Acta Astronautica 220 (2024) 126–140

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

AI-based monocular pose estimation for autonomous space refuelling

Duarte Rondao [*,1], Lei He [2], Nabil Aouf [3]

City, University of London, ECV1 0HB London, United Kingdom

ARTICLE INFO

Keywords:
AI
Deep learning
Spacecraft
Navigation
Docking and berthing

ABSTRACT

Cameras are rapidly becoming the choice for on-board sensors towards space rendezvous due to their small form factor and inexpensive power, mass, and volume costs. When it comes to docking, however, they typically serve a secondary role, whereas the main work is done by active sensors such as lidar. This paper documents the development of a proposed AI-based (artificial intelligence) navigation algorithm intending to mature the use of on-board visible wavelength cameras as a main sensor for docking and on-orbit servicing (OOS), reducing the dependency on lidar and greatly reducing costs. Specifically, the use of AI enables the expansion of the relative navigation solution towards multiple classes of scenarios, e.g., in terms of targets or illumination conditions, which would otherwise have to be crafted on a case-by-case manner using classical image processing methods. Multiple convolutional neural network (CNN) backbone architectures are benchmarked on synthetically generated data of docking manoeuvres with the International Space Station (ISS), achieving position and attitude estimates close to 1 % range-normalised and 1 deg, respectively, an established rule of thumb for the navigation measurement accuracy during final approach. The integration of the solution with a physical prototype of the refuelling mechanism is validated in laboratory using a robotic arm to simulate a berthing procedure.

1. Introduction

For the majority of the 64-year history of space launches, satellites have been seen as an expendable medium: once the propellant is depleted, the mission is ended. Northrop Grumman's Mission Extension Vehicle (MEV) programme has recently challenged this paradigm by achieving the first teleoperated on-orbit servicing (OOS) to reposition existing spacecraft. This has opened up a new market segment where the spacecraft's life cycle can be extended beyond its original planning, avoiding the costs of launching, manufacturing, and keeping a new one. The success of the mission has attracted the attention of the United States Department of Defence, which has awarded the company a contract to study the possibility of servicing commercial and government satellites using robotics technology [1].

Despite its breakthrough, it is incontrovertible that the MEV was built on the shoulders of previous demonstrators: for example, the Kepler Automated Transfer Vehicle's (ATV) first refuelling operation of the International Space Station (ISS) to supply the station's thrusters in 2011 [2], or NASA's Robotic Refuelling Mission which demonstrated the technology to refuel satellites in orbit by robotic means in 2014 [3]. More recently, ESA has also recognised the potential for this new market segment by opening up calls for ideas related to OOS, having previously invested M€50 in support for research and development of relevant technologies [4]. Overall, OOS and manufacturing alone is projected to have a cumulative global market size of over B$4.4 by 2030 [5]. Still, existing satellites were, and still are, built without thinking of their serviceability and, more specifically, refuelling, which is a fulcral part of the servicing operations of these assets and represents a significant cost saving measure.

Paramount to the safe accomplishment of refuelling, and OOS in general, is the estimation of the relative states between the service vehicle (SV) and the client or target vehicle (TV) during docking or berthing. At such small distances, this entails the estimation of the six degree-of-freedom (DOF) relative pose, which is typically achieved with two types of optical sensors: lidar and camera sensors [6]. However, current flight-proven solutions using either sensor require optical corner-cube reflectors to be mounted on the TV [7]. The viability of large-scale OOS involves rendezvous and docking or berthing (RVD/B) via autonomous navigation with minimal or no human input, and the

[*] Correspondence to: Department of Electrical and Electronic Engineering, United Kingdom.
E-mail address: duarte.rondao@city.ac.uk (D. Rondao).

[1] Postdoctoral Research Fellow.

[2] Former Postdoctoral Research Fellow with the Department of Electrical and Electronic Engineering (currently a PhD candidate at Northwestern Polytechnical University, Xi'an, China).

[3] Professor of Robotics and Autonomous Systems with the Department of Electrical and Electronic Engineering.

https://doi.org/10.1016/j.actaastro.2024.04.003
Received 15 November 2023; Received in revised form 29 February 2024; Accepted 2 April 2024
Available online 6 April 2024

List of Acronyms

AI	artificial intelligence
ASMIL	Autonomous Systems and Machine Intelligence Laboratory
ATV	Automated Transfer Vehicle
BiLSTM	bi-directional long-short-term memory
CAD	computer aided design
CLAHE	contrast limited adaptive histogram equalisation
CNN	convolutional neural network
DL	deep learning
DNN	deep neural network
DOF	degree-of-freedom
DRCNN	deep recurrent convolutional neural network
FC	fully connected
FOV	field of view
GAP	global average pooling
GPU	graphics processing unit
IP	image processing
ISS	International Space Station
LSTM	long-short-term memory
MEV	Mission Extension Vehicle
ML	machine learning
OIBAR	Orbital AI-based Autonomous Refuelling
OOS	on-orbit servicing
P*n*P	perspective-*n*-point
PI	proportional integral
RGB	red-green-blue
RNN	recurrent neural network
RV	rendezvous
RVD/B	rendezvous and docking or berthing
SPEC	Spacecraft Pose Estimation Challenge
SV	service vehicle
TLE	two-line element
TV	target vehicle
VBS	vision-based sensor

cost associated with active sensors can hinder the massification of orbital servicers. Indeed, cameras are already rapidly becoming the choice for on-board sensors towards spacecraft rendezvous (RV) due to their small form factor and inexpensive power, mass, and volume costs. The past three years have witnessed a consolidation of AI-based (artificial intelligence) techniques for RV, particularly through deep learning (DL), using monocular cameras which do not make assumptions about the level of cooperation of the target [8]. However, this has not yet been established as a navigation approach for docking.

2. Related work

The use of vision-based sensors (VBSs) for RVD/B has traditionally consisted of establishing geometric relationships derived from the laws of imaging on the focal plane of a lens [7]: the SV illuminates retro-reflectors on the TV-side, which are configured according to a known pattern and are then imaged by the on-board camera. By knowing this configuration (i.e., the relative distances between the pattern markers), and the intrinsic parameters of a calibrated VBS (i.e., the field of view [FOV] and focal length), information on the SV-TV range, line of sight direction, and relative attitude can be computed by detecting said markers on each image. In the context of RVD/B, navigation requires the estimation of said quantities, which make up the 6-DOF relative pose $\boldsymbol{T}_{bt}$ mapping the target vehicle frame of reference $\underrightarrow{\mathcal{F}}_t$ to the service vehicle frame $\underrightarrow{\mathcal{F}}_b$ (Fig. 1). This entails the need for relative navigation sensors which, in the case of a VBS, define two extra frames: the physical camera frame $\underrightarrow{\mathcal{F}}_c$ (which can often be assumed coincident with $\underrightarrow{\mathcal{F}}_b$ without loss of generality) and the image plane frame $\underrightarrow{\mathcal{F}}_{\Pi}$ containing the image of the TV and where the image processing (IP) tasks occur.

In the computer vision literature, this problem is called the perspective-*n*-point (P*n*P [10,11]) problem. P*n*P can be used with only four markers to retrieve the full relative pose, although additional markers can be used to robustify the solution. Traditionally, cooperative spacecraft markers have been designed with concentric patterns of varying sizes to cover different operational sub-ranges and maintain acceptable resolution. While fiducial marker designs can be small, for long-range targets, multi-reflector spots are sometimes needed due to the low power density of the return signal [7]. Recently, there has been renewed interest in cooperative pose estimation literature, driven by the anticipation of autonomously approached spacecraft in future OOS missions. ArUco markers, popular in robot navigation and augmented reality, have been proposed as an alternative for docking operations [12]. These have the advantage a single marker being sufficient in providing the number of correspondences needed to retrieve the relative pose. Despite their simplicity, though, multiple ArUcos of different sizes are often used for robustness, as a reliable marker detection pipeline is crucial for accurate P*n*P-based pose estimation [13]. Furthermore, cooperative markers cannot be applied to existing satellites under missions which had not originally been designed with servicing in mind.

In contrast, modern AI techniques, namely through the advent of DL and deep neural networks (DNNs), have gained resurgence in the field of computer vision from the beginning of the previous decade onward, due to advances in commercial-off-the-shelf graphics processing units (GPUs) and accessibility to large-scale datasets such as ImageNet [14]. The eruption in popularity of DL arguably occurred in 2012 with AlexNet [15], a convolutional neural network (CNN) which won the ImageNet Large Scale Visual Recognition Challenge with a top-5 classification error more than almost 11 percent points lower than the runner-up; the novel use of GPU-based training massively accelerated the process, enabling deep learning to be competitive. CNNs, i.e., DNNs tailored to process image inputs through efficient convolution kernels, later became the norm, as new designs have competed in the challenge each year, resulting in exponential classification score improvements. Two notable examples are GoogLeNet [16], marked not only by a very deep architecture, but also by the implementation of parallel layers to extract multi-scale features; and ResNet [17], which introduced residual connections allowing the breakthrough to even deeper architectures. Most of these state-of-the-art CNNs have been open-sourced and model weights made available from pre-training on image classification tasks using the ImageNet dataset, which has since contributed to the swift evolution of DL in general and in the adoption of such models as CNN front-ends.

It took more than five years for the popularity of CNNs to migrate onto the domain of spacecraft relative pose estimation for RV. In 2019, ESA Kelvins' Spacecraft Pose Estimation Challenge (SPEC) benchmarked estimation errors obtained on assorted image inputs taken with on-board VBS from a simulated RV with the Tango spacecraft under random poses [18]. Although it did not tackle docking or berthing, it did demonstrate the good performance of AI-based approaches in vision-based navigation for space. The proposed techniques are model-based since they operate under the assumption that a priori knowledge of the target's aspect and structure is available in the form of training images. Since the target does not feature any physical fiducial or retro-reflective markers, only "natural" features derived from shape and texture, they become appropriate for uncooperative scenarios.

