unmanned aircraft systems: the tradeoff between resolution and power." *Proceedings of SPIE Defense and Security Symposium*. Vol. 6962, 2008, pp. 696208-1-10.

[22] G. Fasano and M. D’Errico, "Modeling Orbital Relative Motion to Enable Formation Design from Application Requirements." *Celestial Mechanics and Dynamical Astronomy*. Vol. 105, No. 1-3, 2009, pp. 113-139.

[23] H. Schaub and J.L. Junkins, *Analytical Mechanics of Space Systems*, AIAA Education Series, 2003.

[24] B. Ruel, S., Luu, T., Anctil, M., & Gagnon, S. (2008, March), "Target localization from 3d data for on-orbit autonomous rendezvous & docking." *IEEE Aerospace Conference*. pp. 1-11, Big Sky, Montana (USA), March 2008.