73rd International Astronautical Congress (IAC), Paris, France, 18-22 September 2022.

IAC-22.C1.IPB.38.x70501

Deep learning-based relative navigation about uncooperative space objects

Daniel Kaidanovic[a,*], Massimo Piazza[b,*], Michele Maestrini[c], Pierluigi Di Lizia[d]

[a] *Polytechnic University of Milan, 20156 Milan, Italy,* daniel.kaidanovic@mail.polimi.it
[b] *Infinite Orbits, 31000 Toulouse, France,* massimo@infiniteorbits.io
[c] *Polytechnic University of Milan, 20156 Milan, Italy,* michele.maestrini@polimi.it
[d] *Polytechnic University of Milan, 20156 Milan, Italy,* pierluigi.dilizia@polimi.it
[*] Corresponding author

Abstract

The space sector is nowadays experiencing an increase in the demand for missions involving proximity operations. This trend can be partially attributed to the raised awareness of the space debris problem, as an increasingly higher number of Active Debris Removal (ADR) and on-orbit servicing missions are being planned. Consequently, the need for an accurate onboard relative navigation system has become growingly relevant in the industry. This work proposes a software pipeline for spacecraft relative navigation, which leverages deep learning techniques to obtain the relative pose measurements and uses Kalman filtering to reconstruct the relative dynamics and to improve the robustness of the pipeline. Furthermore, a testing procedure involving a Blender-based spaceborne image generator has been devised and applied to validate the results in the case of a realistic image sequence of a rendezvous scenario.
The image processing pipeline is based on a Convolutional Neural Network architecture that scored excellent results in a pose estimation challenge organized by ESA. This architecture has demonstrated centimeter-level position accuracy and degree-level attitude accuracy, along with considerable robustness to changes in background and lighting conditions.
In order to reconstruct the relative state, a set of Kalman filters has been developed to tackle the attitude and position problems separately. For the relative distance, an Extended Kalman Filter has been implemented, as the underlying relative dynamics can be described by a linearized model. Instead, for the more complex attitude problem, the choice fell on an Unscented Kalman Filter due to its superior robustness to high non-linearities.
In addition, proving the robustness of the filtering algorithms was taken as a priority, with thousands of tests aimed at identifying and counteracting the most common failure modes. Moreover, some techniques were also developed for the detection and rejection of measurement outliers.
The whole navigation pipeline was then tested on a set of synthetic image sequences of the TANGO spacecraft in free tumbling conditions. The frames were obtained from a Blender-based spaceborne image generation platform, exploiting a 3D model of the target and relying on an accurate propagation of the relative dynamics.
The proposed filtering pipeline proved to substantially improve the accuracy of the raw deep learning-based measurements by leveraging sequential information, while also increasing the overall robustness.
Keywords: relative navigation, Kalman filter, pose estimation, deep learning, optical simulator, space debris

1. Introduction

The number of resident space objects orbiting Earth, and in particular the portion represented by space debris, have been subject to a steady growth since the beginning of the space age. The increasingly higher awareness and the concerns associated with this problem have recently led to a number of planned Active Debris Removal (ADR) and on-orbit servicing missions.
Since the approached bodies may be represented by uncooperative tumbling objects with potentially high angular velocities, the availability of a robust and accurate relative navigation system represents a key enabling technology in the aforementioned scenarios.
To this aim, two main paths may in principle be pursued:

- ground-based observations, for which one may resort to the use of radars, passive optical tracking or laser ranging
- on-orbit observations, by means of a sensor suite available onboard the chaser spacecraft, which can be used to determine the relative dynamics of the target body.

The former approach is unfortunately unsuitable for close proximity operations, because of the excessive uncertainty in the state of the uncooperative target.
To accurately estimate the relative pose of a target body, a visual approach has to be pursued. Thus, either Light Detection and Ranging (LiDAR) sensors or stereo/monocular cameras would be required.
LiDARs and stereo cameras are generally bulky and expensive solutions, while the use of monocular cameras represents a low Size, Weight and Power (SWAP) cost effective solution. They offer a great alternative, although

they require complex and computationally demanding image processing algorithms to properly work.
It is also important to highlight that the above mentioned procedure needs to be robust and accurate in order to deal with long approach sequences, in a wide range of lighting and background conditions.

The main goal of this work is to build and validate a robust filtering pipeline that leverages monocular pose measurements obtained using the deep learning-based approach presented in [1].
The resulting pose estimation software is then validated by means of a purposely developed `Blender`-based[1] rendezvous simulator, that is capable of rendering realistic image sequences of the TANGO target spacecraft as viewed from a chaser satellite.

In the remainder of this paper we will first of all introduce in Section 2. all the building blocks of our deep learning-based image processing software, as well as detailing the mathematical aspects of our EKF/UKF estimation framework and the associated dynamical models. In Section 3. the outline of our overall relative pose estimation pipeline is presented, including a description of the optical/orbital rendezvous simulator. This will be followed by a performance analysis of the pose estimation pipeline in Section 4. and we will eventually draw some conclusions in Section 5.

2. Theoretical background

2.1 Deep learning-based pose estimation

The state-of-the-art techniques used for estimating the pose of a spacecraft from a monocular image typically make use of an image-processing subsystem that identifies the position in the image frame of certain semantic features of the target spacecraft. This is followed by a pose solver consisting in a geometric optimization subsystem, that fits a known 3D model of the satellite to the features matched in the image.
Depending on the approach adopted for image processing, two main classes of monocular pose estimation methods may be identified: feature-based and deep learning-based pose estimation.
Feature-based methods [2, 3, 4] seek for correspondences between patterns of edges detected in the image and line segments of the known wireframe model of the spacecraft. The approach adopted in this work, leverages deep learning-based methods instead. The latter class of methods makes use of a Convolutional Neural Network (CNN) pipeline whose role may either consist in regressing the position in the image frame of predefined keypoints [1, 5, 6] that later become the input of a pose solver, or in a direct end-to-end estimation of the spacecraft pose [7, 8, 9, 10].

We will now briefly describe in Sections 2.1.1 and 2.1.2 the architecture of the two CNNs that are part of our image processing software used for measuring the pose of an uncooperative spacecraft.

2.1.1 Object detection via YOLOv5

The YOLO CNN architecture [11] is a state-of-the-art one-stage method for object detection. In particular, one of the most recent iterations of this model, named YOLOv5,[2] will be exploited in this work and it has been trained to detect the TANGO satellite in a 416×416 pixels image. The neural network is specifically trained to output the bounding box coordinates associated with the portion of the image containing the spacecraft. Based on this, further processing of the image will exclusively focus on the identified Region of Interest (RoI).