The relative pose can be estimated in an end-to-end fashion by following a direct approach (Fig. 2, top). This involves designing a DNN which generates a pose directly from image inputs, and has been tested for close-range rendezvous on SPEC [19]. It has the advantage of not

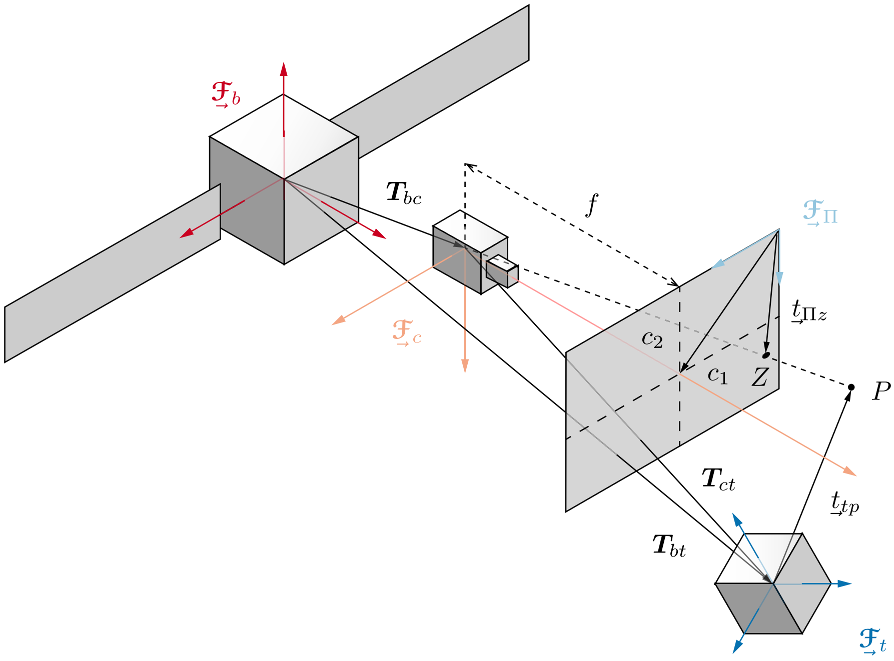

$\underset{\rightarrow}{\mathfrak{F}}_b$
$\boldsymbol{T}_{bc}$
f
$\underset{\rightarrow}{\mathfrak{F}}_\Pi$
$\underset{\rightarrow}{t}_{\Pi z}$
$\underset{\rightarrow}{\mathfrak{F}}_c$
c_2
c_1
Z
P
$\boldsymbol{T}_{ct}$
$\underset{\rightarrow}{t}_{tp}$
$\boldsymbol{T}_{bt}$
$\underset{\rightarrow}{\mathfrak{F}}_t$

Fig. 1. Frames of reference involved in the relative pose estimation problem for RVD/B [9].

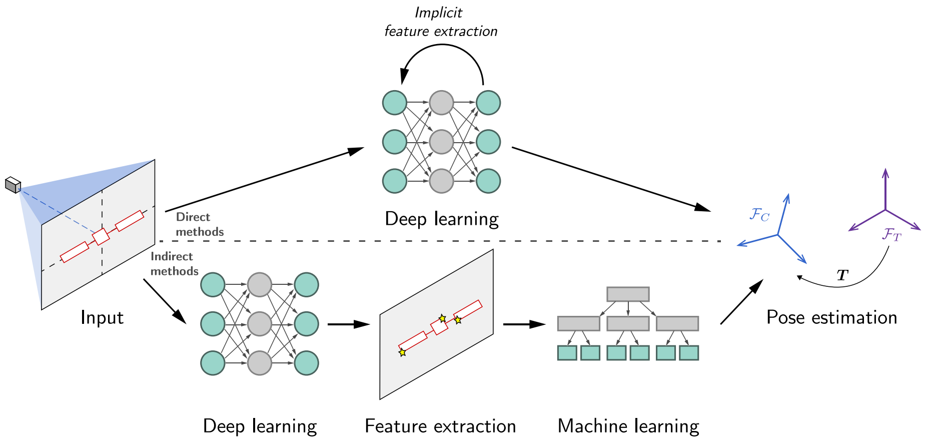
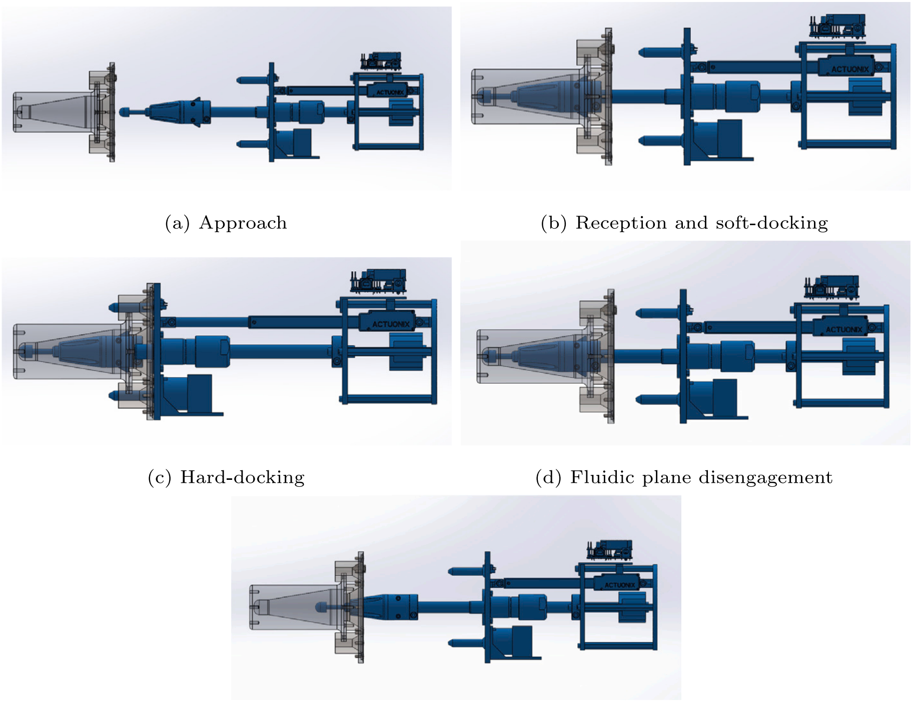


Fig. 2. Direct versus indirect methods for DL-based pose estimation [8].

relying on additional machine learning (ML) pipelines to generate the desired solution. Moreover, it entails a natural framework for the implementation of DL-based temporal modelling, which has been shown to improve the solution for time-series where there is a correlation between successive relative poses [20].

Interestingly, most of the highest scoring SPEC competitors [21] followed a so-called indirect approach towards DL-based pose estimation (Fig. 2, bottom): this entails relaying the use of a CNN entirely to the task of extracting features (usually 2D keypoints), and then using a P*n*P solver, or optimiser techniques such as Levenberg–Marquardt, to retrieve the solution from their correspondences to 3D control points. These are defined by the user at the pre-training stage and can be selected from the surface of a computer aided design (CAD) model of the target. As outlined by Kisantal et al. [18], the optimisation step applied to the 2D–3D correspondences output by these networks enables a refinement of the solution leading in turn to a more accurate pose estimate, explaining their increased performance on the SPEC dataset. As such, indirect methods have since become the preferred choice for DL-based pose estimation in RV. The potential of cameras as cost-effective navigation sensors for uncooperative RV has actually been demonstrated previously through this prism using traditional IP techniques under both model-based [22] and model-free [23] assumptions, as well as in combination with other economical hardware such as range finders [24]. However, DNNs in general benefit from representation learning, thus eliminating the need for manually selecting the most appropriate type of feature, the optimality of which would potentially be linked to each specific scenario or dataset.

To the best of our knowledge, neither technique has been evaluated on docking scenarios, thus making the Orbital AI-based Autonomous Refuelling (OIBAR) project the first to demonstrate the viability of deep neural networks using vision-only inputs for pose estimation of a docking structure. When designing the proposed navigation pipeline, a direct approach was chosen over an indirect one for two main reasons. Firstly, as the docking phase requires continuous estimation of relative states, there is motivation to explore the influence of temporal modelling using a direct framework. While a similar argument could be made for the rendezvous stage, the incentive for single-image pose estimation — often studied with indirect methods in the literature — is greater in this context than in the docking stage. Secondly, it can also be argued that this avenue can lead to a stronger decoupling from the camera intrinsics. Indirect methods may experience difficulties regressing on precise keypoint locations at larger distances if the camera



(a) Approach

(b) Reception and soft-docking

(c) Hard-docking

(d) Fluidic plane disengagement

(e) Separation

Fig. 3. Refuelling operations between TV and SV using the designed docking mechanism for OIBAR [25].

resolution is not sufficiently high, and at shorter distances a subset of these may fall outside the FOV. On the other hand, direct methods are end-to-end and can thus rely on different elements of the input image in such situations.

3. Methodology and design

This section details the approach followed for the execution of the OIBAR project. The design of the docking mechanism has originally been detailed by He et al. [25]; a rundown of the refuelling operations is reiterated below for context. Then, the AI-based navigator introduced in this paper is described.

3.1. Refuelling operations

The refuelling procedure in OIBAR can be broken down into the following sequence of operations, which are also illustrated in Fig. 3.

Initially, the AI-guided SV approaches the TV, adjusting velocity and aligning docking interfaces (Fig. 3(a)). The reception phase then begins with the end-effector probe entering the drogue, utilising a spring–damper to absorb shock and provide retraction force. Soft-docking is facilitated by two spring-loaded latches, preventing accidental detachment during probe entry into the drogue's varying diameter cavity (Fig. 3(b)). After soft-docking, the end-effector becomes restricted for hard-docking, which in turn ensures precise alignment for fluidic plane connection via two alignment pins engaged in the guide cavities on the berthing fixture side. (Fig. 3(c)). Fuel transfer is initiated, and after reaching the target pressure, valves are closed. Electrical connectors transmit control commands and pressure data. Post-refuelling, the end-effector is released, retracting the fluidic plane and automatically extracting the probe (Fig. 3(d)). The latch stepper motor resets the latch for the next refuelling process (Fig. 3(e)).

3.2. Software development

3.2.1. Architecture

Fig. 4 illustrates the base architecture of the AI-based navigation algorithm for OIBAR, termed OibarNet.

The proposed network is a direct (i.e., end-to-end) DNN taking red-green-blue (RGB) images of the TV's berthing fixture at time-step $\tau = k$ and outputting the corresponding 6-DOF pose relative to the SV. The front-end and backbone of OibarNet is the CNN that processes the image inputs (Fig. 4, in orange). Multiple CNN model candidates are considered and evaluated within the project (see Section 4); however, two architectural aspects are kept constant. The first is a dropout layer (Fig. 4, in green) added after the last convolutional layer to prevent overfitting [26]. The second is a global average pooling (GAP) layer (Fig. 4, in pink). GAP converts the CNN output to a fixed-dimension vector dependent only on the number of output channels, regardless of the image inputs spatial dimensions. This allows OibarNet to work with large input resolutions without needing to increase the network depth, and to potentially train it on different datasets without needing to alter the architecture. The CNN-processed features are then subject to a fully connected (FC) layer back-end which estimates the pose via regression.

