Lidar-based pose estimation for non-cooperative rendezvous

Matthieu Dardelle and Cédric Renault***
**Astrium Space Transportation*
51-61, Route de Verneuil, BP 3002, 78133 Les Mureaux Cedex France
***Astrium Space Transportation*
Rue du Général Niox, B.P. 20011, 33165 St Médard en Jalles Cedex France

Abstract

The interest in orbital non-cooperative rendezvous is expanding as it could allow satellite servicing or space debris deorbitation. In this paper, a navigation solution is presented for non-cooperative rendezvous using scanning LIDAR measurements at close range. The complete 6-DOF relative state is estimated with ICP algorithm in closed-loop with a Kalman filter. The ICP algorithm matches two sets of model and measured points to calculate relative attitude and position. This information is the input of a Kalman filter, using free-rotating object equations for attitude estimation and translation equations of motion for a spacecraft relative to an elliptical Keplerian reference orbit for position estimation. An attitude estimation algorithm is used to initialize ICP algorithm, making the presented navigation solution complete and independent.

1. Introduction

A rendezvous simulator was developed to implement this navigation solution and simulation results are presented for a debris representative of H10 Ariane 4 upper stage. This target is cylinder shaped and presents symmetry around roll axis, making the attitude estimation particularly challenging. Simulations are run using a sensor model and give results in terms of relative position and attitude errors for rendezvous with a space debris. Translation and angular velocities are also estimated by the navigation solution. Different relative trajectories are tested for distances between 100m and 25m (station-keeping, linear approach, fly-around).
The interest of adding extended Kalman filter to ICP algorithm is investigated showing promising results. The specific 6DOF initialization issue is addressed in order to propose a full navigation architecture solution based on sensors suite composed of star-tracker, gyrometers, accelerometers and LIDAR.

1.1 Overview

Significant progresses have been made in orbital rendezvous these last years. For example, Automated Transfer Vehicle (ATV) is capable to perform an autonomous rendezvous with the International Space Station since 2008. Nevertheless, information about target's position and attitude is provided to the chaser vehicle and different helps like reflectors can be used to facilitate the rendezvous.
The problematic of non-cooperative rendezvous is to perform the same operation with space objects unable to provide any information about their state to the chaser. Furthermore, the targets could be space debris for which no docking port or specific materials to facilitate rendezvous exist. In that frame, the chaser vehicle must be able to make a complete relative navigation to approach the target by using only measurements from its embedded sensors. In this paper, the close rendezvous is studied using a scanning LIDAR sensor, based on laser Time–of-flight principle to give coordinates of points hitting the target. It has the advantage of providing information directly exploitable by the navigation and it is not sensible to illumination conditions, unlike camera sensors. LIDAR points are processed by a navigation solution to estimate both relative position and attitude. The particular interest of this solution is to use jointly attitude and position estimation algorithm with a Kalman filter to benefit from knowledge of the target's dynamics evolution.

1.2 Related Work

Different solutions exist and have already been studied in the literature. We can distinguish algorithms adapted to monovision and those adapted to cloud of 3D points which can be obtained with stereovision or LIDAR sensors.

In [1] and [2] both position and attitude are estimated at close range by using a monocular sensor. It consists in matching edges with a geometric model of the target. The initialization problem was not studied. The same principle is used in [3] but a Kalman filter is implemented to use dynamics equations. All these solutions have the same drawback of using data that must be processed to extract useful information like edges and cannot estimate easily all the attitude configurations as the sensor information is limited.

In [4], the Iterative Closest Point algorithm is used with clouds of 3D points obtained with stereovision or LIDAR sensor. The ICP algorithm is in closed-loop with a Kalman filter. This solution shows good results with an experimental setup. However, the initialization problem is not addressed in this paper.

[5] shows techniques to initialize the attitude for a close rendezvous navigation algorithm. Polygonal Aspect Hashing and Geometric Hashing are adapted to cloud of 3D points and could be used in a first-step before ICP algorithm. [6] uses voxel representation from 3D measurements to estimate inertia matrix of the target and deduce an attitude. This algorithm does not need any initialization and could be used to give a first estimate to another navigation algorithm.

[7] presents an algorithm adapted to stereovision sensors. Feature points are used to reconstruct target’s shape and estimate relative state between the chaser and the target. An Extended Kalman Filter is used based on the dynamics equations.