2.1.2 Landmark regression via HRNet

The unprecedented accuracy demonstrated by the High-Resolution Network (HRNet) architecture [12] in the field of human pose estimation led to the decision of implementing this model in our pipeline. This CNN processes the previously identified RoI and it has been trained to regress 11 heatmaps with a size of 416×416, corresponding to the 11 semantic keypoints chosen for the TANGO satellite. The final predicted landmark locations are then obtained as the individual peaks in each heatmap, which will appear as very sharp 2D pseudo-Gaussians.

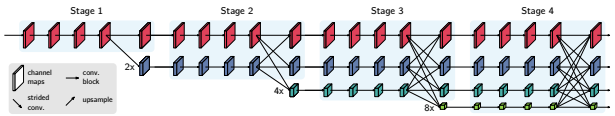


Fig. 1: Main body of the HRNet architecture

As can be seen from Figure 1, a peculiar aspect of HRNet is that, in contrast to conventional CNNs, it connects high-to-low resolution subnetworks in parallel rather than in series, thus retaining the highest-resolution representation throughout the whole inference process. This yields extremely high accuracy of the regressed landmark locations.

2.2 PnP problem

The Perspective-n-Point (PnP) problem consists in estimating the pose of an object, given a set of n points of the object itself with known (or estimated) 3D model coordinates, and given the corresponding 2D projections detected in the image, that are in our case the keypoint locations regressed by HRNet.
Pose estimation methods also leverage the knowledge of

[1] https://www.blender.org (accessed on August 29th 2022)

[2] https://github.com/ultralytics/yolov5 (accessed on August 20th 2022)

the camera intrinsics, so as to seek for the pose that yields the best fit between the resulting projection of the object's points and the corresponding detected keypoints in the image frame.
State-of-the-art PnP solvers are basically divided into two categories. Iterative solvers minimize a measure of the fit error between the projected model points and the image points. Multi-stage analytical approaches, such as the Efficient PnP (EPnP) method [13], will instead leverage a linearized form of the projection equations.
In this work, both the EPnP method and an iterative solver based on the Levenberg-Marquardt Method (LMM) are used in the pose solver that fits the measured pose to the set of detected landmarks in the RoI.

2.3 Relative translational dynamics

The relative distance problem consists in describing how the separation between two spacecrafts orbiting a main body evolves over time in the Local Vertical Local Horizontal (LVLH) reference frame.
In this frame, the relative distance $\mathbf{r}_r$ and the relative velocity $\mathbf{v}_r$ are defined as

$$\mathbf{r}_r = x\hat{\mathbf{r}} + y\hat{\boldsymbol{\theta}} + z\hat{\mathbf{h}} \tag{1}$$

$$\mathbf{v}_r = \dot{x}\hat{\mathbf{r}} + \dot{y}\hat{\boldsymbol{\theta}} + \dot{z}\hat{\mathbf{h}} \tag{2}$$

where $[\hat{\mathbf{r}}, \hat{\boldsymbol{\theta}}, \hat{\mathbf{h}}]$ are the unit vectors associated with the LVLH frame.
The evolution of the relative distance between two spacecrafts may then be expressed using the following set of differential equations [14]:

$$\begin{cases} \ddot{x} = 2\dot{\nu}\dot{y} + \ddot{\nu}y + \dot{\nu}^2 x + \dfrac{\mu}{\bar{r}^2} - \dfrac{\mu(\bar{r}+x)}{[(\bar{r}+x)^2+y^2+z^2]^{3/2}} \\ \ddot{y} = -2\dot{\nu}\dot{x} - \ddot{\nu}x + \dot{\nu}^2 y - \dfrac{\mu y}{[(\bar{r}+x)^2+y^2+z^2]^{3/2}} \\ \ddot{z} = -\dfrac{\mu z}{[(\bar{r}+x)^2+y^2+z^2]^{3/2}} \end{cases} \tag{3}$$

where μ indicates the standard gravitational parameter of the main attractor, ν is the true anomaly and $\bar{r}$ represents the distance between the centers of gravity of the main attractor and the chaser.
The complete formulation of the problem also requires to describe the orbital dynamics of the chaser in terms of true anomaly ν and position $\bar{r}$ as:

$$\ddot{\bar{r}} = \bar{r}\dot{\nu}^2 - \frac{\mu}{\bar{r}^2} \tag{4}$$

$$\ddot{\nu} = -\frac{2\dot{\bar{r}}\dot{\nu}}{\bar{r}} \tag{5}$$

2.4 Relative attitude dynamics

The general equations governing the attitude motion of a rigid body are known as the Euler equations [15], whose form is:

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \mathbf{M} \tag{6}$$

$\mathbf{J}$ represents the inertia matrix of the object, while $\boldsymbol{\omega}$ is its angular velocity and $\mathbf{M}$ indicates the sum of the applied torques.

For the aim of this dissertation, torque-free dynamics will be assumed by setting $\mathbf{M} = \mathbf{0}$, thus neglecting the effect of disturbance torques, whose integral effect would be quite low over the typical time scales of a rendezvous sequence.
The angular velocity notation proposed in [16] will be used in the remainder of this section, which is defined in Equations (7) to (9).

$$\boldsymbol{\omega}_r = \boldsymbol{\omega}_t - \boldsymbol{\Gamma}\boldsymbol{\omega}_c \tag{7}$$

$$\dot{\boldsymbol{\omega}}_r = \dot{\boldsymbol{\omega}}_t - \boldsymbol{\Gamma}\dot{\boldsymbol{\omega}}_c + \dot{\boldsymbol{\omega}}_{\text{app}} \tag{8}$$

$$\dot{\boldsymbol{\omega}}_{\text{app}} = \boldsymbol{\omega}_r \times \boldsymbol{\Gamma}\boldsymbol{\omega}_c \tag{9}$$

where $\boldsymbol{\omega}_t$ and $\boldsymbol{\omega}_c$ are respectively the target and chaser angular velocities expressed in their body-fixed reference frames, which are linked by a rotation matrix $\boldsymbol{\Gamma}$. $\boldsymbol{\omega}_r$ represents the relative angular velocity expressed in the target body frame and $\dot{\boldsymbol{\omega}}_{\text{app}}$ is the apparent angular acceleration.
Furthermore, the attitude parametrization that has been selected for this work is based on Modified Rodrigues Parameters (MRPs) $\boldsymbol{\zeta}$ and their shadow-set counterpart $\boldsymbol{\zeta}_S$ [17], which are respectively defined in Equations (10) and (11) in terms of the vector part of the attitude quaternion $\tilde{\mathbf{q}}$ and the quaternion's scalar component q_0.