As part of the project, the temporal modelling of the CNN features is also considered. This is investigated via the inclusion of a recurrent neural network (RNN) between the GAP output and the FC input.

3.2.2. Relative pose representation

The first FC block head maps the CNN output to a 3-vector estimate of the position, whereas the second head maps it to the 6-dimensional estimate of the attitude formulated by Zhou et al. [27]. This representation, denoted by $\boldsymbol{r}$, is obtained from the direction cosine matrix $\boldsymbol{R}$ through the mapping:

$$\begin{aligned} &\mathrm{SO}(3) \to \mathbb{R}^6 \\ &\boldsymbol{R} \mapsto \left[— \boldsymbol{R}_{:,1}^{\top} —, \quad — \boldsymbol{R}_{:,2}^{\top} — \right] = \boldsymbol{r}^{\top}, \end{aligned} \tag{1}$$

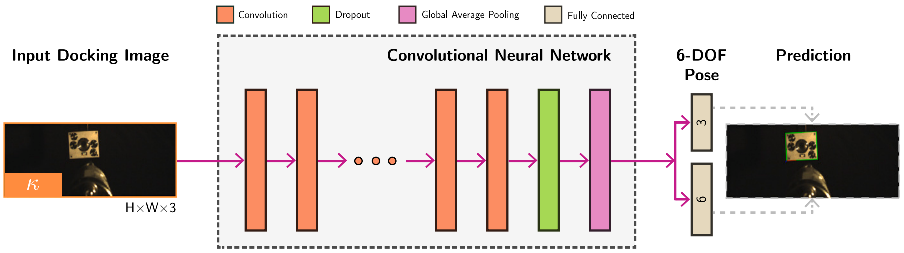


Fig. 4. OibarNet base DNN architecture for AI-based docking navigation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

i.e., discarding the third column of $\boldsymbol{R}$ and stacking the result. The transformation $\boldsymbol{r} \mapsto \boldsymbol{R}$ involves in turn reshaping $\boldsymbol{r}$ into a 3×2 matrix followed by a Gram–Schmidt orthogonalisation. The attitude representation warrants special attention, as the 4-dimensional quaternion, $\boldsymbol{q}$, is normally used to represent the attitude of a spacecraft due to its low dimensionality and lack of singularities. However, its antipodal ambiguity-induced discontinuities (i.e., $\boldsymbol{q} = -\boldsymbol{q}$) have been shown to yield sub-optimal results in a deep learning environment compared to the 6D representation — further details are given in Ref. [27]. For post-processing or error quantification, $\boldsymbol{q}$ can then be obtained from $\boldsymbol{R}$ using well-known isomorphisms [28].

3.2.3. Loss function

The combination of predicted position and attitude quantities in a single loss function requires the incorporation of a scaling factor since these two quantities normally deal in different magnitudes [19]. Typically, this scaling factor has been considered a hyperparameter part of the DNN's tuning process, which is sub-optimal.

In contrast, OibarNet follows the approach of Cipolla et al. [29] and attributes one weight each to the position and attitude, σ_t and σ_r, respectively, which become learnables and converge during the training process. The weights represent the task-specific variances of two Gaussian distributions, yielding a combined L^2 norm loss:

$$\mathcal{L} = \mathcal{L}_\mathrm{r} \exp\left(-2\hat{\sigma}_\mathrm{r}\right) + \mathcal{L}_\mathrm{t} \exp\left(-2\hat{\sigma}_\mathrm{t}\right) + 2\left(\hat{\sigma}_\mathrm{r} + \hat{\sigma}_\mathrm{t}\right), \tag{2}$$

where

$$\mathcal{L}_\mathrm{r} = \sum_{i=1}^{B} \|\hat{\boldsymbol{r}}^{(i)} - \boldsymbol{r}^{(i)}\|, \;\; \mathcal{L}_\mathrm{t} = \sum_{i=1}^{B} \|\hat{\boldsymbol{t}}^{(i)} - \boldsymbol{t}^{(i)}\|. \tag{3}$$

Here, $\hat{\boldsymbol{r}}, \hat{\boldsymbol{t}}$ are the 6D attitude and 3D position estimated by the network, respectively; $\boldsymbol{r}, \boldsymbol{t}$ are the corresponding ground truths; $\|\cdot\|$ denotes the L^2 norm; and B is the batch size.

4. Demonstration and testing

In this section, the methodology adopted for demonstrating the reliability of the developed AI-based solution for space docking and refuelling under OIBAR. The validation tests are divided into three fronts: hardware validation, software validation, and integration validation. The former has been reported by He et al. [25], the latter two are introduced herein.

4.1. Software validation

Supervised ML algorithms require labelled and well-structured datasets not only for evaluation, but also for training. This is especially true, and even more relevant, for DL-based methods, which require large and diverse batches of data to learn how to generalise towards unseen scenarios due to the very large number of parameters at play.

However, labelled datasets for spacecraft pose estimation are scarce and expensive to obtain due to the intricate environmental conditions that must be emulated. As such, the first step in the software validation campaign for OIBAR is to create a framework that allows for the generation of synthetic data: in contrast to real world sets, synthetic data is generally inexpensive and allows the possibility of having virtually unlimited samples.

Once this simulation environment is defined, the next step entails using it to produce synthetic docking trajectories. Lastly, an architecture selection round is performed based on the estimation performance on the data.

4.1.1. Simulation environment

The simulation environment for OIBAR is composed of two main components: a simulator designed in MATLAB/Simulink to replicate the orbital motion of the SV and TV under the influence of Earth; and an interface with the open-source 3D modelling software Blender.[4] for the purpose of generating synthetic but realistic imagery of the TV as viewed from the SV on-board VBS based on the states computed by the simulator.

The MATLAB/Simulink orbital simulator propagates 6-DOF pose of a body orbiting Earth based on an initial state at a given date, which is obtained from two-line element (TLE) sets. The effects of aerodynamic drag and the nonspherical mass distribution of Earth (J_2 zonal coefficient) are implemented as acting force perturbations. Aerodynamic and gravity gradient torques are implemented as acting torque perturbations. Using the planetary ephemerides blocks available in Simulink, the relative states of Earth and the Sun are also computed.

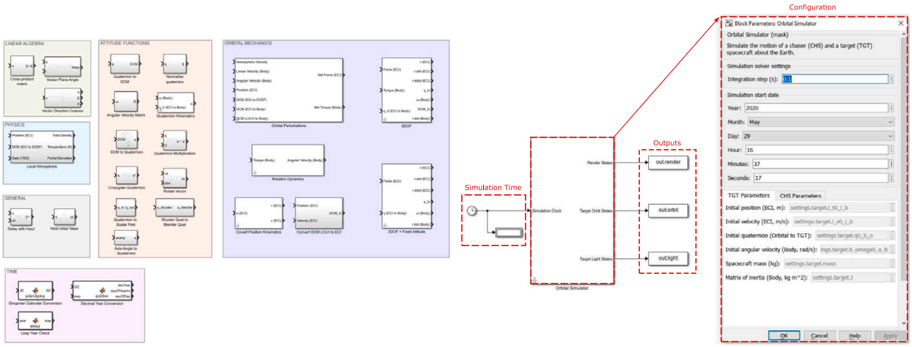
Fig. 5 illustrates the developed orbital trajectory simulator environment. The approach to building the simulator followed a modular design (Fig. 5(a)) which aimed to create basic blocks with fundamental functions, such as linear algebra, attitude manipulation and kinematics, and orbital mechanics, in order to facilitate any needed changes or customisation to the environment. The front-end (Fig. 5(b)) allows for a simple configuration of initial states, including the date and TV pose, permitting configurations for different scenarios.

The propagated states are then saved and interfaced with Blender, allowing the recreation of realistic images of the RVD/B dynamics of an Earth-orbiting SV and TV to be used for algorithm testing. Fig. 6 illustrates an example of a simulated scene within Blender. Besides the TV and SV, the Sun and Earth states have both been imported as well, ensuring a correct correspondence to the simulated time of day and solar phase angles.

4.1.2. Synthetic dataset generation

Using the simulator presented in Section 4.1.1, a synthetic VBS docking dataset was generated to validate the AI-based navigation system. The simulated scenario was chosen to be a refuelling of the ISS by the Kepler ATV. The CAD model of the OIBAR docking mechanism was imported into the Blender environment and attached to the vehicles: the end-effector replaced the ATV's existing RVD/B system, and the berthing fixture replaced the docking ports on the ISS. Due to a size mismatch, the original OIBAR CADs were scaled up to accommodate

[4] https://www.blender.org



Block Parameters: Orbital Simulator

Orbital Simulator (mask)

Simulate the motion of a chaser (CHS) and a target (TGT) spacecraft about the Earth.

Simulation solver settings

Integration step (s): 0.1

Simulation start date

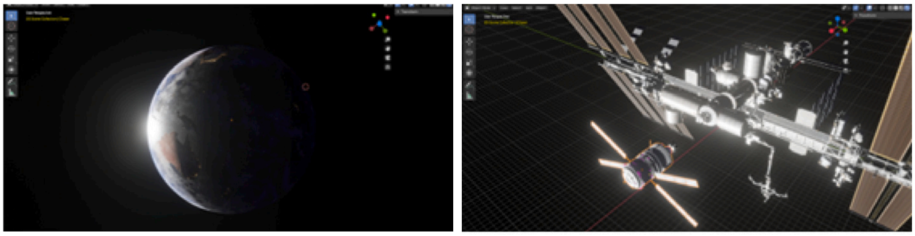
Year: 2020
Month: May
Day: 29
Hour: 16
Minutes: 37
Seconds: 17

TGT Parameters | CHS Parameters

Initial position (ECI, m): settings.target.i_t0_i_b
Initial velocity (ECI, m/s): settings.target.i_v0_i_b
Initial quaternion (Orbital to TGT): settings.target.q0_b_o
Initial angular velocity (Body, rad/s): ings.target.b_omega0_o_b
Spacecraft mass (kg): settings.target.mass
Matrix of inertia (Body, kg m^2): settings.target.J

(a) libOrbSim modular library (b) High-level view of orbital simulator

Fig. 5. MATLAB/Simulink orbital trajectory simulator environment..



(a) Zoomed out view (b) Detailed view

Fig. 6. Blender 3D rendering environment.

the vehicles within the simulation. In total, six different docking port locations on the ISS were considered; the inclusion of several docking ports was deemed beneficial as it grants diversity in terms of backgrounds, approach vectors (i.e., R-bar and V-bar) and illumination conditions.