1.3 Notation and coordinate frames

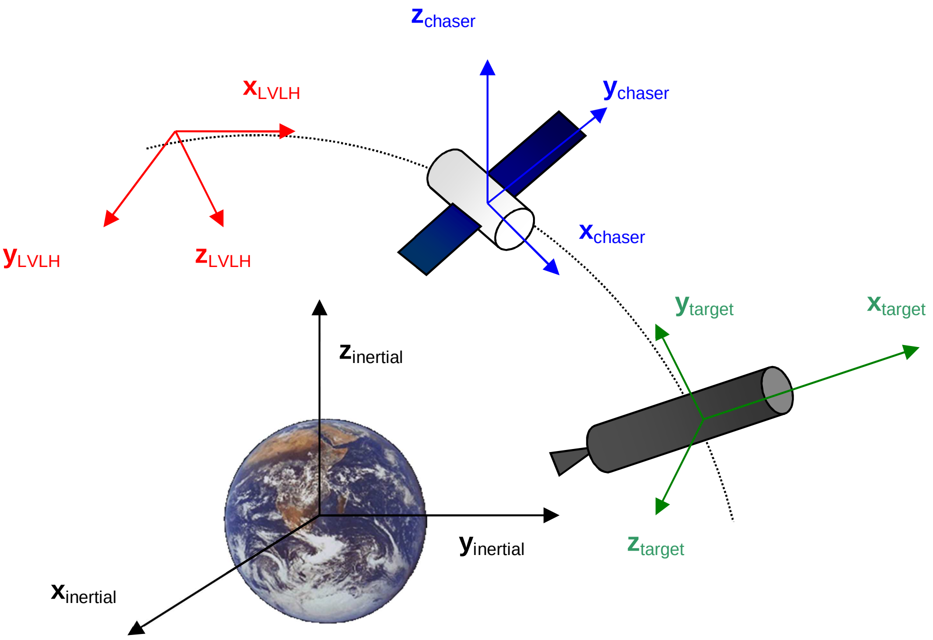

Figure 1: Coordinate frames used for navigation

Measurements provided by the LIDAR are 3-D points expressed in chaser frame. Thus, the measurements are representative of target’s position and attitude expressed in chaser frame.

The absolute navigation of the chaser gives information of position and attitude of the chaser frame with regard to the inertial frame. Furthermore, equations of the dynamics of the target are known with respect to this inertial frame.

Finally, LVLH frame (Local-Vertical Local-Horizontal) is used to express translation equations of motion for a spacecraft relative to an elliptical Keplerian reference orbit. The LVLH frame is expressed as follow : z-axis points toward the Earth (local vertical), x-axis is in the velocity frame and orthogonal to z-axis and finally y-axis completes the orthogonal frame.

In the equations listed above, rotations are represented by quaternions. A quaternion is composed of a scalar and vectorial part.

$$q = \begin{bmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos(\Phi/2) \\ \vec{e} \cdot \sin(\Phi/2) \end{bmatrix} \quad \textbf{(1)}$$

The rotation is expressed as a rotation-axis $\vec{e}$ and an angle of rotation Φ around this axis.

2. Navigation Description

In this section, the algorithms used in the navigation solution are described. This solution is formed of three main blocks which are the pose estimation algorithm, the navigation filter and the attitude initialization algorithm.

2.1 Overview

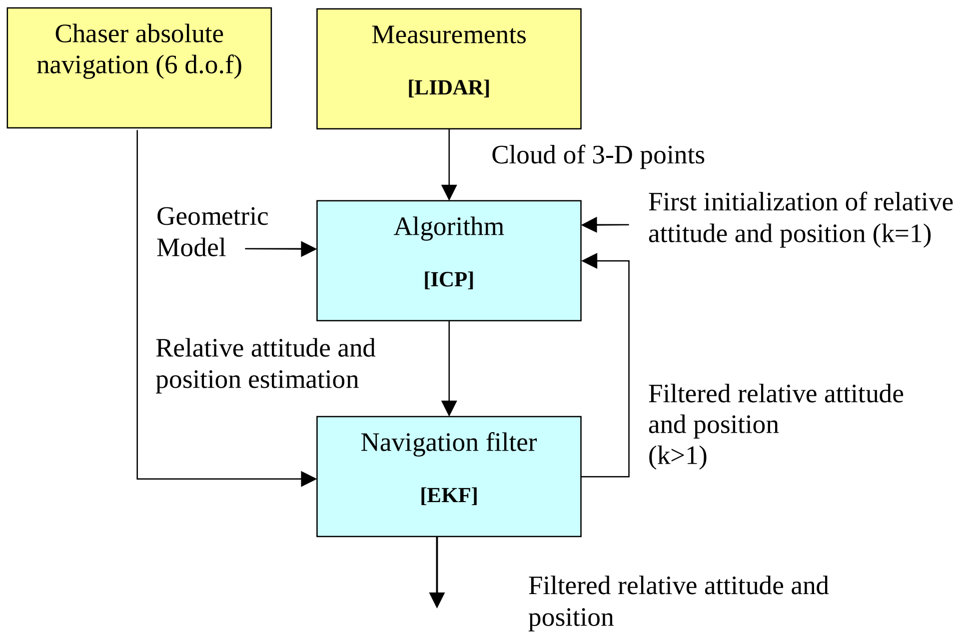

Figure 2: Navigation solution structure

This figure shows the general structure of the navigation solution. The ICP algorithm needs to be initialized at the first iteration, and then the algorithm runs in closed loop with a navigation filter. The estimation at an iteration k is used as initialization for the ICP algorithm at iteration k+1.

2.2 Pose estimation algorithm: Iterative Closest Point (ICP)

Iterative Closest Point is a model-based iterative algorithm for attitude estimation of a target. Two sets of points are required: a set of data points and a set of points from the geometric model of the target.

Therefore, this algorithm is adapted to clouds of 3-D points which are directly obtained with LIDAR sensor. This algorithm provides the quaternion and the translation vector of the target with regard to the chaser frame. The quaternion is a rotation representation directly linked to the attitude matrix A (or rotation matrix). The rotation matrix and translation vector r applied to measured points matches them to the model points. A_0 and r_0 are attitude matrix and translation vector used for initialization.