$$\boldsymbol{\zeta} = \frac{\tilde{\mathbf{q}}}{1+q_0} \qquad \text{if } \|\boldsymbol{\zeta}\| < 1 \tag{10}$$

$$\boldsymbol{\zeta}_S = \frac{-\tilde{\mathbf{q}}}{1-q_0} \qquad \text{if } \|\boldsymbol{\zeta}\| > 1 \tag{11}$$

A relation connecting the MRPs with the aforementioned attitude matrix $\boldsymbol{\Gamma}$ is attained as

$$\boldsymbol{\Gamma}(\boldsymbol{\zeta}) = \mathbf{I}_3 - \alpha_1^{\text{A}}[\boldsymbol{\zeta}\times] + \alpha_2^{\text{A}}[\boldsymbol{\zeta}\times]^2 \tag{12}$$

where $\mathbf{I}_3$ is the identity matrix and the the following notation is used:

$$\begin{cases} \alpha_1^{\text{A}} := 4\dfrac{1-\boldsymbol{\zeta}^{\text{T}}\boldsymbol{\zeta}}{(1+\boldsymbol{\zeta}^{\text{T}}\boldsymbol{\zeta})^2} \\ \alpha_2^{\text{A}} := 8\dfrac{1}{(1+\boldsymbol{\zeta}^{\text{T}}\boldsymbol{\zeta})^2} \\ [\boldsymbol{\zeta}\times] := \begin{bmatrix} 0 & -\zeta_3 & \zeta_2 \\ \zeta_3 & 0 & -\zeta_1 \\ -\zeta_2 & \zeta_1 & 0 \end{bmatrix} \end{cases} \tag{13}$$

The evolution of the relative attitude can be eventually described in terms of MRPs and relative angular velocity as

$$\begin{cases} \dot{\boldsymbol{\zeta}}_r = \frac{1}{4}\boldsymbol{\Sigma}(\boldsymbol{\zeta}_r)\boldsymbol{\omega}_r \\ \mathbf{J}_t\dot{\boldsymbol{\omega}}_r + \boldsymbol{\omega}_r \times \mathbf{J}_t\dot{\boldsymbol{\omega}}_r = \mathbf{M}_{\text{app}} - \mathbf{M}_g - \mathbf{M}_{ci} \end{cases} \tag{14}$$

where $\boldsymbol{\Sigma}(\boldsymbol{\zeta}) := (1-\boldsymbol{\zeta}^{\mathrm{T}}\boldsymbol{\zeta})\mathbf{I}_3 + 2\boldsymbol{\zeta}\boldsymbol{\zeta}^{\mathrm{T}} + 2[\boldsymbol{\zeta}\times]$, while $\mathbf{J}_t$ is the inertia matrix of the target, $\boldsymbol{\zeta}_r$ is the MRP describing the relative attitude and the torque contributions are defined in Equations (15) to (17).

$$\mathbf{M}_{\text{app}} = \mathbf{J}_t\dot{\boldsymbol{\omega}}_r \times \boldsymbol{\Gamma}\boldsymbol{\omega}_c \tag{15}$$

$$\begin{aligned} \mathbf{M}_g = {} & \boldsymbol{\Gamma}\boldsymbol{\omega}_c \times \mathbf{J}_t\boldsymbol{\Gamma}\boldsymbol{\omega}_c + \ldots \\ & + (\boldsymbol{\omega}_r \times \mathbf{J}_t\boldsymbol{\Gamma}\boldsymbol{\omega}_c + \boldsymbol{\Gamma}\boldsymbol{\omega}_c \times \mathbf{J}_t\boldsymbol{\omega}_r) \end{aligned} \tag{16}$$

$$\mathbf{M}_{ci} = \mathbf{J}_t\boldsymbol{\Gamma}\dot{\boldsymbol{\omega}}_c \tag{17}$$

$\mathbf{M}_{\text{app}}$ represents the apparent torque, $\mathbf{M}_g$ is the gyroscopic torque and $\mathbf{M}_{ci}$ indicates the chaser-inertial torque.

2.5 Kalman filtering

The Kalman filter (KF) and its variants are among the most efficient and common recursive estimation techniques used in the aerospace field. In its basic form, the KF is defined as a predictor-corrector algorithm [18], as it is composed of both a prediction and a correction phase.
Let us consider a linear discrete-time system in the form

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k + \mathbf{w}_{(k)} \\ \mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \end{cases} \tag{18}$$

where $\mathbf{x}_k$ and $\mathbf{u}_k$ respectively represent the state and the input at time step k, $\mathbf{w}_k$ and $\mathbf{v}_k$ are used to identify the process and measurement noise, whereas $\mathbf{y}_k$ indicates the measurement output at time step k.
Let $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ denote the state transition, input transition and observation matrices, respectively.

The aforementioned prediction phase consists in propagating both the state and the related covariance $\hat{\mathbf{x}}_k^+$, $\mathbf{P}_k^+$, based on their values at the previous time step as follows:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}\hat{\mathbf{x}}_k^+ + \mathbf{G}\mathbf{u}_k \tag{19}$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}\mathbf{P}_k^+\mathbf{F}^{\mathrm{T}} + \mathbf{Q}_k \tag{20}$$

This yields the expected mean value of the state $\hat{\mathbf{x}}_{k+1}^-$ and the covariance matrix $\mathbf{P}_{k+1}^-$ for the current time step.
Note that $\mathbf{Q}_k$ represents the covariance matrix associated with process noise, while $\mathbf{R}_k$ is associated with measurement noise.
The following step, which corresponds to Equations (21) to (23), is called a correction step, as it corrects the predicted state and the covariance matrix. It does so by exploiting the incoming measurements and an optimal weighting factor called the Kalman gain $\mathbf{K}_{k+1}$, whose expression can be derived using a minimum-variance approach.

$$\mathbf{K}_{k+1} := \mathbf{P}_{k+1}^-\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}_{k+1}^-\mathbf{H}^{\mathrm{T}} + \mathbf{R}_{k+1})^{-1} \tag{21}$$

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{H}\hat{\mathbf{x}}_{k+1}^-) \tag{22}$$

$$\mathbf{P}_{k+1}^+ = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}_{k+1}^- \tag{23}$$

Note that the output of the correction step becomes the input of the successive filter iteration, as the filter uses recursive information to improve its guesses.

Let us now introduce in Sections 2.5.1 and 2.5.2 the variants of the Kalman filter that are used in the estimation framework developed in this work.