The MATLAB/Simulink simulator is used to propagate the trajectory of the ISS from an initial TLE set, as well as the Earth and Sun states. The ATV SV guidance trajectories are generated directly relative to the TV frame; in particular, relative to the docking port considered for docking. Each SV trajectory begins at a relative shaft-plane-berthing-plane distance of 10 m and consists of three parts:

(1) **Acquisition.** This stage is characterised by a large cross-track motion (x–y plane) whereby the VBS is acquiring the target, prior to the end-effector and berthing fixture axes coinciding, but keeping the relative attitudes aligned. The SV translates between two randomly generated waypoints at radial distances between 1–2 m from the alignment axis, before reducing this distance to zero. The linear velocity in this stage varies from 0.09–0.12 m/s.
(2) **Forced translation.** Once the previous stage is concluded, the SV end-effector and RV berthing fixture are aligned in terms of a common along-track axis (z-axis), still at a relative distance of 10 m. The SV then translates along this axis at a nominal velocity of 0.03 m/s to close the distance until 3 m. To add variations amongst sequences, small perturbations are randomly generated and added to the translational and rotational motions; this is achieved by modelling a simple proportional integral (PI) controller and generating the next pose state from the feedback error. The magnitudes of the allowed perturbations vary from ±0.002 m/s for the along-track velocity, ±0.01 m for the cross-track position, and ±0.1 deg for the attitude; all with a probability of occurrence of 10 %.
(3) **Alignment and soft-docking.** This final stage begins with the position- and attitude-wise alignment of the fluidic and berthing planes from whatever misalignment state the previous stage may have ended in. The SV then translates towards the TV, while keeping an aligned attitude, until the spring protruding pin enters the central cavity, the latches are engaged, and soft-docking is achieved.

The average sequence duration is ∼ 5 min, and the synthetic dataset comprises 12 sequences in total. Furthermore, the synthetic dataset is composed of two subsets. The first one, synthetic/iss, consists of nominal scenario conditions where the docking mechanism is mounted on one of the docking ports of the ISS. The second subset, synthetic/perlin, models similar relative trajectories, but removes all meshes except for the docking mechanism, replacing the background with randomised Perlin noise. The objective of synthetic/perlin is to complement the synthetic/iss subset to provide additional training data and to help OibarNet focus on extracting features of the target and ignore the background and environment, such as the appearance of Earth behind the target. Table 1 summarises the characteristics of the generated dataset. The Sun elevation angle, i.e., the angle between the Sun direction vector and the orbital velocity vector, was selected to ensure daylight conditions for each specific docking port, whereby variance on the visual docking conditions was introduced by selecting values close to sunrise or sunset periods. Fig. 7 illustrates a few sample frames from the dataset.

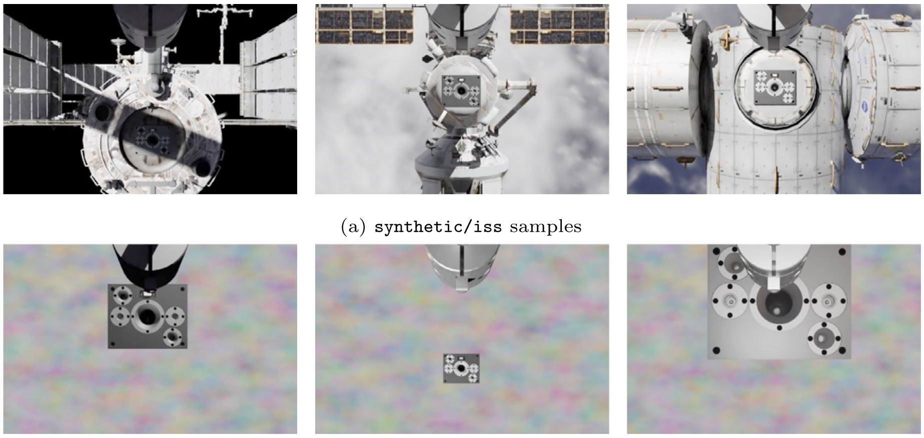
The synthetic dataset emulates the VBS used in the integration testing (see Section 4.4, Table 3); images are generated at a resolution of 744 × 480 px and a framerate of 10 Hz.

4.1.3. Training

From Table 1, sequences 1 and 8 were selected exclusively for testing, whereas the remaining sequences were used for training.

Further, the latter were divided according to a 80 %/20 % split to include validation data as well and allow the benchmarking of different models. Due to the data consisting of long temporal sequences, these were partitioned into smaller ones to guarantee the proportions of the intended split in an unbiased way. To achieve this, each sub-sequence i's length in seconds defined by randomly sampling a power of two, n_i, according to the following formula:

$$\mathrm{len}(i) = 2^{n_i}. \quad (4)$$



(a) synthetic/iss samples

(b) synthetic/perlin samples

Fig. 7. Sample frames from OIBAR's synthetic dataset.

Table 1
Synthetic dataset characteristics used for training, validation, and testing of OibarNet.

Sequence	Docking port	Approach axis		Sun elevation angle (deg)	Background		Duration (s)
		V-bar	R-bar		iss	perlin	
1	1	+		37	×		332
2	1	+		75		×	319
3	2		−	56	×		358
4	2		−	146		×	336
5	3	−		127		×	348
6	3	−		165	×		327
7	4		−	56		×	333
8	4		−	146	×		329
9	5		+	56	×		333
10	5		+	146		×	342
11	6		+	56		×	323
12	6		+	146	×		355

The values for n_i were sampled from the set of integers $\{6,\dots,10\}$, resulting in sub-sequence lengths belonging to the set of $\{64,\dots,1024\}$ seconds.

Image augmentation was performed online on the training data to prevent overfitting (i.e., to boost generalisation performance during inference with new, unseen data). This was applied in two fronts. The first one is related to transform operations in the IP domain: randomised changes in terms of brightness, contrast, colour, Gaussian noise and blur, for example, are generated to robustify the network against potentially unpredicted imaging conditions during deployment. The second one is related to operations in the pose domain, whereby randomly generated perspective transforms are applied to images in the sequence to simulate deviations in the trajectory (i.e., translation shifts, in-plane rotations, homography-induced off-plane rotations). The latter is of particular importance since, despite the sequence partitioning for training, the forced translation phase dominates each sequence, generating an imbalance on the distribution of position states. The reader is directed towards Rondao [9] for illustrations of the implemented augmentation techniques. Table 2 compiles the quantitative parameters of the randomised transformations used during training.

OibarNet is implemented in MATLAB R2021b using a custom-developed library. Models are trained for 100 epochs with a cyclical learning rate decay of 5 cycles [30]. The Adam optimiser [31] is used. A dropout probability of 0.2 is used. Training is performed on City, University of London's high performance computing facility Hyperion using one NVIDIA® Quadro RTX™ 8000 GPU with 48 GB VRAM.

4.1.4. Testing

The test results are presented in terms of the position and attitude error metrics, respectively:

$$\delta\tilde{t} := \|\hat{t} - t\|, \tag{5}$$

$$\delta\tilde{q} := 2\arccos\left(\hat{q}^{-1} \otimes q\right)_4, \tag{6}$$

where the subscript "4" denotes the scalar element of the resulting quaternion. Additionally, the position error is also assessed in terms of the relative range:

$$\delta\tilde{t}_r := \frac{\delta\tilde{t}}{\|t\|}. \tag{7}$$

It is noted that the pose estimation accuracy requirements for OIBAR have been defined by He et al. [25] as 5 % of range for the position and 5 deg for attitude. This is driven by the fact that the functional testing setup is configured to use a robotic arm, i.e., more closely simulating a berthing operation. In such cases, pose estimation accuracy becomes less critical, and the "1 %–1 deg" rule of thumb normally applied for the final approach can be relaxed by a factor of 5 [7].

4.1.5. CNN architecture selection

From the relative pose estimation point of view, a few differences may be expected between an RV manoeuvre and a docking sequence. Firstly, a reduced variation in the attitude is expected during docking since the SV is expected to be inside the cone-shaped approach corridor of the TV [7]; in opposition, the target may be tumbling during RV. Secondly, an increased apparent variation in the position can be expected during docking, as due to the reduced relative distance any small shift will result in a large displacement of the TV berthing fixture in the FOV.

To better assess the influence of these factors, multiple CNN architectures are benchmarked. The baseline is Darknet-19 [32], which has successfully been applied in the past to the problem of pose estimation in RV [20]. To analyse the effect of increasing the capacity of the model, Darknet-53 [33] and ResNet-101 [17] are included. Lastly, it

Table 2
Image augmentation randomisation parameters used in the training process of OibarNet..

Transform	Parameter		Unit	Description
	Minimum	Maximum		
Channel shift	−20	20	–	Pixel intensity shift value
Gaussian blur	7	13	px	Kernel size
Gaussian noise	3×10^{-3}	1×10^{-2}	–	Variance
JPEG compression	2	8	–	Intensity
Median blur	7	13	px	Kernel size
Patch dropout	10	10	%	Mean image area covered
	3	5	%	Patch size relative to smallest image dimension
Brightness	−0.2	0.2	–	Intensity
Contrast	0.8	1.2	–	Intensity
CLAHE	2	6	–	Pixel intensity clip range
	8	8	–	Number of tiles
Gamma	0.35	1.50	–	Factor
Camera rotation (3 DOF)	−5	5	deg	Magnitude per axis
Image rotation (in-plane)	−5	5	deg	Magnitude
Image translation	−150	150	px	Magnitude

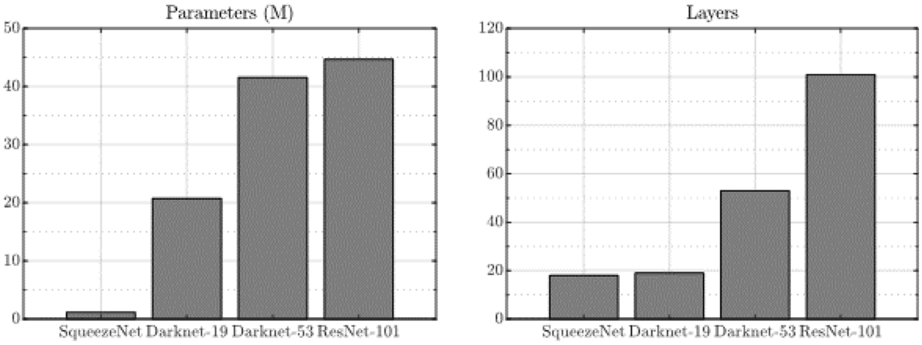


Fig. 8. Characteristics of the four different benchmarked CNN models.

is also important to verify the change in performance when reducing the capacity, and SqueezeNet [34] is thus included in the benchmark as well.