The first step is to find the correspondence between data points and points from the model. For each measured point, the following optimization problem is resolved:

$$v_i = \arg\min_{v_j \in M} \left\| A_o (u_i + r_0) - v_j \right\|, \forall i = 1,...,m \quad \textbf{(2)}$$

Then, the second step is to find the attitude matrix and translation vector which minimize the error between the rotated and translated measured points and the model points.

$$\{r, A\} = \arg\min_{r,A} \left\| A(u_i + r) - v_i \right\|^2, \forall v_i \in V, \forall u_i \in U \quad \textbf{(3)}$$

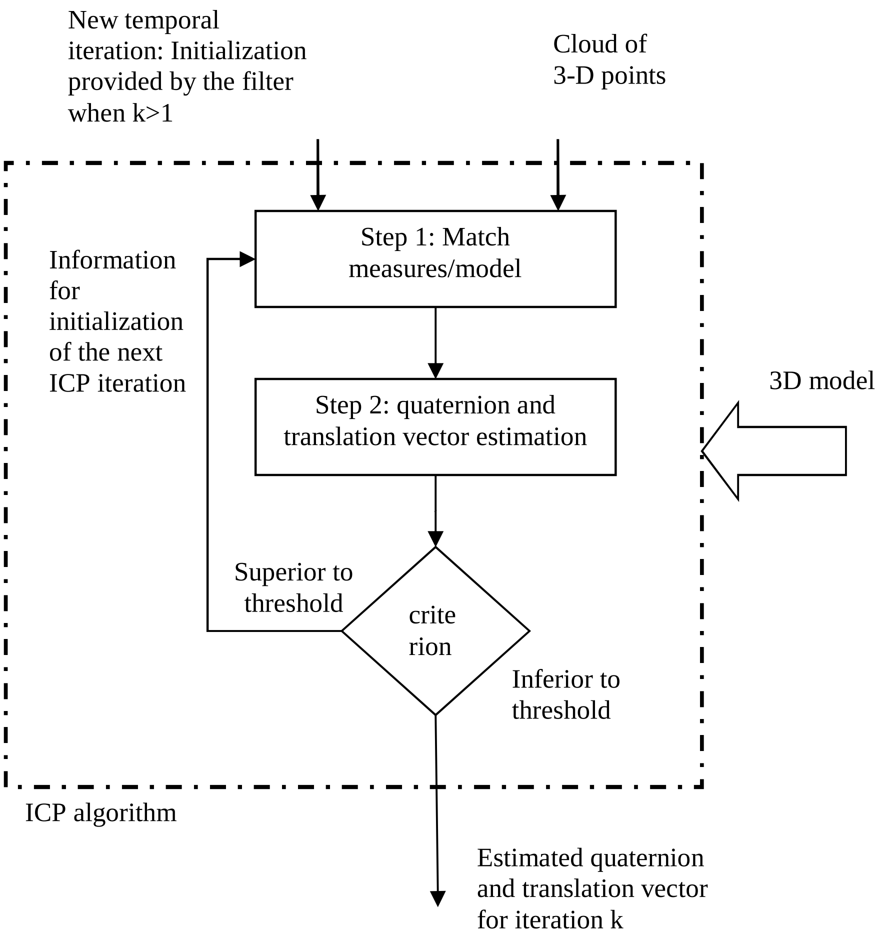

Figure 3: ICP algorithm

In [4], a method to calculate directly the optimal solution for the quaternion and the translation vector without solving the optimization problem is presented. The estimated position and attitude are given with regard to the sensor frame. In this study, we consider that the sensor frame is identical to the chaser. Thus, we have directly position and attitude expressed in chaser frame.

This algorithm needs a good initialization to provide correct results. Otherwise, the ICP could converge toward a local minimum. The solution of the previous iteration can be used to initialize the algorithm at the next measurement.

However, two problems remain: the algorithm is sensitive to noise and obstruction of the target and secondly, the first initialization must be calculated beforehand.

In order to solve the first problem, the results of the ICP can be filtered by a Kalman Filter and provided as the initialization of the next temporal iteration. A solution to solve the problem of initialization is presented in the next part.

2.3 ICP initialization

ICP algorithm must be first initialized in attitude and position to converge toward the good solution. Thus, we must design a specific algorithm able to give a rough estimation of the relative state using only geometric model of the target and measurements provided by the LIDAR.

The following solution can be used to initialize the target attitude for Iterative Closest Point algorithm. This is inspired from voxel-generation method presented in [6].

First, the inertia matrix of the target is computed from measured points:

$$\begin{aligned}
J_{xx} &= \sum \left\{ (y_i - y_m)^2 + (z_i - z_m)^2 \right\} \\
J_{yy} &= \sum \left\{ (x_i - x_m)^2 + (z_i - z_m)^2 \right\} \\
J_{zz} &= \sum \left\{ (x_i - x_m)^2 + (y_i - y_m)^2 \right\} \\
J_{xy} &= J_{yx} = -\sum (x_i - x_m)(y_i - y_m) \\
J_{yz} &= J_{zy} = -\sum (y_i - y_m)(z_i - z_m) \\
J_{zx} &= J_{xz} = -\sum (z_i - z_m)(x_i - x_m)
\end{aligned} \qquad \textbf{(4)}$$

$$J = \begin{bmatrix} J_{xx} & J_{yx} & J_{zx} \\ J_{xy} & J_{yy} & J_{zy} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \qquad \textbf{(5)}$$

The sums are done on measured points $\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$ and $\begin{pmatrix} x_m \\ y_m \\ z_m \end{pmatrix}$ is the centroid of the image.

Then, the rotation matrix describing the attitude of the principal geometric axes with respect to the inertial frame is formed by the eigenvector matrix of J.

$$J = R \Lambda R^T \qquad \textbf{(6)}$$

With R the rotation matrix equal to the eigenvector matrix of J and Λ the diagonal eigenvalue matrix of J.
However, the eigenvectors can be positive or negative and only one solution allows a correct matching between rotated measurements and model points as done by the ICP algorithm. Thus, the error between rotated points and corresponding model points is computed for each of 8 possible solutions, and finally the solution corresponding to the smallest error is retained. The initial position is simply given by the mean coordinates of the measured cloud of points. This solution is only adapted to the case where the rendezvous sensor has the complete target in its field-of-view. Furthermore, only targets with specific shapes and mass repartition could be adapted to this initialization algorithm.