2.5.1 Extended Kalman Filter

As the linear KF cannot deal with non-linear problems, some alternative architectures had to be devised.
A common solution to this issue is the Extended Kalman Filter (EKF) [19], which exploits a linearization of the problem to get around the present non-linearities.
Given the generic non-linear model in Equation (24)

$$\begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \\ \mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k) \end{cases} \tag{24}$$

the Jacobian matrices of the state transition function $f(\cdot)$ and the measurement function $h(\cdot)$ can be computed as:

$$\mathbf{F} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k^+} \qquad \mathbf{H} = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k^-} \tag{25}$$

Once the Jacobians are available, the process becomes analogous to the linear KF. The resulting prediction-correction steps are reported in Equations (26) to (30).

$$\hat{\mathbf{x}}_{k+1}^- = f(\hat{\mathbf{x}}_k^+, \mathbf{u}_k, 0) \tag{26}$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}\mathbf{P}_k^+\mathbf{F}^{\mathrm{T}} + \mathbf{Q}_k \tag{27}$$

$$\mathbf{K}_{k+1} := \mathbf{P}_{k+1}^-\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}_{k+1}^-\mathbf{H}^{\mathrm{T}} + \mathbf{R}_{k+1})^{-1} \tag{28}$$

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - h(\hat{\mathbf{x}}_{k+1}^-, 0)) \tag{29}$$

$$\mathbf{P}_{k+1}^+ = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}_{k+1}^- \tag{30}$$

2.5.2 Unscented Kalman Filter

In the event of highly non-linear systems the EKF may fall short, thus requiring a different filtering architecture.
By leveraging the so called unscented transform for the propagation of state and covariance, the Unscented Kalman Filter (UKF) [20] can be defined.
Considering the generic nonlinear system in Equation (24), a set of sigma points can be sampled from a

normally distributed state. A typical algorithm for the sampling of sigma points $\hat{\mathbf{x}}_k^{(\mathrm{i})}$ consists in selecting them as follows:

$$\hat{\mathbf{x}}_k^{(\mathrm{i})} = \hat{\mathbf{x}}_k^{+} + \tilde{\mathbf{x}}^{(\mathrm{i})}, \qquad i = 1, \ldots, 2n \tag{31}$$

$$\tilde{\mathbf{x}}^{(\mathrm{i})} = \left(\sqrt{c\mathbf{P}_k^{+}}\right)_i^{\mathrm{T}}, \qquad i = 1, \ldots, n \tag{32}$$

$$\tilde{\mathbf{x}}^{(\mathrm{n+i})} = -\left(\sqrt{c\mathbf{P}_k^{+}}\right)_i^{\mathrm{T}}, \qquad i = 1, \ldots, n \tag{33}$$

where c is a tuning parameter used to define the spread and weight of the sigma points.
The latter are then propagated and a weighted average is computed to define the predicted state and covariance using Equations (34) to (36).

$$\hat{\mathbf{x}}_{(k+1)}^{(\mathrm{i})} = f(\hat{\mathbf{x}}_k^{(\mathrm{i})}, \mathbf{u}_{k+1}) \tag{34}$$

$$\hat{\mathbf{x}}_{k+1}^{-} = \sum_{i=0}^{2n} W_M^{(\mathrm{i})} \hat{\mathbf{x}}_{k+1}^{(\mathrm{i})} \tag{35}$$

$$\mathbf{P}_{k+1}^{-} = \sum_{i=0}^{2n} W_c^{(\mathrm{i})} (\hat{\mathbf{x}}_{k+1}^{(\mathrm{i})} - \hat{\mathbf{x}}_{k+1}^{-})(\hat{\mathbf{x}}_{k+1}^{(\mathrm{i})} - \hat{\mathbf{x}}_{k+1}^{-})^{\mathrm{T}} + \mathbf{Q}_k \tag{36}$$

where $W_M^{(\mathrm{i})}$ and $W_c^{(\mathrm{i})}$ are weighting parameters.
The obtained predicted state and covariance are then used to correct the sigma points as:

$$\hat{\mathbf{x}}_{k+1}^{(\mathrm{i})} = \hat{\mathbf{x}}_{k+1}^{-} + \tilde{\mathbf{x}}^{(\mathrm{i})}, \qquad i = 1, \ldots, 2n \tag{37}$$

$$\tilde{\mathbf{x}}^{(\mathrm{i})} = \left(\sqrt{c\mathbf{P}_{k+1}^{-}}\right)_i^{\mathrm{T}}, \qquad i = 1, \ldots, n \tag{38}$$

$$\tilde{\mathbf{x}}^{(\mathrm{n+i})} = -\left(\sqrt{c\mathbf{P}_{k+1}^{-}}\right)_i^{\mathrm{T}}, \qquad i = 1, \ldots, n \tag{39}$$

The new set of sigma points is then used in the correction step to define: the expected measurement $\hat{\mathbf{y}}_{(k+1)}^{-}$, the measurement covariance $\mathbf{P}_y$ and the cross covariance $\mathbf{P}_{xy}$.
This is done as follows:

$$\hat{\mathbf{y}}_{k+1}^{(\mathrm{i})} = h(\hat{\mathbf{x}}_{k+1}^{(\mathrm{i})}) \tag{40}$$

$$\hat{\mathbf{y}}_{(k+1)}^{-} = \sum_{i=1}^{2n} W_M^{(\mathrm{i})} \hat{\mathbf{y}}_{k+1}^{(\mathrm{i})} \tag{41}$$

$$\mathbf{P}_y = \sum_{i=1}^{2n} W_c^{(\mathrm{i})} (\hat{\mathbf{y}}_{k+1}^{(\mathrm{i})} - \hat{\mathbf{y}}_{k+1}^{-})(\hat{\mathbf{y}}_{k+1}^{(\mathrm{i})} - \hat{\mathbf{y}}_{k+1}^{-})^{\mathrm{T}} + \mathbf{R}_k \tag{42}$$

$$\mathbf{P}_{xy} = \sum_{i=1}^{2n} W_c^{(\mathrm{i})} (\hat{\mathbf{x}}_{k+1}^{(\mathrm{i})} - \hat{\mathbf{x}}_{k+1}^{-})(\hat{\mathbf{y}}_{k+1}^{(\mathrm{i})} - \hat{\mathbf{y}}_{k+1}^{-})^{\mathrm{T}} \tag{43}$$

The obtained quantities are then used to define a Kalman gain and correct the state and covariance as:

$$\mathbf{K}_{k+1} = \mathbf{P}_{xy}\mathbf{P}_y^{-1} \tag{44}$$

$$\hat{\mathbf{x}}_{k+1}^{+} = \hat{\mathbf{x}}_{k+1}^{-} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^{-}) \tag{45}$$

$$\mathbf{P}_{k+1}^{+} = \mathbf{P}_{k+1}^{-} - \mathbf{K}_{k+1}\mathbf{P}_y\mathbf{K}_{k+1}^{\mathrm{T}} \tag{46}$$

3. Relative pose estimation pipeline

3.1 Synthetic image generation

A Blender-based architecture has been proposed for rendering realistic images of the TANGO spacecraft, whose 3D model was kindly provided by OHB Sweden.