Fig. 8 summarises the number of parameters, in millions, and number of layers of the four different CNN models considered for benchmarking.

4.2. Integration validation methodology

The goal of the integration testing is to validate the combination of the hardware and software blocks outlined above. In this setup, the navigation VBS is incorporated into the hardware setup [25] to acquire a stream of images to be processed by the navigation algorithm (Section 4.1) during the docking manoeuvre emulated by the robotic setup.

Fig. 9 illustrates the integration validation setup. A blackout backdrop is placed behind the target berthing fixture to simulate the imaging conditions of a featureless deep space background. The target itself is illuminated by a single 400 W halogen directional floodlight. The distance at which this illumination source was placed from the target was adjusted to simulate an irradiance of approximately 1361 W/m^2, typical for low Earth orbit, under the following rationale. The light is modelled as a point source illuminating a cone with an apex angle θ and generatrix r. These represent, respectively, the beam spread and the distance between the apex (or origin) and the target being illuminated. Let $\theta = 60$ deg, which is a typical beam spread for wide flood halogen lamps, such as the one used herein. Given θ, the corresponding solid angle Ω can be calculated as:

$$\Omega = 2\pi \left(1 - \cos\left(\frac{\theta}{2}\right)\right) = 0.8418\,\text{sr}. \tag{8}$$

The solid angle Ω, the irradiance E, the power P, and the distance r are related by the formula:

$$E = \frac{P}{\Omega \cdot r^2}. \tag{9}$$

Table 3
Technical data — DFK 22BUC03.

Parameter	Units	Value
Resolution	px	744 × 480
Maximum frame rate	Hz	76
Focal length	mm	3.5
Horizontal FOV	deg	65.6
Vertical FOV	deg	44.7

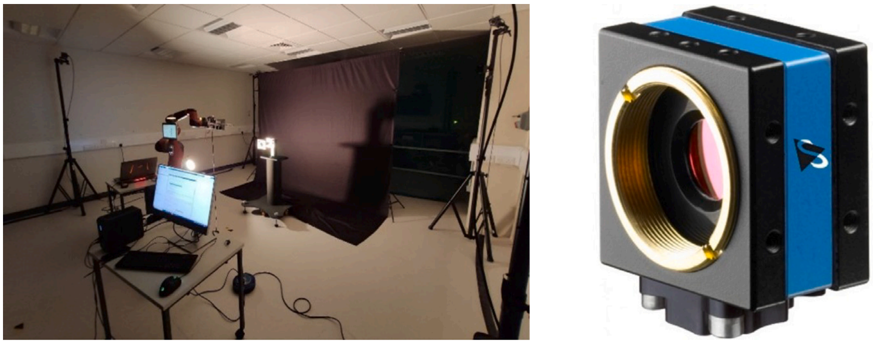
Solving the above for r and plugging in the remaining values yields a distance of approximately 0.6 m at which the lamp should be positioned relative to the berthing fixture in the experiment. This distance also allows for the floodlight to be conveniently positioned outside of the imaging sensor's FOV mounted on the robotic arm manipulator.

The SV and TV are placed inside the capture volume of an OptiTrack.[5] motion capture system for recording the ground truth measuring approximately 5 × 5 × 3 m. OptiTrack can record 6-DOF pose data of rigid and flexible bodies by detecting, tracking, and triangulating passive near infrared markers placed on targets. The data can be saved or stream over a local network in real-time.

The OptiTrack setup at City consists in six PrimeX 13 cameras with a resolution of 1280 × 1024 px running at a native framerate of 240 Hz, capable of achieving positional errors less than ±0.20 mm and rotational errors less than 0.5 deg.

The used VBS is the Imaging Source DFK 22BUC03 colour camera with a $\frac{1}{3}$ inch format CMOS sensor (Onsemi MT9V024) and a native resolution of 744 × 480 px, fitted with a Kowa LM4NCL 3.5 mm focal length lens. Table 3 summarises the technical characteristics of the VBS.

[5] https://optitrack.com



(a) Laboratory setup

(b) DFK 22BUC03 VBS

Fig. 9. Integration validation setup at City, University of London’s ASMIL.

The workstation consists of an Intel® NUC 9 Pro with an NVIDIA® RTX™ 3060 Ti Mini GPU with 8 GB VRAM. The workstation is used for both experimental data offline validation of OibarNet and real-time online testing of the network, at a framerate of 10 Hz.

4.2.1. Experimental dataset generation

The docking imaging sequences acquired with the experimental setup follow the same structure as the synthetic dataset (Section 4.1.2) albeit with two key differences. The first one is that all experimental sequences feature the same type of background (black, deep space). The second is that, rather than implementing PIinduced pose perturbations during the forced translation (Phase 2), a static misalignment of the pose is randomly introduced in each sequence at the beginning of the phase, which is then corrected at the beginning of the final one.

In total, 12 experimental trajectories are collected, whereby the angle of illumination alternates between port and starboard. The average sequence duration is ∼3.15 min. The first 10 sequences are used for training and validation of the model according to the methodology of Section 4.1.3. Sequences `experimental/11` and `experimental/12` are used exclusively for testing.

4.2.2. Ground truth calibration toolbox

The OptiTrack system used to record the ground truth measures the poses of rigid bodies equipped with infrared markers. However, it does not directly output the relative pose between the VBS and the TV (as illustrated in Fig. 1, Section 2), which is required by the navigation algorithm.

To this end, a toolbox was developed in MATLAB to calibrate the output OptiTrack data and generate the required relative pose, based on the work of Valmorbida et al. [35] and Pasqualetto Cassinis et al. [36]. The output of the calibration toolbox are the static transforms $\boldsymbol{T}_{ic}$, mapping the camera frame $\underrightarrow{\mathcal{F}}_c$ to the frame of reference $\underrightarrow{\mathcal{F}}_i$ defined by the physical markers placed on its housing and tracked by OptiTrack, and $\boldsymbol{T}_{sb}$, mapping the target’s body frame $\underrightarrow{\mathcal{F}}_b$ to the frame of reference $\underrightarrow{\mathcal{F}}_s$ defined by the markers placed on it. These transforms then make possible to map the OptiTrack marker-defined rigid bodies’ poses, which are measured relative to $\underrightarrow{\mathcal{F}}_o$, the system’s arbitrary global frame of reference, into a useable ground truth $\boldsymbol{T}_{bc}$ defined in terms of the VBS frame of reference. The overall uncertainty of the framework can be quantified in terms of the total reprojection error between model and image corners and is estimated to be below 20 px for ranges above 2 m [36]. However, this upper bound is expected to be lower for the present application due to the shorter relative ranges considered. Fig. 10 illustrates the output of the calibration procedure on select samples of the dataset.

5. Results

5.1. Software testing

5.1.1. CNN model benchmarking

Fig. 11 illustrates the results of the different models trained on the `synthetic` dataset, presented in terms of mean position and attitude errors averaged per trajectory. It can be seen that the performance of SqueezeNet is considerably worse than the baseline Darknet-19, yielding errors twice as large for both position and attitude. Both networks have a very similar number of layers, but Darknet-19 has substantially more learnable parameters (as indicated in Fig. 8); the reduced attitude variance in the docking manoeuvres is thus shown not to justify a decrease in parameters.

Interestingly, the error for Darknet-53 actually increases with respect to the baseline. Once the capacity of the CNN is further increased with ResNet-101, though, the error decreases again, making the network the best performing model (except on position validation error, which is slightly larger than Darknet-19’s).

The results of Fig. 11 are presented with the caveat that they represent average errors per trajectory, but where the data is not composed of random images but time sequences. As such, while a histogram visualisation is useful for a first analysis of each model’s performance, it is also important to look at how these perform in specific, individual situations. For example, Fig. 12 represents the qualitative performance of each model on a single frame of one of the synthetic validation sequences; the rectangular boundary of the berthing fixture is reprojected in green using the predicted pose, and the axes of the estimated frame $\underrightarrow{\mathcal{F}}_t$ are also shown. The results show that SqueezeNet is overfitting at least on the position state, as it expects the berthing fixture to be located in the centre of the FOV, when in reality the SV end-effector is still misaligned. The other three models with increased capacity demonstrate no issues in estimating the correct relative position.

Consider now, however, the performance on one training sequence, as illustrated in Fig. 13: SqueezeNet (a) is shown to be underfitting, but so is Darknet-19 (b). This suggests that increasing the capacity would benefit OibarNet, as confirmed by the frame output by Darknet-53 (c) showing a better fit, despite the summary metrics in Fig. 12. The performance with ResNet-101 (d) is slightly better even, confirming it as the choice for the final CNN model in OibarNet.

5.1.2. Performance evaluation on synthetic dataset

In this subsection, the performance of the navigation algorithm is evaluated on the test sequences `synthetic/01` and `synthetic/08`, as outlined in Section 4, and according to the selected ResNet-101 CNN architecture for OibarNet.

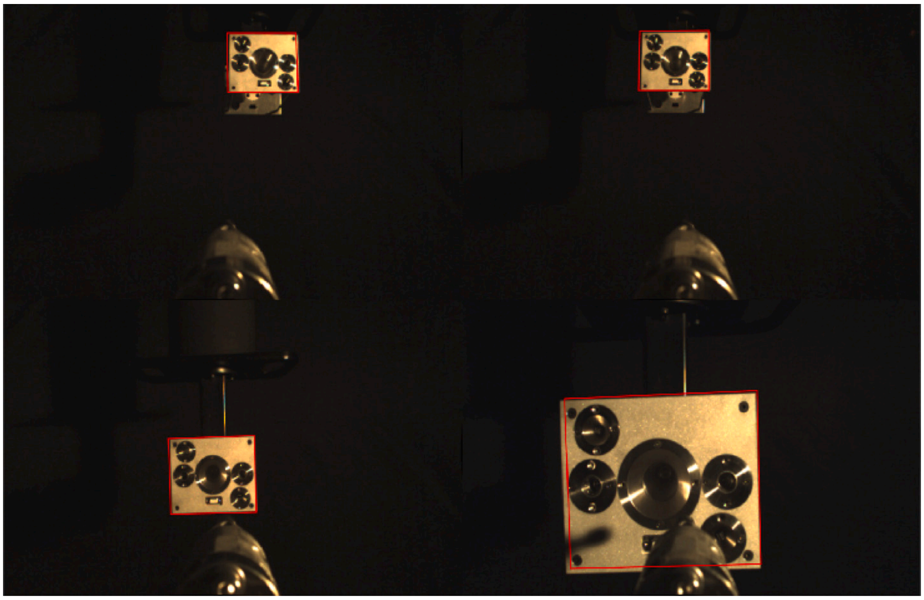


Fig. 10. Ground truth calibration toolbox output, visualised on some frames of the experimental dataset by reprojecting the target's CAD model (in red) according to the measured pose. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

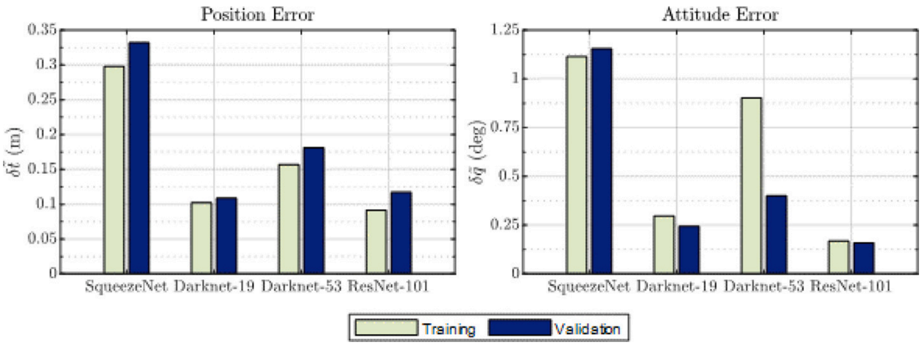


Fig. 11. Average training and validation pose estimation errors for different CNN architectures trained on the `synthetic` dataset.