2.4 Navigation Extended Kalman Filters

Here, the navigation filter is used in order to estimate the complete relative state of the target: attitude, angular velocity, position and velocity. As the dynamics in position and attitude are independent, we expose the equations used separately.

- Attitude equations

The equations presented in this part are used to estimate the attitude and the angular velocity of the target, using the output of the Iterative Closest Point as measurement.
The states relative to the attitude dynamics are:

$$X = [\omega_x \quad \omega_y \quad \omega_z \quad \theta_x \quad \theta_y \quad \theta_z] \tag{7}$$

States ω_x, ω_y and ω_z are the angular velocities of the target frame with regard to the inertial frame expressed in target frame.

States θ_x, $\theta_y$and $\theta_z$are the attitude angles corresponding to the integration of angular velocities.
This representation of the attitude as three angles has the advantage of having a physical reality, which allows a simpler and more efficient tuning.

The evolution of the angular velocities is given by the Euler's equations, with null perturbation torques. This case is representative of the movement of debris.

$$\begin{aligned} I_{xx}\dot{\omega}_x &= (I_{yy} - I_{zz})\omega_y\omega_z \\ I_{yy}\dot{\omega}_y &= (I_{zz} - I_{xx})\omega_z\omega_x \\ I_{zz}\dot{\omega}_z &= (I_{xx} - I_{yy})\omega_x\omega_y \end{aligned} \tag{8}$$

The evolution of the attitude is given by the following expression:

$$\begin{aligned} \dot{\theta}_x &= \omega_x \\ \dot{\theta}_y &= \omega_y \\ \dot{\theta}_z &= \omega_z \end{aligned} \tag{9}$$

In order to design the filter, this system is linearized around an equilibrium point.

$$A(\bar{X}) = \frac{\partial a}{\partial x}(\bar{X}) = \begin{bmatrix} 0 & p_x\overline{\omega_z} & p_x\overline{\omega_y} & 0 & 0 & 0 \\ p_y\overline{\omega_z} & 0 & p_y\overline{\omega_x} & 0 & 0 & 0 \\ p_z\overline{\omega_y} & p_z\overline{\omega_x} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

$$\Delta\dot{X} = A(\overline{X})\Delta X \tag{11}$$

With:

$$\begin{aligned} p_x &= \left(I_{yy} - I_{zz}\right)/I_{xx} \\ p_x &= (I_{zz} - I_{xx})/I_{yy} \\ p_x &= \left(I_{xx} - I_{yy}\right)/I_{zz} \end{aligned} \tag{12}$$

The attitude provided by the ICP algorithm is the attitude of the target expressed in the chaser frame. We apply a change of coordinates in order to have the attitude of the target expressed in the inertial frame.

$$q_{target \to inertial} = q_{chaser \to inertial} \times q_{target \to chaser} \quad \textbf{(13)}$$

We assume that the attitude of the chaser is perfectly known.
In order to use the measurement expressed as a quaternion, the predicted quaternion is calculated:

$$q_{pred} = q_{previous} \otimes \delta q \quad \textbf{(14)}$$

With:

$$\delta q = \begin{bmatrix} \left(1 - (dt/2 \cdot \omega_x)^2 - \left(dt/2 \cdot \omega_y\right)^2 - (dt/2 \cdot z)^2\right)^{1/2} \\ dt/2 \cdot \omega_x \\ dt/2 \cdot \omega_y \\ dt/2 \cdot \omega_z \end{bmatrix} \quad \textbf{(15)}$$

$q_{previous}$ is the estimated quaternion at the previous iteration. The innovation is built as the difference between the predicted and measured quaternions, and is converted to angles representation.
The observation equation is:

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot X \quad \textbf{(16)}$$

- Position equations

The relative position between the chaser and the target satellites is also given by the Iterative Closest Point algorithm and can be filtered by an Extended Kalman filter in order to improve performance of the algorithm.
The translation equations of motion for a spacecraft relative to an elliptical Keplerian reference orbit are:

$$\begin{aligned}
&\ddot{x} - 2\omega_0 \dot{z} - \dot{\omega_0} z - {\omega_0}^2 x + \frac{\mu \cdot x}{[x^2 + y^2 + (h_0 - z)^2]^{3/2}} = u_x + v_x \\
&y + \frac{\mu \cdot y}{[x^2 + y^2 + (h_0 - z)^2]^{3/2}} = u_y + v_y \\
&\ddot{z} + 2\omega_0 \dot{x} + \dot{\omega_0} x - {\omega_0}^2 z + \frac{\mu}{{h_0}^2} - \frac{\mu \cdot (h_0 - z)}{[x^2 + y^2 + (h_0 - z)^2]^{3/2}} = u_z + v_z \\
&\ddot{h_0} - h_0 \cdot {\omega_0}^2 + \frac{\mu}{{h_0}^2} = 0 \\
&\dot{\omega_0} + \frac{2 \cdot \dot{h_0} \cdot \omega_0}{h_0} = 0
\end{aligned} \quad \textbf{(17)}$$

The term x is the in-track position component, z is the position along the negative radial direction and y is the position along the negative orbit normal direction. h_0 and ω_0 are the radius and angular velocity of the reference orbit. μ is the gravitational parameter of the planet. Finally, u and v are inputs and noise on the three translation components. These inputs are used to consider chaser boosts.

