IMPROVING INERTIAL-BASED UAV LOCALIZATION USING DATA-EFFICIENT DEEP REINFORCEMENT LEARNING

Dimitrios Tsiakmakis, Nikolaos Passalis, and Anastasios Tefas
Computational Intelligence and Deep Learning Group
Artificial Intelligence and Information Analysis Lab.,
Department of Informatics,
Aristotle University of Thessaloniki, Thessaloniki, Greece
{*dtsiakma, passalis, tefas*}*@csd.auth.gr*

***Abstract*—Precise localization is a critical task for many Unmanned Aerial Vehicle (UAV)-based applications. Inertial-based navigation, which relies on Inertial Measurement Units (IMUs), is extensively used to this end, due to its low-cost and small footprint. However, IMU-based localization leads to accumulating significant localization errors. To overcome this limitation, in this paper we propose a data-efficient Deep Reinforcement Learning (DRL) method that enables learning how to correct localization errors from IMUs leading to more precise localization. In contrast with supervised approaches, the proposed method employs a novel data augmentation and regularization approach, which requires collecting a minimal number of real examples, while it is also platform-agnostic and can account for manufacturing impressions. The effectiveness of the proposed method is demonstrated both in a simulation environment, as well as using a real UAV.**

***Index Terms*—Deep Reinforcement Learning, Inertial-based Localization, Data Augmentation**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly used in various applications, ranging from precision agriculture [1] and search and rescue missions [2] to indoor surveillance [3]. A common point between these applications, along with virtually every UAV-based application, is the need for precise UAV localization. UAV localization is critical both for mission control purposes, i.e., some tasks are related to the location of a UAV, as well as for safety purposes, i.e., avoid flights over restricted areas. Several different approaches have been developed for UAV localization, with each one relying on different sensors and providing a different level of accuracy.

Perhaps among the most well known localization approaches is using satellite-based radio-navigation systems, such as the Global Positioning System (GPS) [4], [5]. Despite its low cost the accuracy of GPS and related systems is usually low. Indeed, according to the official GPS documentation, GPS-enabled devices are normally accurate to within a 4.9 meters (16 feet), which is unacceptable for many applications. At the same time, there are several locations where there is no GPS coverage [6], while such approaches cannot be used indoors. The use of real-time kinematic positioning can further reduce the errors introduced in satellite-based radio navigation [7], yet it typically requires the use of extra base stations, which increases the cost and reduces the flexibility of UAVs. Light detection and ranging approaches [8], [9], also known as LIDAR, can be also used to provide accurate localization, especially when coupled with simultaneous localization and mapping (SLAM) approaches [10]. However, such approaches involve the use of very expensive sensors and they have greater computational and energy demands.

On the other hand, the use of Inertial Measurement Units (IMUs) [11], which is a combination of accelerometers, gyroscopes, and magnetometers can provide very low-cost solutions that also do not rely on any kind of external hardware or communication (e.g., satellites, base stations, etc.). The localization is accomplished by utilizing IMU data for dead reckoning, called Inertial Navigation System (INS) [12]. The recent demand for smaller sensors that can be integrated into cutting-edge technologies, has prompted engineers to build a Micro Electro-Mechanical System (MEMS) which can provide low-cost and low-footprint sensors that can be very easily integrated with virtually any UAV and provide real-time measurements. Despite the cost and flexibility benefits of such systems, they also come with accuracy limitations. IMU sensors monitor the linear acceleration and rotational velocity of the body with just a very small degree of inaccuracy every time. However, over long periods these errors can accumulate leading to significant position drifts that can comprise their application, especially when used as a sole localization sensor in mission critical applications.

These limitations have fueled research on methods for improving inertial-based navigation for UAVs [13], [14], [15]. Many recent approaches built upon Deep Learning (DL)-based models that allow for significantly improving the localization process. However, despite these improvements, these approaches suffer from a significant drawback. They mostly rely on supervised learning (either regression-based

This work was supported by the European Union's Horizon 2020 Research and Innovation Program (OpenDR) under Grant 871449. This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

or classification-based), which in turn requires a large number of samples to be collected and annotated to train the corresponding methods. At the same time, such approaches are typically linked to the hardware used for data collection and their performance deteriorates when deployed on different hardware, requiring collecting data again and re-training the models. Furthermore, even when using the same hardware, manufacturing tolerances might lead to sensors that have different noise characteristics, which make the application of supervised learning approaches challenging.