Table 4
Summary performance statistics on the two test sequences of the synthetic dataset. Position errors are range-normalised. "Std." denotes standard deviation.

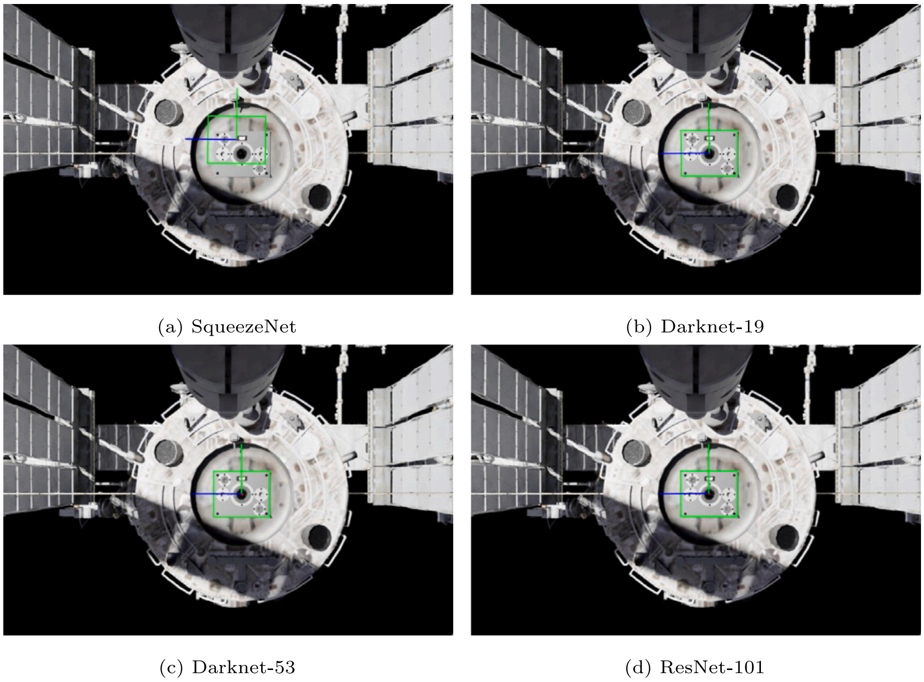
Sequence	Position error (%)			Attitude error (deg)			Requirement compliance (%)	
	Mean	Median	Std.	Mean	Median	Std.	Position	Attitude
`synthetic/01`	1.81	1.39	1.53	0.29	0.26	0.20	96.33	100
`synthetic/08`	0.56	0.43	0.50	0.28	0.26	0.17	99.91	100

Fig. 14 showcases the attained pose estimation errors for each sequence using the final OibarNet model; the position errors are normalised as a percentage of range. Table 4 summarises these statistics.

The figures demonstrate that, for `synthetic/01`, OibarNet fulfils the 5 % maximum range-normalised position error requirement (defined in Ref. [25]) for most of the trajectory. The exception is a segment corresponding to phase 1 (acquisition) whereby the SV moves to a waypoint representing a large displacement relative to the alignment axis, representing about 3.7 % of the sequence's duration. After this period, the error converges to values below 2.5 % of range, further decreasing as the SV closes in on the TV, until the beginning of phase 3 (alignment and soft-docking), where the very short range causes the error to rise, but not above the requirement threshold. The attitude estimation performance is shown to fully comply with the 5 deg maximum error requirement.

The position estimation performance of the navigation algorithm on `synthetic/08` is observed to be better than the previous sequence, as an improvement of 1.25 percent points on the mean value and 0.96 percent points on the median value are achieved. Furthermore, the position estimate is virtually fully compliant with the defined requirement, save for a singular spike (less than 0.1 % of the trajectory). The attitude estimation is again entirely compliant and practically does not surpass 1 deg in error.

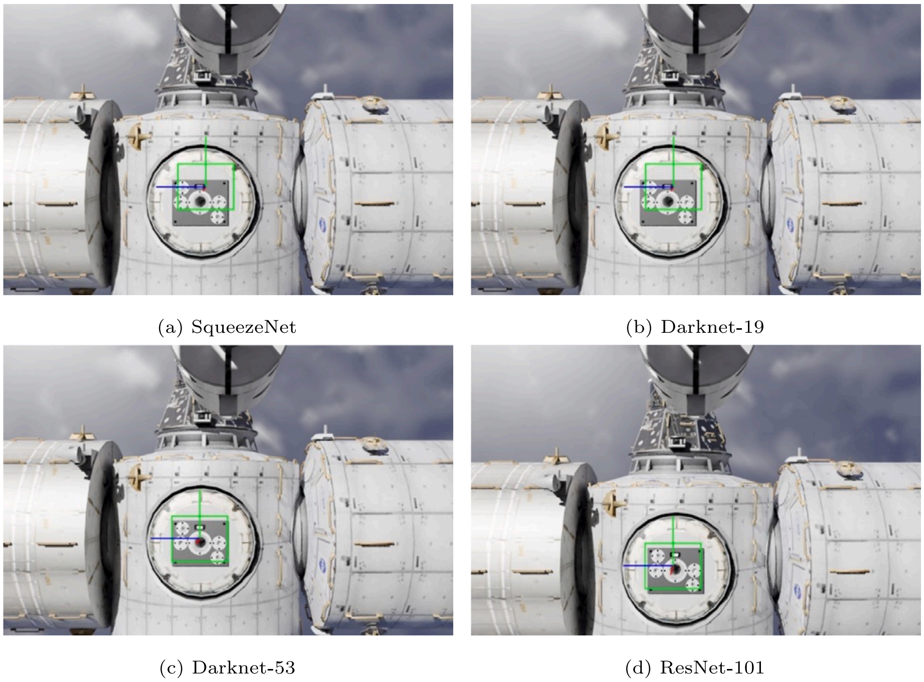
Figs. 15 and 16 exhibit some frames from each sequence with the respective qualitative pose estimation fit overlaid. On `synthetic/01` the TV structure surrounding the berthing fixture is quite complex, which from the IP point of view represents a more challenging background than the case of `synthetic/08`, despite it being an R-bar trajectory which includes Earth. Additionally, the illumination conditions on the former appear to make the berthing fixture harder to distinguish from the ISS structure relative to the latter. Both aspects



(a) SqueezeNet (b) Darknet-19

(c) Darknet-53 (d) ResNet-101

Fig. 12. Qualitative pose estimation performance on a validation sequence of the `synthetic` dataset for different CNN models.



(a) SqueezeNet (b) Darknet-19

(c) Darknet-53 (d) ResNet-101

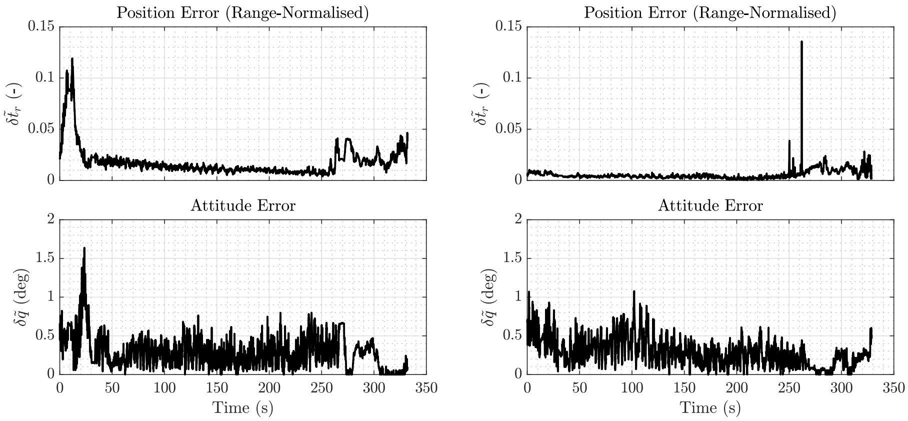
Fig. 13. Qualitative pose estimation performance on a training sequence of the `synthetic` dataset for different CNN models.

could provide an explanation to the increased position error seen in the beginning of `synthetic/01`.

5.1.3. Effect of temporal modelling

The designed OibarNet pipeline uses a CNN front-end to process incoming images and extract features. However, these images are processed individually, whereby the data as a whole represents a time sequence depicting a docking manoeuvre, which implies that each sample is time correlated. Modifications to CNN architectures have been proposed in the past to account for this correlation and shown to improve the relative pose estimation error for rendezvous. Specifically, deep recurrent convolutional neural networks (DRCNNs) include a recurrent sub-network as the back-end of the pipeline that models the features extracted by the CNN [20].

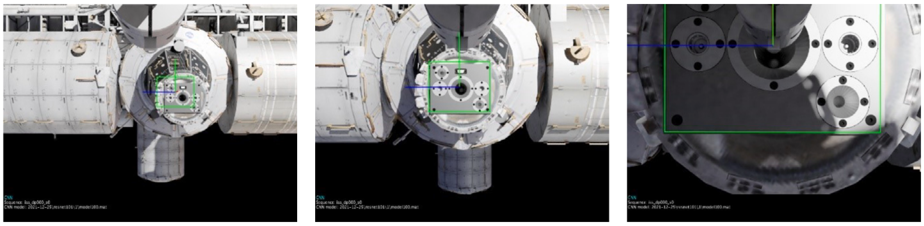
This test investigates the effect of applying a DRCNN to the problem of relative pose estimation for docking. To this end, the trained CNN model was appended with a recurrent model, further trained on the output of the CNN for the same dataset, consisting of bi-directional



(a) `synthetic/01`

(b) `synthetic/08`

Fig. 14. Estimated position and attitude errors over time on the two test sequences of the synthetic dataset.