The following states are implemented in the filter to describe the relative position dynamics:

$$X_{tr} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & h_0 & \dot{h}_0 & \omega_0 \end{bmatrix}^T \quad \textbf{(18)}$$

As previously, we need to linearize these equations in order to design the Extended Kalman filter. The Jacobian matrix is thus calculated.

$$\Delta \dot{X}_{tr} = A_{tr}(\overline{X}_{tr}) \Delta X_{tr} \quad \textbf{(19)}$$

As for the attitude part, we must apply a change of coordinates to the position provided by the ICP algorithm in order to have it in LVLH frame. The chaser's absolute navigation is assumed perfect, so we know without error the attitude between chaser and LVLH frames.

$$r_{LVLH} = q_{chaser \to LVLH} \times r_{chaser} \quad \textbf{(20)}$$

Thus, the observation equation is:

$$y = \begin{bmatrix} 1_{3\times3} & 0_{3\times6} \\ 0_{6\times3} & 0_{6\times6} \end{bmatrix} \cdot X_{tr} \quad \textbf{(21)}$$

3. Simulation results

A simulator was developed to simulate close rendezvous with a space debris. This simulator is coded in Matlab and allows the simulation of the satellites trajectories, the sensor measurements and navigation solution. The results presented in this section are obtained with a target representative of H10 upper-stage and LIDAR measurements. Several trajectories are considered to obtain position and attitude precision.

3.1 Simulation conditions

The target considered is a H10 upper-stage. It is approximately 10m higher and is nearly symmetric around z-axis. Thus, it is particularly challenging to estimate the roll angle. The geometric model is formed of approximately 100 points and is shown in the next figure.

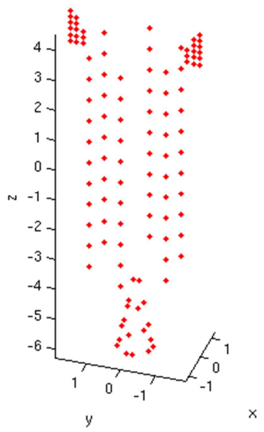

Figure 4: H10 geometric model

The H10 target is in flat-spin with an angular velocity equal to 0,3 deg/s. It means that it rotates around a transverse angle and is stabilized in the inertial frame. The considered orbit is circular and located at an altitude of 1000 km. Measurements are generated and used at a frequency of 1 Hz, while the navigation filters run at a frequency of 10 Hz. A noise on chaser's absolute navigation is added to be coherent with the use of navigation sensors such as GPS, startracker, gyrometers and accelerometers.

Several simulations are run using the complete navigation solution.

- Station-keeping at 50m
- Station-keeping at 25m
- Approach from 100m to 50m
- Fly-around from 50m to 25m

The fly-around maneuver consists in applying an acceleration boost when being at 50m, making the chaser turning around the target with a relative distance between 50m and 25m.

3.2 LIDAR measurements generation

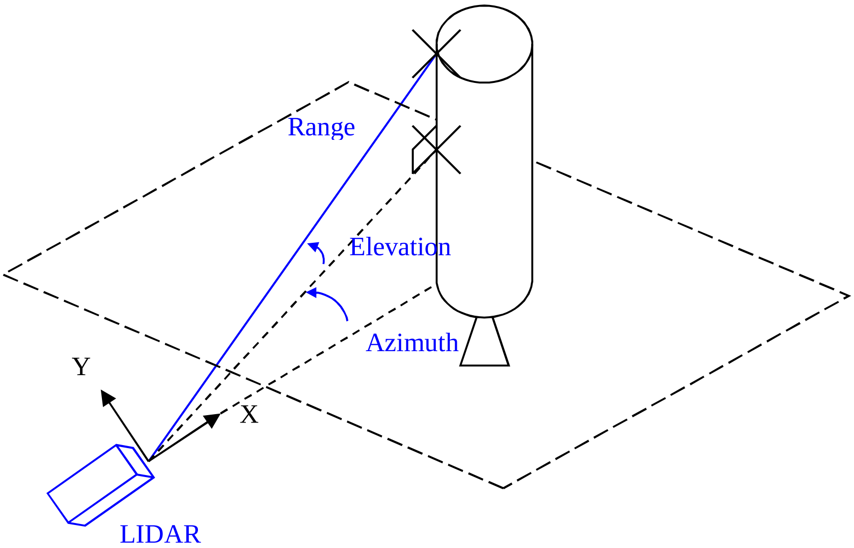

Figure 5: LIDAR principle

Once the lines of sight are known, we calculate their intersection with the planes of the target. If a line of sight has more than one intersection, we only keep the intersection with the closest plane. By doing this operation for each line of sight, we can keep only target points in front of the sensor for attitude estimation.

- Image Resolution

This parameter will define the horizontal vertical resolution of the output images. For a LIDAR simulation, resolution should be taken quite low to be realistic. It is a trade off because high resolutions require a lot of calculation time in the reconstruction algorithm. In the following simulations, resolution is equal to 0.46°. Considering a 40° field-of-view, it gives a scan resolution of roughly 90x90 (~8000 scans).