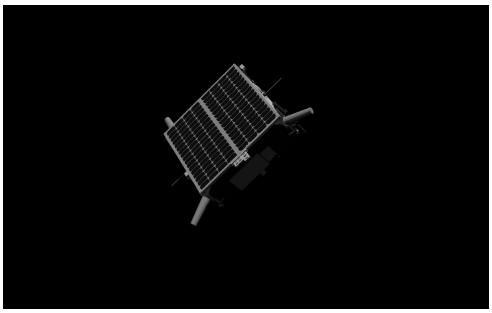


Fig. 2: Example of a generated frame

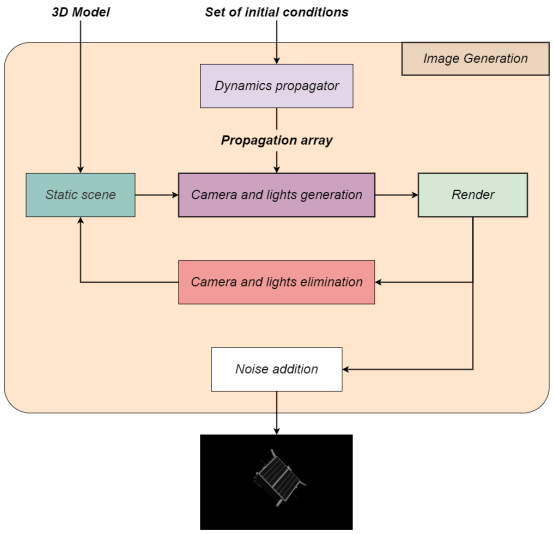


Fig. 3: Architecture of the image generation software

The pipeline, shown in Figure 3, starts by determining the relative position of the camera with respect to the target body frame. It then generates a virtual camera in the scene and orients it based on the provided attitude matrix. Afterwards, depending on the orbital position of the chaser and the considered season, a set of custom lights is placed in the scene to mimic realistic lighting conditions.
Once all components are put in place, a frame is rendered

using the `EEVEE` engine. A Gaussian noise filter is then applied and the image can eventually be saved in grayscale format.
The pose labels can either be provided for a static dataset of random poses or for a coherent sequence. This is implemented as two different rendering modes, which are both needed for testing and validation purposes.

- Static dataset: the relative attitude and position data is provided in a pre-generated `.json` file and the pipeline treats each frame individually.
- Image sequence: the initial relative pose, angular velocity and orbit are provided along with a sequence length and the related number of frames per second (FPS). The sequence is then propagated in time using Equations (3) and (14).

This entire process is fully automated and scripted via Python bindings of the Blender API.

3.2 *Image processing*

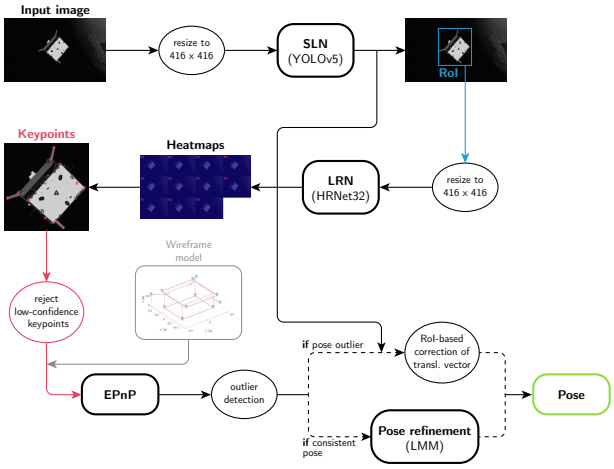


Fig. 4: Architecture of the image processing pipeline at inference time

The outline of the architecture of the image processing pipeline is represented in Figure 4 and it consists of three main subsystems.
The first subsystem, called the Spacecraft Localization Network (SLN) and described in Section 2.1.1, is responsible for identifying the RoI in the image. It is followed in the pipeline by the Landmark Regression Network (LRN), that is detailed in Section 2.1.2, whose role is to detect semantic keypoints of the target spacecraft inside the RoI.
The third and last subsystem is the pose solver, which uses the techniques introduced in section 2.2 and, given the landmarks identified by LRN, seeks for the corresponding best pose fit based on the known wireframe model of the target. More specifically, it first runs the EPnP algorithm to obtain an initial estimate of the pose and, in a nominal situation (i.e. if no pose outlier is detected), it successively refines the initial solution using the LMM.

3.3 *Pose filtering*

The filtering pipeline, consists in a KF-based architecture whose aim is to improve the raw pose output obtained from the CNN-based image processing pipeline.

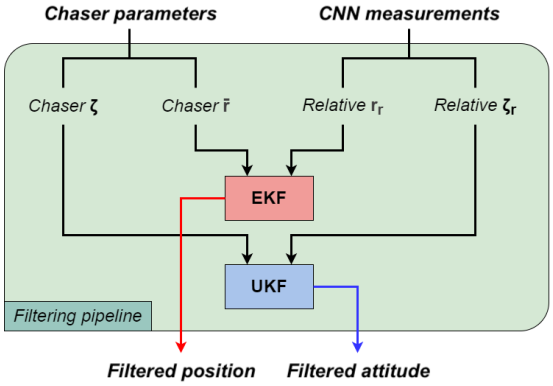


Fig. 5: Architecture of the pose filtering pipeline

As can be seen from Figure 5, it was chosen to separate and apply different filtering techniques to the relative distance and the relative attitude problem. This is also based on the assumption that the attitude of the chaser is perfectly known and available.
This approach was pursued as the relative attitude is characterized by a strongly non-linear behavior, as compared to the simpler relative distance problem. In particular, the EKF was applied to the simpler and more linear relative separation problem, whereas the UKF was identified as the most suitable option to tackle the more complex relative attitude problem.
For both the EKF and the UKF, the process covariance $\mathbf{Q}$ has been assumed null, while the measurement covariance $\mathbf{R}$ has been carefully tuned, based on the error statistics obtained on SPEED by the image processing pipeline.
The EKF uses a different translation measurement covariance depending on the relative distance, because of the strong correlation. In particular three main sections have been defined (0 to 15, 15 to 25 and > 25 meters), each having a different $\mathbf{R}$ value.
For what concerns the UKF attitude measurement covariance, a weak correlation between the error and the MRP norm has been detected, which led to conservatively adopt a single covariance computed only over measurements that resulted into $\|\boldsymbol{\zeta}\| > 0.4$.