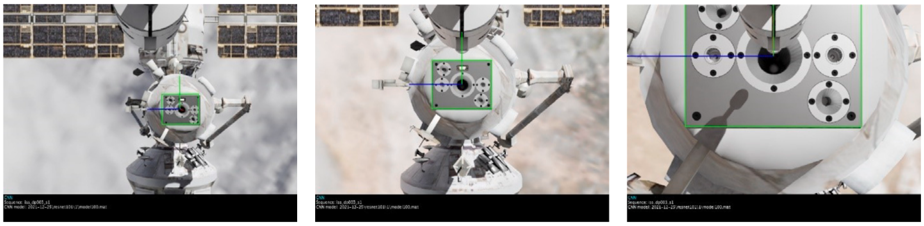


(a) $\tau = 20\,\mathrm{s}$

(b) $\tau = 160\,\mathrm{s}$

(c) $\tau = 305\,\mathrm{s}$

Fig. 15. Qualitative pose estimation performance on the `synthetic/01` test sequence.



(a) $\tau = 20\,\mathrm{s}$

(b) $\tau = 155\,\mathrm{s}$

(c) $\tau = 289\,\mathrm{s}$

Fig. 16. Qualitative pose estimation performance on the `synthetic/08` test sequence.

long-short-term memory (BiLSTM) cells [37]. Contrary to regular long-short-term memories (LSTMs), BiLSTMs run sequence inputs in two directions: one from past to future, and the other from future to past, thus preserving information from both past and future. This feature can be beneficial for RVD/B pose estimation problems since trajectories are continuous, meaning that not only do the previous states influence the present, but states in the future provide context to the preceding ones.

The results of the benchmark are illustrated in Fig. 17. It can be seen that the addition of a recurrent layer degrades not only the validation performance, but also the training performance; this is witnessed both in terms of position and attitude estimation. Adding more recurrent layers lowers the error on the attitude estimate, but even with three layers this is still higher than that obtained for the CNN alone. Furthermore, the position error is shown not to decrease.

This study represents an interesting result since it is seemingly counter-intuitive and diverges from the findings reported by Rondao et al. [20]. However, whereas the apparent relative motion during an RV is typically smooth and predictable (e.g., SV at a hold point observing the TV tumbling), the docking trajectories modelled within the scope of OIBAR are actually more dynamic and include higher stochasticity due to the random perturbations added during the approach phase. As such, one explanation towards the poor performance of the DRCNN in this case could be the failure in modelling these high-frequency, random changes in motion, in which the CNN indeed an advantage as it is processing each time-step individually.

Further avenues of research could still be pursued, however. For example, the inclusion of attention-based mechanisms remains to be investigated for RVD/B, where the network would be capable of self-learning weights to be attributed to each time-step in the sequence, thus becoming able to let certain segments influence the estimate more than others (e.g., placing less attention on the immediate perturbations and more on the overall along-track motion).

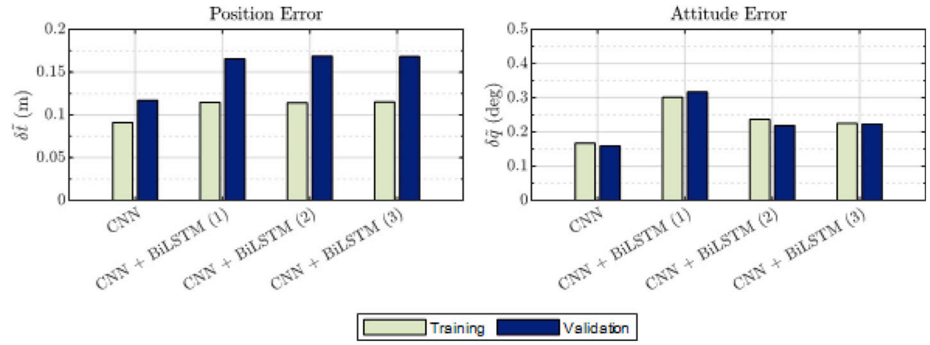
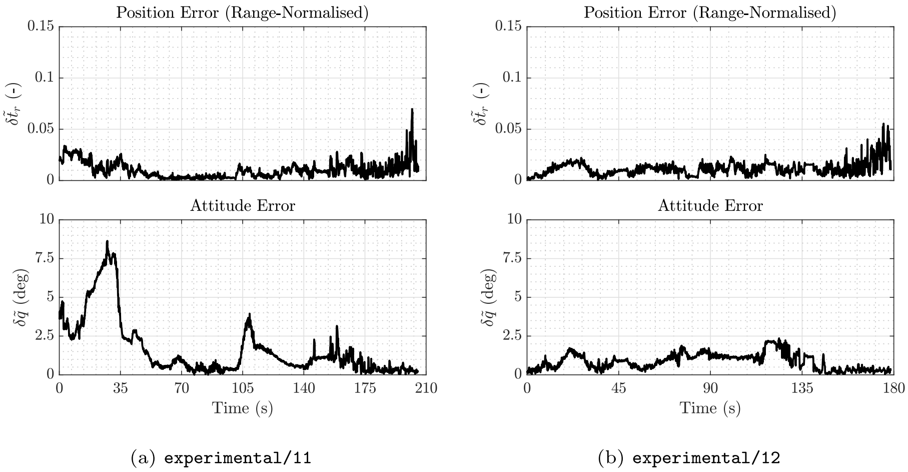


Fig. 17. Effect of adding recurrent neural layers on the training and validation performance of the synthetic dataset. The numbers in parenthesis denote the number of recurrent layers.



(a) experimental/11

(b) experimental/12

Fig. 18. Estimated position and attitude errors over time on the two test sequences of the experimental dataset.

Due to the attained results, the OibarNet architecture was not altered for the integration tests.

5.2. Integration testing

5.2.1. Performance evaluation on experimental dataset

This section is analogous to Section 5.1.2 with the difference that the selected OibarNet CNN architecture is evaluated and tested on experimental data collected in laboratory. As outlined in Section 4, the performance of the navigation algorithm is evaluated on the test sequences `experimental/11` and `experimental/12`.

Fig. 18 displays the attained pose estimation errors for both trajectories. Table 5 summarises the performance metric statistics. Lastly, Figs. 19 and 20 illustrate qualitative estimation results for a few frames of the `experimental/11` and `experimental/12` sequences, respectively. The relative position estimation error follows a similar trend to the synthetic dataset case: lower during the approach phase and increasing in the final alignment and soft-docking phase. Overall, the curves oscillate more in amplitude for both trajectories; this is a possible by-product of using real-data which can be contaminated with random errors (e.g., sensor noise) and systematic errors (e.g., errors in the motion capture system calibration), which are not seen in the ideal development conditions of synthetic datasets. The reduced number of training samples relative to the synthetic case also affects the solution (i.e., the experimental trajectories are shorter). Nevertheless, the requirement compliance is virtually 100 % for both trajectories.

The experimental evaluation demonstrates, on average, a higher attitude error than the synthetic evaluation case. In particular, for `experimental/11`, a spike in the initial 35 seconds of the sequence cause the error to surpass 7.5 deg which brings down the requirement compliance to 91.2 %. This is due to the SV travelling to a waypoint during the acquisition phase that is quite distinctive from the others present in the training data, making the berthing fixture appear in the top right corner of the FOV close to the image edge (Figure 27 a). However, the proposed training scheme which includes image augmentation prevents the error from diverging, and the estimate begins to recover after $\tau = 35$ s, reaching minimum values during and immediately before the final phase. In `experimental/12`, the attitude estimation error is bounded at 2.5 deg.

6. Conclusions

OOS is now becoming increasingly important and represents a significant cost saving measure, opening up a new global market. Whereas the latest OOS initiatives and demonstrators have focused on clear near-term commercial opportunities, such as life extension and end-of-life, longer-term OOS segments expected to emerge this decade such as refuelling are set to unravel novel and wider business opportunities, and have the potential to unlock new orbital ecosystems.

On this basis, City, University of London has developed OIBAR, a novel AI-based solution for space docking and refuelling applications consisting of the combination of two major components: a vision-based orbital relative navigation algorithm to safely approach and dock to the target vehicle; and an intelligent hardware mechanism achieving the mechanical docking and refuelling operation of the target. The present document reported the development and achievements of the OIBAR project, namely the design procedure of its key features adopted to tackle the problem, the modelling of the mechanism and software architecture, and the validation of the combined solution. Functional testing of the prototype was performed in laboratory using a 7-DOF

Table 5
Summary performance statistics on the two test sequences of the experimental dataset. Position errors are range-normalised. "Std." denotes standard deviation.

Sequence	Position error (%)			Attitude error (deg)			Requirement compliance (%)	
	Mean	Median	Std.	Mean	Median	Std.	Position	Attitude
`experimental/11`	1.02	0.84	0.80	1.65	0.91	1.85	99.71	91.20
`experimental/12`	1.17	1.08	0.67	0.86	0.87	0.52	99.72	100.00

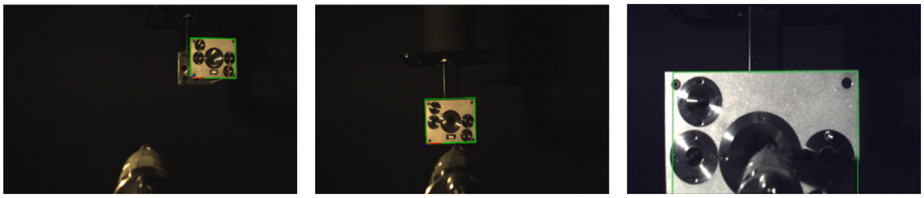

(a) $\tau = 20\,\mathrm{s}$ (b) $\tau = 160\,\mathrm{s}$ (c) $\tau = 305\,\mathrm{s}$

Fig. 19. Qualitative pose estimation performance on the `experimental/11` test sequence.