Deep Reinforcement Learning (DRL) can overcome these limitations [16], since it enables autonomous agents to learn just by interacting with the environment. Indeed, DRL methods have shown to achieve remarkable results in a variety of tasks in recent years, often outperforming humans [17], [18]. However, directly applying DRL for improving inertial-based navigation for UAVs is not directly feasible since: a) a feedback signal is still required in order to measure the quality of the learned policy and b) a large number of episodes are typically required for learning. Even though the first limitation can be easily addressed, e.g., by using visual cues to provide a feedback signal, the low-data efficiency of DRL approaches still pose a significant limitation that prohibits such approaches from being deployed in practice.

Based on the aforementioned observations, in this work we propose a pipeline that can allow for easing these limitations, enabling data-efficient DRL on UAVs for inertial-based navigation. The proposed method employs a two-stage pipeline. In the first stage, a backbone is trained using supervised learning in a simulator. Acquiring ground truth annotations in a simulator is easy and cheap, so this approach can enabled us to train a backbone that can capture the dynamics of the behavior of IMUs without targeting a specific sensor.

Then, the employed DL model is fine-tuned using DRL on a real UAV. In our paper, we focused on improving the data efficiency of DRL methods when applied to the problem of inertial-based UAV localization error correction, which is not a simple combination of existing methods. We concentrated on developing methods that would allow us to train and deploy DRL approaches directly on UAVs, which requires minimizing the amount of data gathered. The selection of DRL was not arbitrary; we proceed with this framework due to the difficulty of acquiring ground truth data in real time.

Since this can be an especially data-intensive process, we further propose: a) a data augmentation method that can generate multiple simulated episode trajectories just from one real episode; this is essential in order to maximize the quantity of information that might be exploited without running repeated episodes, and b) a regularizer than can provide additional feedback when fine-tuning the learned policy based on the sign of the measured reward signal. These methods are intended to minimize the experiments performed using a real UAV, as well as are problem-specific and, to the best of our knowledge, neither has ever been proposed. For acquiring a reward signal, we propose a simple, yet efficient visual landmark-based approach that can be used even with low-resolution cameras. As we demonstrated through extensive experiments on regressing the 2D position of a UAV, the proposed method can indeed lead to significant performance improvements over the employed baseline approaches.

The rest of the paper is structured as follows. First, Section II introduces the proposed methodology, while the experimental evaluation of the proposed method is provided in Section III. Finally, conclusions are drawn in Section IV.

II. Proposed Method

A. Background

The simplest method to localize a UAV using a inertial-based approach is to employ a first-order numerical approach to solve ordinary differential equations (ODEs), which is sometimes referred to as Euler's method. Specifically, Euler's method employs the basic formula:

$$y(t+h) = y(t) + h * f(t,x), \tag{1}$$

where the $f(t,x)$ is simply the dx/dt amount. In our case, time-step is represented by h, time by t, position by x, and velocity by $f(t,x)$. Thus, we estimate the next instant position, taking into account an initial position at every constant time-step. Note, we assume that velocity between two measurements remains constant throughout the flight. This simple approach enables UAV localization through IMU sensors that can be provide acceleration/speed estimates. However, the noise that it is introduced by IMUs can lead to a significant drift in the estimation of UAV position using this approach.