In addition, the UKF also incorporates an outlier rejection algorithm, based on the evaluation of the Mahalanobis distance between expected and actual measurements obtained from the image processing pipeline. If such a distance exceeds the 2σ threshold of the corresponding fitted inverse χ^2 distribution, then the

update step of the UKF will be temporarily disabled.

3.3.1 EKF specifics

The Jacobians defined in Equation (25) can be analytically derived for the relative separation problem introduced in Section 2.3. The resulting expressions are given in Equations (47) and (48)

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial y} & \frac{\partial \ddot{x}}{\partial z} & 0 & \frac{\partial \ddot{x}}{\partial \dot{y}} & 0 \\ \frac{\partial \ddot{y}}{\partial x} & \frac{\partial \ddot{y}}{\partial y} & \frac{\partial \ddot{y}}{\partial z} & \frac{\partial \ddot{y}}{\partial \dot{x}} & 0 & 0 \\ \frac{\partial \ddot{z}}{\partial x} & \frac{\partial \ddot{z}}{\partial y} & \frac{\partial \ddot{z}}{\partial z} & 0 & 0 & 0 \end{bmatrix} \tag{47}$$

$$\mathbf{H} = \begin{bmatrix} & & & 0 & 0 & 0 \\ & \mathbf{A}_{C/L} & & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \end{bmatrix} \tag{48}$$

where the direction cosine matrix $\mathbf{A}_{C/L}$ represents the camera to LVLH frame rotation, while the partial derivative terms in Equation (47) can be explicitly written as follows.

Let $r_c := [(\bar{r}+x)^2 + y^2 + z^2]^{3/2}$, then:

$$\frac{\partial \ddot{x}}{\partial x} = \frac{\mu}{r_c^3}\left[3\left(\frac{\bar{r}+x}{r_c}\right)^2 - 1\right] + \dot{\nu}^2 \tag{49}$$

$$\frac{\partial \ddot{x}}{\partial y} = 3\mu\left(\frac{\bar{r}+x}{r_c^5}\right)y - 2\dot{\nu}\frac{\dot{\bar{r}}}{\bar{r}} \tag{50}$$

$$\frac{\partial \ddot{x}}{\partial z} = 3\mu\left(\frac{\bar{r}+x}{r_c^5}\right)z \tag{51}$$

$$\frac{\partial \ddot{x}}{\partial \dot{y}} = 2\dot{\nu} \tag{52}$$

$$\frac{\partial \ddot{y}}{\partial x} = 3\mu\left(\frac{\bar{r}+x}{r_c^5}\right)y + 2\dot{\nu}\frac{\dot{\bar{r}}}{\bar{r}} \tag{53}$$

$$\frac{\partial \ddot{y}}{\partial y} = \frac{\mu}{r_c^3}\left[3\left(\frac{y}{r_c}\right)^2 - 1\right] + \dot{\nu}^2 \tag{54}$$

$$\frac{\partial \ddot{y}}{\partial z} = 3\mu\left(\frac{y}{r_c^5}\right)z \tag{55}$$

$$\frac{\partial \ddot{y}}{\partial \dot{x}} = -2\dot{\nu} \tag{56}$$

$$\frac{\partial \ddot{z}}{\partial x} = 3\mu\left(\frac{\bar{r}+x}{r_c^5}\right)z \tag{57}$$

$$\frac{\partial \ddot{z}}{\partial y} = 3\mu\left(\frac{y}{r_c^5}\right)z \tag{58}$$

$$\frac{\partial \ddot{z}}{\partial z} = \frac{\mu}{r_c^3}\left[3\left(\frac{z}{r_c}\right)^2 - 1\right] \tag{59}$$

3.3.2 UKF specifics

The weighting parameters introduced in Equations (35) and (36), can be written in terms of three tuning parameters (α, β, κ) as follows:

$$W_M^{(0)} = 1 - \frac{n}{c} \qquad W_M^{(i)} = \frac{1}{2c} \tag{60}$$

$$W_c^{(0)} = (2 - \alpha^2 + \beta) - \frac{n}{c} \qquad W_c^{(i)} = \frac{1}{2c} \tag{61}$$

where $c = \alpha^2(n+\kappa)$. The values were eventually tuned as $\alpha = 10^{-3}$, $\beta = 0$, $\kappa = 2$.

An important consequence associated with the use both standard MRPs and their shadow-set counterpart is the need to accordingly transform covariances. In particular the shadow-set covariance can be retrieved as

$$\mathbf{P}_k^{\mathrm{S}} = \mathbf{\Lambda}\mathbf{P}_k\mathbf{\Lambda}^{\mathrm{T}} \quad \text{where } \mathbf{\Lambda} := 2\boldsymbol{\zeta}^{-4}(\boldsymbol{\zeta}\boldsymbol{\zeta}^{\mathrm{T}}) - \boldsymbol{\zeta}^{-2}\mathbf{I} \tag{62}$$

4. Results

4.1 Image generation & processing

A first very relevant step, that is preliminary to the testing campaign of our full pipeline, consists in the validation of the image generation procedure. To this aim, a replica of the SPEED dataset [21] has been created in order to compare the CNN inference results obtained on the original dataset and its replica. Only the training set portion can actually be exploited for comparison, given that test labels are not available to the public. As a consequence the original training set has been re-partitioned in order to train CNNs only on a subset, so that the remaining labels are available for testing purposes.

Minor a-priori biases were inevitable, due to slight differences in the 3D model of TANGO and because of the unavailability of lighting information in the original dataset. Nonetheless, the comparison resulted into a very close match in terms of pose estimation performance attained by the image processing pipeline, thus validating our approach. The corresponding results are reported in Table 1, where $\mathbf{E}_t$ is the translation error, E_q is the quaternion error, while $\mathbf{E}_\theta$ indicates the Euler angle error.

Table 1: CNN-based pose estimation performance obtained on SPEED and our replica

	SPEED (original)	**SPEED replica**
	Mean pose estimation error	
E_t	10.36 cm	10.10 cm
$\mathbf{E}_t$	[0.52 0.56 10.25] cm	[0.58 0.65 9.96] cm
E_q	2.24°	2.32°
$\mathbf{E}_\theta$	[1.57° 0.84° 1.72°]	[1.65° 0.83° 1.65°]
	Standard deviation of the error	
$\sigma_{\mathbf{E}_t}$	[1.62 1.71 30.44] cm	[1.79 2.45 39.82] cm
$\sigma_{\mathbf{E}_\theta}$	[8.92° 5.11° 10.82°]	[9.41° 4.72° 9.91°]

A few prediction visualization examples of the pose estimation results obtained by our CNN-based image processing pipeline are reported in Figure 6. These results have been obtained on test images from the SPEED dataset.