- LIDAR noise

To enhance realism of LIDAR measurements, an error is introduced. For each component, a Gaussian noise with specific variance and bias is added. The range variance and bias are linearly dependant of the distance between the chaser and the target. Here are the LIDAR parameters considered for the study:

Table 1: LIDAR model parameters

Horizontal Field of View	40° Max	Azimuth noise (1σ)	0.4°
Vertical Field of View	40° Max	Azimuth bias	0.03°
Minimum range	0.7 m	Range noise at minimum (1σ)	0.1 m
Maximum range	2000 m	Range noise at maximum (1σ)	0.38 m
Elevation noise (1σ)	0.4°	Range bias at minimum	0.05 m
Elevation bias	0.03°	Range bias at maximum	0.7 m

Azimuth and elevation bias and variance are the same whatever the distance is. A bias of 0.03° and a noise with a standard-deviation of 0.4° is added on every point. The error on range (bias and variance) is function of the distance.

We also add a spatial frequency bias, which corresponds to a noise function of the laser direction. For example, the elevation frequency bias is the highest when the azimuth is equal to zero.

3.3 Results

The attitude and position errors are given in the chaser frame and are shown on the following figures. The attitude error is given in terms of Euler angles. φ is the first rotation around y-axis, then Θ represents the rotation around z-axis and finally Ψ is the last rotation around x-axis.

Table 2: Navigation filter tuning

Navigation filter parameter	**Value (1σ)**
Attitude part	
Initial covariance matrix	0.1°/s on angular velocity, 2° on attitude
Model noise	0.01 °/s on angular velocity, 0.1° on attitude
Measurement noise	5° on attitude
Position part	
Initial covariance matrix	$0_{9\times9}$
Model noise	0.03 m on position; 0.01 m/s on speed
Measurement noise	0.5 m

The initialization solution gives the following results for unitary simulations:

Table 3: Initialization solution results

Studied case	**Initial position error /X Y Z-axis**	**Initial attitude error φ, θ, Ψ**
Station-Keeping at 50m	-0.59m / 0.33m / -0.12m	0.31° / -0.14° / -0.22°
Station-Keeping at 25m	-0.66m / 0.11m / -0.01m	-0.75° / -2.05° / -0.38°
Approach from 100m to 50m	-0.45m / 0.12m / -0.08m	0.04° / -5.36° / -3.15°
Fly-around from 50m to 25m	-0.54m / 0.41m / -0.08m	1.29° / 1.22° / -0.16°

We observe a different result between the Station-Keeping at 50m and the fly-around, which is also initialized at 50m. This difference probably comes from the measurement noise on the cloud of points, which is not the same in both cases. However, the attitude error stays below 1.5° on the three Euler angles.

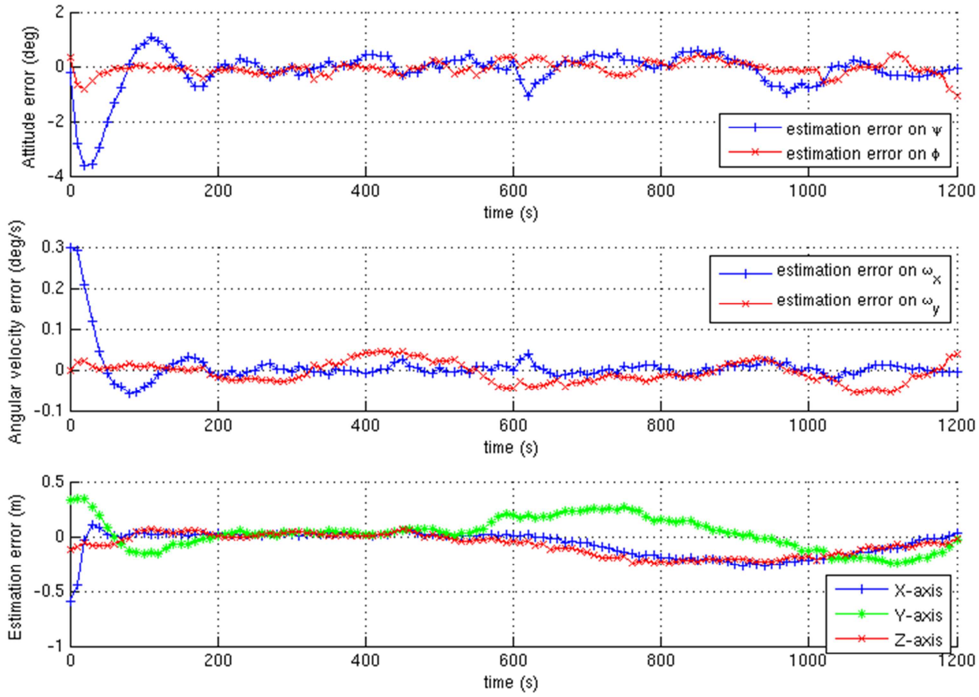

Figure 6: Estimation errors – Attitude, angular velocity and position for Station-Keeping at 50m