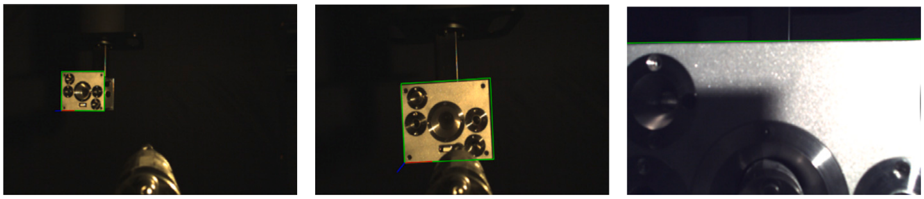

(a) $\tau = 20\,\mathrm{s}$ (b) $\tau = 160\,\mathrm{s}$ (c) $\tau = 305\,\mathrm{s}$

Fig. 20. Qualitative pose estimation performance on the `experimental/12` test sequence.

robotic manipulator to simulate docking/berthing trajectories and a state-of-the-art Optitrack ground truth measurement system to assess the quality of the navigation solution.

A CNN-based direct VBS navigation algorithm was proposed to estimate the relative states between SV and TV to achieve docking. A MATLAB/Simulink simulator was developed to generate synthetic data intended to train and evaluate the solution. A benchmarking campaign was performed to assess the best architecture candidate. The final model reported average errors per trajectory of 1.19 % and 0.29 deg for range-normalised position and for attitude, respectively, with accompanying average standard deviations of 1.14 % and 0.19 deg. The performance requirements were satisfied for nearly the whole length of the test sequences. The inclusion of BiLSTM-based recurrent layers was analysed but found not to improve the base CNN model.

Lastly, the combined solution was assessed through an integration testing campaign. The navigator was trained and tested on experimental data collected in laboratory using the mechanical docking prototype. Similarly to the synthetic dataset results, these have achieved near-complete compliance with the proposed accuracy requirements, thus validating the findings. Some differences between the two sets have been observed, however, namely in terms of an overall increase in the mean attitude error which can be attributed to an increased variation in the possible attitude states induced by the waypoint programming on the robotic manipulator. An enlargement of the training dataset poses is expected to further reduce the error.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Duarte Rondao reports financial support was provided by UK Research and Innovation. If there are other authors they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research has been supported by the UK Robotics and Artificial Intelligence (RAI) hub in Future AI and Robotics for Space (FAIR-SPACE: EP/R026092).

References

[1] Vicki Cox, Northrop Grumman's wholly owned subsidiary, SpaceLogistics, selected by DARPA as commercial partner for robotic servicing mission, 2020, URL https://news.northropgrumman.com/news/releases/northrop-grummans-wholly-owned-subsidiary-spacelogistics-selected-by-darpa-as-commercial-partner-for-robotic-servicing-mission.

[2] Patrice Benarroche, Martial Vanhove, Mauro Augelli, ATV operations: From demo flight to human spaceflight partner, in: SpaceOps 2014 Conference, American Institute of Aeronautics and Astronautics, 2014, http://dx.doi.org/10.2514/6.2014-1665.

[3] Laurie Metcalfe, Tara Hillebrandt, Robotic refuelling mission: Demonstrating satellite refuelling technology on board the ISS, in: Proceedings of 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space, I-SAIRAS, European Space Agency, Montréal, Canada, 2014.

[4] European Space Agency, ESA invites ideas to open up in-orbit servicing market, 2021, URL https://www.esa.int/Safety_Security/Clean_Space/ESA_invites_ideas_to_open_up_in-orbit_servicing_market.

[5] Astroscale, Fair-Space, Catapult Space Applications, UK In-Orbit Servicing Capability, Technical Report, UK Space Agency, Swindon, UK, 2021, URL https://sa.catapult.org.uk/wp-content/uploads/2021/05/Catapult-Astroscale-Fairspace-Platform-for-Growth-report-final-27-05-21.pdf.

[6] Bong Wie, Vaios Lappas, Jesús Gil-Fernández, Attitude and orbit control systems, in: Malcolm Macdonald, Viorel Badescu (Eds.), The International Handbook of Space Technology, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-41101-4, 2014, pp. 323–369, http://dx.doi.org/10.1007/978-3-642-41101-4_12.

[7] Wigbert Fehse, Sensors for rendezvous navigation, in: Automated Rendezvous and Docking of Spacecraft, in: Cambridge Aerospace Series, Cambridge University Press, 2003, pp. 218–282, http://dx.doi.org/10.1017/CBO9780511543388.008.

[8] Jianing Song, Duarte Rondao, Nabil Aouf, Deep learning-based spacecraft relative navigation methods: A survey, Acta Astronaut. 191 (2022) 22–40, http://dx.doi.org/10.1016/j.actaastro.2021.10.025.

[9] Duarte Rondao, Multimodal Navigation for Accurate Rendezvous Missions (Ph.D. thesis), Cranfield University, 2021, Available from: https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.883372.

[10] Richard Szeliski, Structure from motion and SLAM, in: Computer Vision: Algorithms and Applications, Springer International Publishing, Cham, 2022, pp. 543–594, http://dx.doi.org/10.1007/978-3-030-34372-9_11.

[11] Martin A. Fischler, Robert C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM (ISSN: 1557-7317) 24 (6) (1981) 381–395, http://dx.doi.org/10.1145/358669.358692.

[12] Katsuyoshi Tsujita, Asumi Nishimura, Development of an autonomous soft docking system of small spacecraft using visual guidance, in: 2022 IEEE/SICE International Symposium on System Integration, SII, IEEE, 2022, http://dx.doi.org/10.1109/sii52469.2022.9708765.

[13] Claudio Vela, Giancarmine Fasano, Roberto Opromolla, Pose determination of passively cooperative spacecraft in close proximity using a monocular camera and AruCo markers, Acta Astronaut. (ISSN: 0094-5765) 201 (2022) 22–38, http://dx.doi.org/10.1016/j.actaastro.2022.08.024.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, http://dx.doi.org/10.1109/cvpr.2009.5206848.

[15] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90, http://dx.doi.org/10.1145/3065386.

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2015, http://dx.doi.org/10.1109/cvpr.2015.7298594.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, 2015, arXiv, http://dx.doi.org/10.48550/ARXIV.1512.03385.

[18] Mate Kisantal, Sumant Sharma, Tae Ha Park, Dario Izzo, Marcus Martens, Simone D'Amico, Satellite pose estimation challenge: Dataset, competition design, and results, IEEE Trans. Aerosp. Electron. Syst. 56 (5) (2020) 4083–4098, http://dx.doi.org/10.1109/taes.2020.2989063.

[19] Pedro F. Proença, Yang Gao, Deep learning for spacecraft pose estimation from photorealistic rendering, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 6007–6013.

[20] Duarte Rondao, Nabil Aouf, Mark A. Richardson, ChiNet: Deep recurrent convolutional learning for multimodal spacecraft pose estimation, IEEE Trans. Aerosp. Electron. Syst. (2022) 1–13, http://dx.doi.org/10.1109/taes.2022.3193085.

[21] Bo Chen, Jiewei Cao, Alvaro Parra, Tat-Jun Chin, Satellite pose estimation with deep landmark regression and nonlinear pose refinement, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop, ICCVW, IEEE, 2019, http://dx.doi.org/10.1109/iccvw.2019.00343.

[22] Duarte Rondao, Nabil Aouf, Mark A. Richardson, Vincent Dubanchet, Robust on-manifold optimization for uncooperative space relative navigation with a single camera, J. Guid. Control Dyn. 44 (6) (2021) 1157–1182, http://dx.doi.org/10.2514/1.g004794.

[23] Vincenzo Capuano, Kyunam Kim, Alexei Harvard, Soon-Jo Chung, Monocular-based pose determination of uncooperative space objects, Acta Astronaut. 166 (2020) 493–506, http://dx.doi.org/10.1016/j.actaastro.2019.09.027.

[24] Renato Volpe, Giovanni B. Palmerini, Marco Sabatini, A passive camera based determination of a non-cooperative and unknown satellite's pose and shape, Acta Astronaut. 151 (2018) 805–817, http://dx.doi.org/10.1016/j.actaastro.2018.06.061.

[25] Lei He, Duarte Rondao, Nabil Aouf, A novel mechanism for orbital AI-based autonomous refuelling, in: 2023 AIAA Guidance, Navigation, and Control Conference, American Institute of Aeronautics and Astronautics, 2023.

[26] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, Ruslan R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, 2012, arXiv, http://dx.doi.org/10.48550/ARXIV.1207.0580.

[27] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, Hao Li, On the continuity of rotation representations in neural networks, 2018, arXiv, http://dx.doi.org/10.48550/ARXIV.1812.07035.

[28] F. Landis Markley, John L. Crassidis, Fundamentals of Spacecraft Attitude Determination and Control, Springer, New York, 2014, http://dx.doi.org/10.1007/978-1-4939-0802-8.

[29] Roberto Cipolla, Yarin Gal, Alex Kendall, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, http://dx.doi.org/10.1109/cvpr.2018.00781.

[30] Leslie N. Smith, Cyclical learning rates for training neural networks, in: 2017 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2017, http://dx.doi.org/10.1109/wacv.2017.58.

[31] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv, http://dx.doi.org/10.48550/ARXIV.1412.6980.

[32] Joseph Redmon, Ali Farhadi, YOLO9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2017, http://dx.doi.org/10.1109/cvpr.2017.690.

[33] Joseph Redmon, Ali Farhadi, YOLOv3: An incremental improvement, 2018, arXiv:1804.02767.

[34] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size, 2016, arXiv, http://dx.doi.org/10.48550/ARXIV.1602.07360.

[35] Andrea Valmorbida, Mattia Mazzucato, Marco Pertile, Calibration procedures of a vision-based system for relative motion estimation between satellites flying in proximity, Measurement 151 (2020) 107161, http://dx.doi.org/10.1016/j.measurement.2019.107161.

[36] Lorenzo Pasqualetto Cassinis, Alessandra Menicucci, Eberhard Gill, Ingo Ahrns, Manuel Sanchez-Gestido, On-ground validation of a CNN-based monocular pose estimation system for uncooperative spacecraft: Bridging domain shift in rendezvous scenarios, Acta Astronaut. (ISSN: 0094-5765) 196 (2022) 123–138, http://dx.doi.org/10.1016/j.actaastro.2022.04.002.

[37] Alex Graves, Santiago Fernández, Jürgen Schmidhuber, Bidirectional LSTM networks for improved phoneme classification and recognition, in: Włodzisław Duch, Janusz Kacprzyk, Erkki Oja, Sławomir Zadrożny (Eds.), Artificial Neural Networks: Formal Models and their Applications – ICANN 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-28756-8, 2005, pp. 799–804, http://dx.doi.org/10.1007/11550907_126.