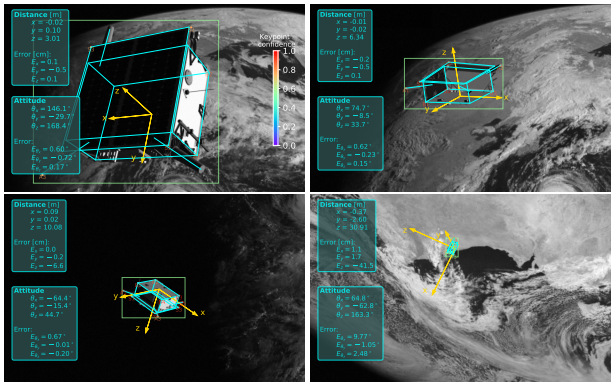


Fig. 6: CNN-based prediction visualization examples of SPEED test images

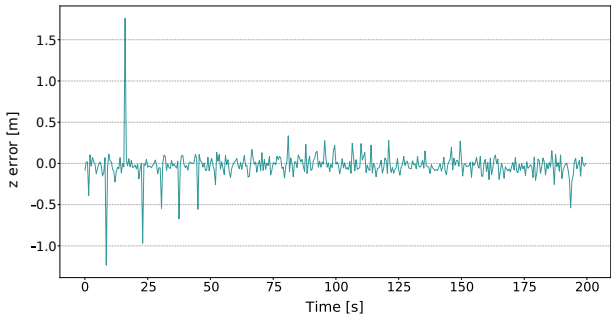
4.2 Pose filtering

Following this, a set of dynamic sequences were generated and evaluated by the CNN pipeline. This process was performed in order to provide further validation, and to produce a set of real input baselines to test the filters. The obtained results are reported in Table 2.

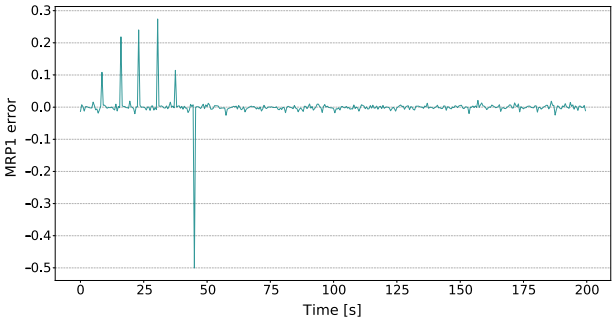
Table 2: CNN-based pose estimation performance on rendered rendezvous image sequence

	Mean pose estimation error
E_t	8.36 cm
$\mathbf{E}_t$	[0.50 0.48 8.26] cm
E_q	2.74°
$\mathbf{E}_\theta$	[1.47° 0.69° 2.47°]

Figures 7 and 8 report the results obtained by our pose estimation pipeline on a 200 second long rendezvous image sequence, in correspondence of a tumbling rate of 36 deg/s and an acquisition rate of 2 FPS.

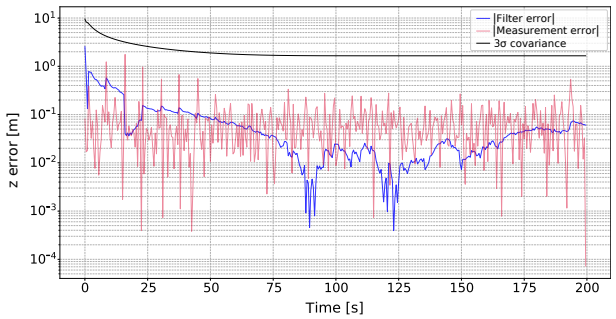


(a) Translation error (boresight component)

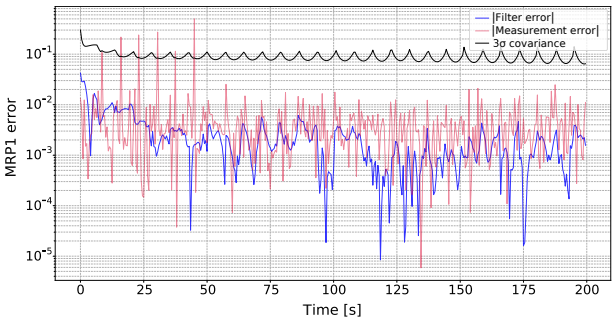


(b) Rotation error (ζ_1 component)

Fig. 7: Measurement errors of the CNN pipeline before filtering (rendezvous sequence)



(a) EKF performance (boresight component)



(b) UKF performance (ζ_1 component)

Fig. 8: Filtering performance and comparison with measurement error (rendezvous sequence)

As highlighted in Figure 7, some of the raw pose measurements resulted into outliers, which are all successfully filtered out as can be clearly seen from Figure 8. Besides rejecting and correcting outliers, our EKF/UKF approach also shows to steadily improve estimation accuracy by as much as one order of magnitude, upon reaching convergence.

Despite the promising results, it was chosen to further evaluate the robustness and reliability of the filtering pipeline.
This was accomplished by performing thousands of runs on pseudo-measurements, whose error statistics is based on the covariance analysis described in Section 3.3.

The EKF proved to be extremely reliable and robust, with zero failures across thousands of tests. The UKF was tested over a certain range of tumbling rates of the target spacecraft and it has demonstrated excellent performance in a wide range of scenarios, even though a number of failures were detected. As expected, for a given fixed number of FPS = 2, the failure rate becomes increasingly higher as the tumbling rate increases, as reported in Table 3.

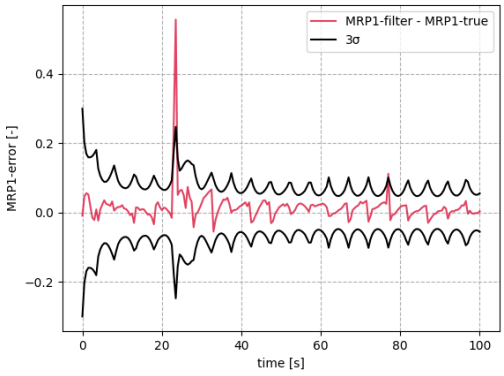
Table 3: Results of UKF stress testing campaign

Tumbling rate	FPS	No. tests	No. UKF failures
24 deg/s	2	1000	24
36 deg/s	2	1000	39
72 deg/s	2	1000	83