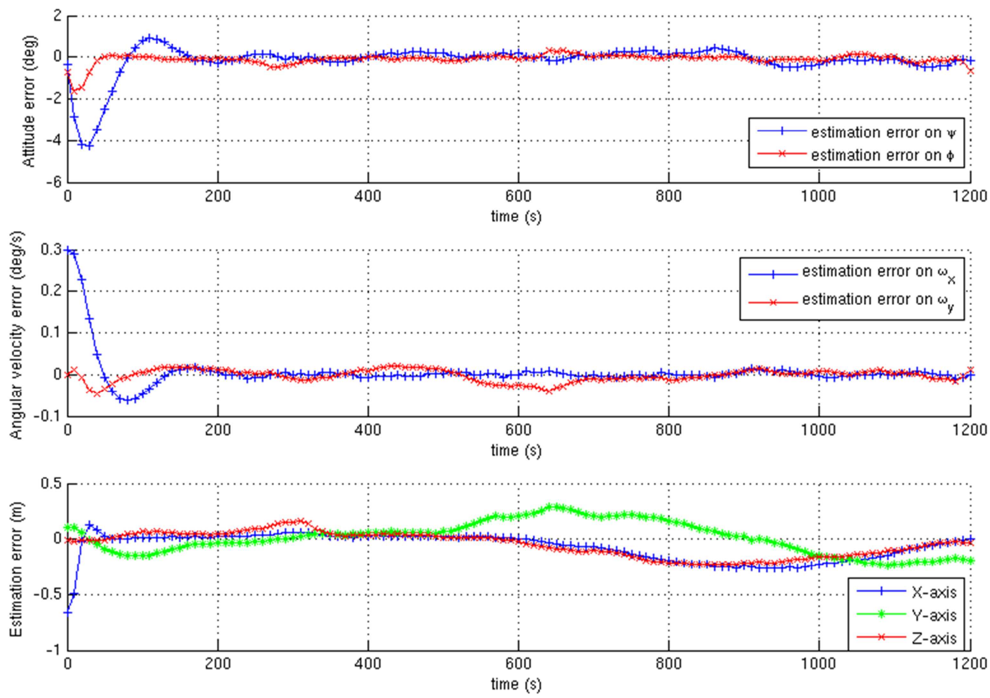

Figure 7: Estimation errors – Attitude, angular velocity and position for Station-Keeping at 25m

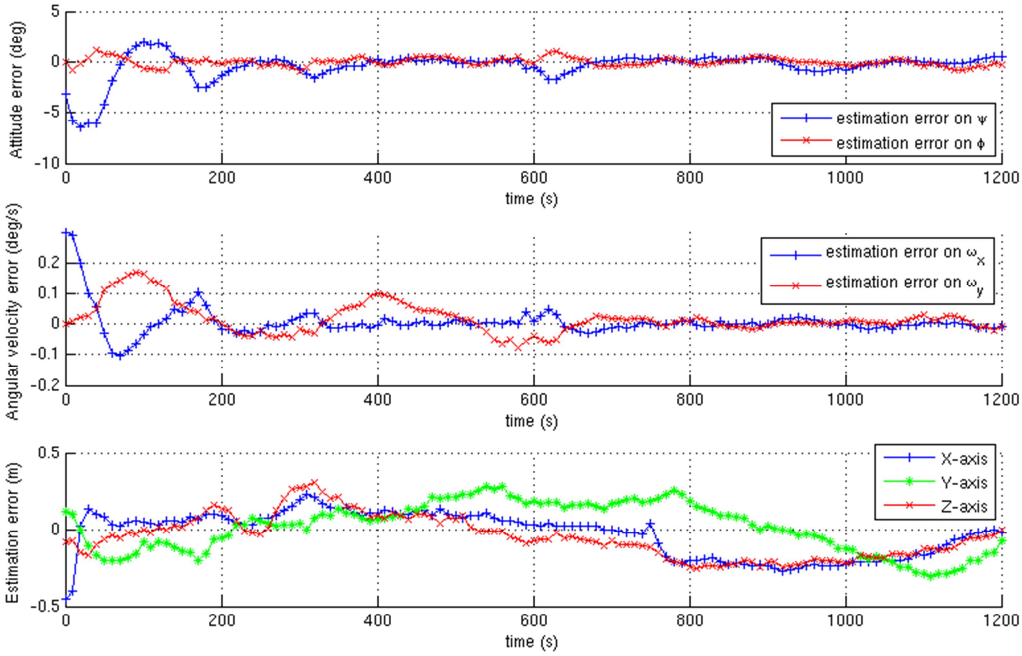

Figure 8: Estimation errors – Attitude, angular velocity and position for approach from 100m to 50m

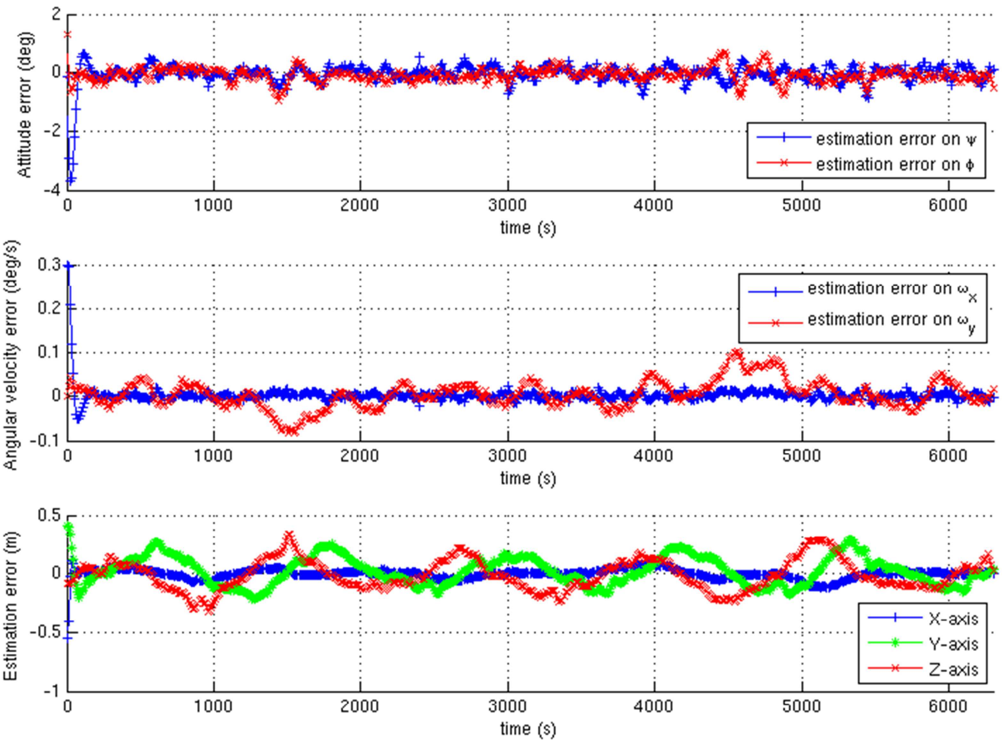

Figure 9: Estimation errors – Attitude, angular velocity and position for fly-around between 50m and 25m