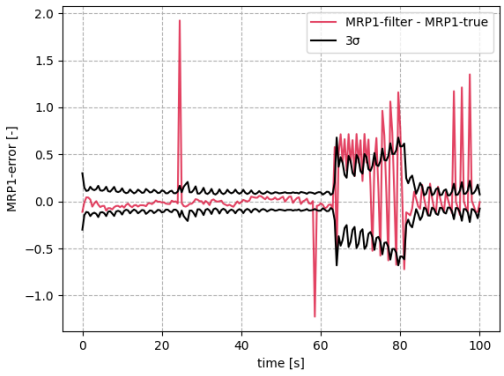
4.3 UKF failure analysis

To better understand the underlying mechanisms triggering the observed failures of the UKF, a dedicated analysis was carried out, which led to the identification of four main failure modes that have been carefully studied:

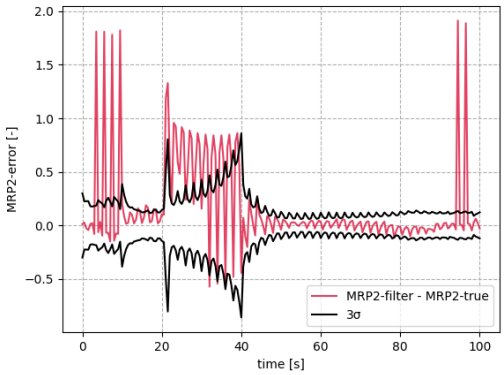
- *localized spike* (≈ 65% of the failures), consists in an isolated sudden increase of the error followed by its instantaneous falloff (see Figure 9a)
- *extended failure* (≈ 30% of the failures), consists in a prolonged failure caused by a sequence of successive outliers (see Figure 9b)
- *initial failure*, a subset of the extended failure scenario, occurring in correspondence of the very first UKF iterations (see Figure 9c)
- *total failure* (≈ 5% of the failures), consists in the complete divergence of the filter. It is generally caused by an extended sequence of outliers or a very bad initial guess (see Figure 9d).



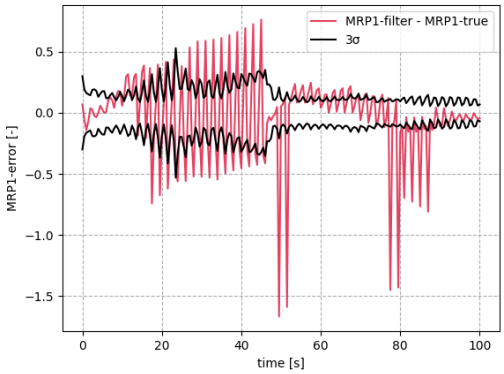
(a) Localized spike



(b) Extended failure



(c) Initial failure



(d) Total failure

Fig. 9: Examples of UKF failure scenarios

5. Conclusions

In this paper, a robust and accurate spacecraft relative pose estimation pipeline has been presented and validated in a rendezvous scenario. The proposed architecture leverages static pose measurements obtained via CNN-based image processing of monocular grayscale images. The CNN processing is integrated with an EKF/UKF filtering pipeline that exploits sequential information to improve the robustness and to accurately reconstruct the relative dynamics.

To this aim, a Blender-based optical simulator has been developed, which enables generating both still frame datasets and coherent rendezvous sequences.

A stress testing campaign that leverages pseudo-pose-measurements has been carried out, which revealed a minor number of occurrences during which the UKF may fail at converging, mostly in correspondence of extremely high tumbling rates. On the contrary, the EKF performed flawlessly through the entire range of tested relative angular velocities and zero failure scenarios were identified.

References

[1] Massimo Piazza, Michele Maestrini, and Pierluigi Di Lizia. Monocular relative pose estimation pipeline for uncooperative resident space objects. *Journal of Aerospace Information Systems*, pages 1–20, 2022.

[2] Simone D'Amico, Mathias Benn, and John L Jørgensen. Pose estimation of an uncooperative spacecraft from actual space imagery. *International Journal of Space Science and Engineering 5*, 2(2):171–189, 2014.

[3] Sumant Sharma, Jacopo Ventura, and Simone D'Amico. Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous. *Journal of Spacecraft and Rockets*, 55(6):1414–1429, 2018.

[4] Vincenzo Capuano, Shahrouz Ryan Alimo, Andrew Q Ho, and Soon-Jo Chung. Robust features extraction for on-board monocular-based spacecraft pose acquisition. In *AIAA Scitech 2019 Forum*, page 2005, 2019.

[5] Tae Ha Park, Sumant Sharma, and Simone D'Amico. Towards robust learning-based pose estimation of noncooperative spacecraft. *arXiv preprint arXiv:1909.00392*, 2019.

[6] Bo Chen, Jiewei Cao, Alvaro Parra, and Tat-Jun Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[7] Pedro F Proença and Yang Gao. Deep learning for spacecraft pose estimation from photorealistic rendering. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6007–6013. IEEE, 2020.

[8] Sumant Sharma, Connor Beierle, and Simone D'Amico. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In *2018 IEEE Aerospace Conference*, pages 1–12. IEEE, 2018.

[9] S Sharma, C Beierle, and S D'Amico. Towards pose determination for non-cooperative spacecraft using convolutional neural networks. In *Proceedings of the 1st IAA Conference on Space Situational Awareness (ICSSA)*, pages 1–5, 2017.

[10] Sumant Sharma and Simone D'Amico. Neural network-based pose estimation for noncooperative spacecraft rendezvous. *IEEE Transactions on Aerospace and Electronic Systems*, 2020.

[11] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[12] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[13] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.

[14] John L Junkins and Hanspeter Schaub. *Analytical mechanics of space systems*. American Institute of Aeronautics and Astronautics, 2009.

[15] F Landis Markley and John L Crassidis. *Fundamentals of spacecraft attitude determination and control*, volume 1286. Springer, 2014.

[16] Mauro Massari and Mattia Zamaro. Application of sdre technique to orbital and attitude control of spacecraft formation flying. *Acta Astronautica*, 94(1):409–420, 2014.

[17] Hanspeter Schaub, John L Junkins, et al. Stereographic orientation parameters for attitude dynamics: A generalization of the rodrigues parameters. *Journal of the Astronautical Sciences*, 44(1):1–19, 1996.

[18] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[19] O. Montenbruck and E. Gill. Satellite orbits: Models, methods, and applications. *Applied Mechanics Reviews*, 55(2), 04 2002.

[20] Simon J Julier, Jeffrey K Uhlmann, and Hugh F Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of 1995 American Control Conference-ACC'95*, volume 3, pages 1628–1632. IEEE, 1995.

[21] Mate Kisantal, Sumant Sharma, Tae Ha Park, Dario Izzo, Marcus Märtens, and Simone D'Amico. Satellite pose estimation challenge: Dataset, competition design and results. *IEEE Transactions on Aerospace and Electronic Systems*, 2020.