The particular shape of the target makes the estimation of the roll angle very difficult. Results show that the error can be very high on this angle. Thus, only error on transverse angles and associated angular velocities are presented on the previous figures.

The simulations show that the navigation solution allows good convergence of the transverse angles and position. The initialization provided by the specific implemented solution is good enough to converge toward to correct attitude.

The error on the transverse angles is inferior to 0.5° after convergence when the chaser is in station-keeping 25m behind the target and inferior to 1° when the distance is equal to 50m. The linear approach scenario shows an estimation error decrease on attitude, which is interesting for our application, as we need a good performance when the target becomes closer. We can notice that position is estimated with an error up to 30cm, but does not decrease significantly when the distance is low. The fly-around simulation is run on 6300s, which is the duration of the maneuver. We see that it does not improve the results of the navigation in comparison with station-keeping simulations at 50m or 25m. It can be explained by the fact that the fly-around does not bring more visibility or information about the pose of the target in this case.

Improvements of the navigation filter could lead to a better roll estimation but we assume that the performance is principally wanted on transverse angles which are the most important ones if we want to capture the H10.

4. Conclusion and future work

A navigation technique has been presented to estimate both relative position and attitude of a space debris. This solution is formed of a navigation algorithm adapted to cloud of points obtained by LIDAR or stereovision sensors, a specific algorithm for attitude initialization and an Extended Kalman Filter.

The navigation algorithm uses a geometric model of the target to match the set of measured points. A simulator has been developed to obtain first results with H10 target, considering different trajectories and a realistic LIDAR sensor model. The attitude error stays below 0.5° when the distance between the two satellites is equal to 25m, while the position error is below 30cm along the three axes. These results show the advantage of using a navigation filter, which allows good performance even when there is not always a good target visibility and the level of performance we can expect when considering cylinder shaped debris. Furthermore, the considered initialization method allows good convergence of the ICP algorithm even when it is used at 100m. One way to improve the navigation solution would be to use the initialization method in parallel with the ICP algorithm to ensure that the result is correct.

The chaser’s absolute navigation is only used for changes of frames and does not have a significant impact on the relative navigation performance.

Future work on the subject will concern results with loss of visibility in last few meters and improvements of the filter’s tuning and capacities to estimate the roll angle correctly. The initialization method could be improved in order to be adapted to all kind of targets. Finally, research could be done about a navigation method which would not need a geometric model of the target as it would reconstruct it online from measurements.

References

[1] T. Tzschichholz, T. Boge, H. Benninghoff "A Flexible Image Processing Framework for vision-based navigation using monocular imaging sensors" *ESA GNC Conference 2011*

[2] A. Petit, N.Despré, E. Marchand, K. Kanani, F. Chaumette, S. Provost, G. FLandin "3D Model-based tracking for space autonomous rendezvous" *ESA GNC Conference 2011*

[3] T. Boge, H. Benninghoff, T. Tzschichholz "Hardware-in-the-loop rendezvous simulator using a vision based sensor" *ESA GNC Conference 2011*

[4] Farhad Aghili "Automated Rendezvous & docking (AR&D) without impact using a reliable 3D vision" *AIAA Guidance, Navigation and Control Conference 2010*

[5] S. Ruel, D. Ouellet, T. Luu, D. Laurendeau "Autonomous Tracking Initialization from Tridar data for autonomous rendezvous & docking" *Neptec Design Group*

[6] Matthew D. Lichter and Steven Dubowsky "Estimation of state, shape and inertial parameters of space objects from sequences of range images" *Proc. of SPIE vol. 5267 (Intelligent Robots and Computer Vision XXI : Algorithms, Techniques and Active Vision)*

[7] F. Schnitzer, A. Sonnenburg, K. Janschek, G. Willich "SLAM-based 3D shape estimation for rendezvous with uncooperative and unknown target spacecraft" *ESA GNC Conference 2011*

[8] Mohamed Okasha, Brett Newman "Relative Motion Guidance, Navigation and Control for Autonomous Orbital Rendezvous" *AIAA Guidance, Navigation and Control Conference 2011*

[9] Sean Augenstein, Stephen M. Rock "Simultaneous Estimation of Target Pose and Shape using the FastSLAM Algorithm" *AIAA Guidance, Navigation and Control Conference 2009*

[10] Fuyuto Terui, Heichachiro Kamimura, Shin'ichiro Nishida, "Motion estimation to a Failed Satellite on Orbit using Stereo Vision and 3D Model Matching", *9th Int. Conf. Control, Automation, Robotics and Vision, Singapore, 5-8th December 2006*

[11] Christopher D. Karlgaard, Hanspeter Schaub "Adaptive Nonlinear Huber–Based Navigation For Rendezvous in Elliptical Orbit" *AIAA 2010-7665, AAS/AIAA Astrodynamics Specialist Conference Toronto, Canada, August 2-5, 2010*