Neural Networks (NNs) can be employed in a supervised learning setting in order to learn how these errors should be corrected, allowing for improving the localization accuracy. Let $G_{\mathbf{W}} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ denote a regression model, parameterized by weights $\mathbf{W}$, with m inputs and n outputs. Also, let $\mathbf{v} \in \mathbb{R}^m$ denote a vector that contains the most recent velocity measurements, including the current one, provided by the IMU. Then, the model $\mathbf{y} = G_{\mathbf{W}}(\cdot)$ can be trained to provide corrected estimates for the current velocity, denoted by $\mathbf{y} \in \mathbb{R}^n$. Note that typically $n = 2$, since we are interested in estimating the speed in the 2d plane, ignoring the speed in the vertical to this plane axis (height), since altimeter sensors can provide reliable estimates for the vertical speed. Similarly, m is typically set to $m = 2 \times T$, where T denotes the history (number of time steps) to include in the input that will be fed to the neural network that will provide the corrected speed estimates. Training $G_{\mathbf{W}}(\cdot)$ is straightforward, since we just need to collect enough training samples of IMU velocity estimates and the corresponding ground truth velocities. Then, the mean square error can be used for training the neural network estimator using gradient descent. Furthermore, note that typically the estimator $G_{\mathbf{W}}(\cdot)$ is fitted to regress the velocity errors instead of the actual velocities, since this accelerates the learning process. After estimating the velocity error, then the corrected velocity can be used in (1) to acquire a more reliable estimation of the UAV's position.

B. Data-efficient DRL-based training

Even through the aforementioned process can be easily performed inside a simulation environment, it is very expensive to perform using real UAVs, since extra equipment is required for measuring the accurate position of a UAV and a large number of samples need to be collected. Therefore, in this paper we propose a two step pipeline that consists of the following steps: a) train a generic DL-based backbone model in a simulator to correct generic IMU errors and b) fine-tune this model on a real UAV using DRL. This process can overcome the need to collect a large number of annotated training samples using a real UAV. However, as mentioned in Introduction, DRL methods are also data intensive. To overcome this limitation, we proposed to use a data augmentation method coupled with a regularizer that can increase data efficiency.

In this work, we propose to employ a DRL agent in order to provide *continuous corrections* to UAVs estimates. More specifically, we introduce a *virtual agent* that controls the estimation of the UAV's position. Hence, there are two positions: the actual UAV and a sphere indicating its estimated position. The DRL agent controls the latter by providing continuous corrections in the two axes of the 2d plane. This setup also enables an easy way to acquire the feedback signal for training the agent both in simulation and in real word. More specifically, in simulation, for each episode the UAV runs a predetermined course, e.g,. 2 meters to the North and 1 meter to the East. Then, when the episode is finished, we project the virtual UAV's position as a black mark onto the floor, and then, the UAV uses its camera to snap an image and provide the reward signal. To present this concept with an example to be more intuitively, if the position of the UAV is accurate, the black mark will be centered in the captured image. In contrast, the black mark would be in a different location if the positions of the actual and virtual locations are different. Then, the reward for each axis k can be calculated as:

$$R_k = \frac{1}{1 + |p_k|}, \tag{2}$$

where p_k is the distance in pixels between the black mark and the center of the captured image (which represents the position of the UAV). In real deployment, the black mark will represent the desired UAV position based on the provided control command. Then, the reward can be calculated in a similar fashion and provide the same behavior (maximize as the agent better corrects the displacement estimations). This process enables training the DRL agent without having access to ground truth data regarding the actual speed and/or displacement on each step.

In this work, we employ Proximal Policy Optimization (PPO) [19] for training the agent. This is without loss of generality, since any DRL method that can support continuous action spaces can be used. Furthermore, since the aim is to accelerate the learning process as much as possible, we employed the supervised learning model that was pre-trained on the simulator to initialize the weights of the actor model. Therefore, the DRL method is employed to fine-tune the DL model to the actual hardware used in the UAV. To further increase the efficiency of the learning process we designed and used a data augmentation method to create additional episodes during the training. The main concept is that the reward of an episode remains unchanged if the angle of velocity vectors and the actions are rotated simultaneously. To this end, the proposed method selects the episode with the highest reward from the buffered episodes and then several synthetic episodes are created by rotating the velocities and actions by a random angle $\phi \in [0, 360)$.

Finally, to further increase the learning speed and minimize the number of training episodes required to fine-tune the agent to the actual IMU used, we propose employing a hint regularizer that provides additional supervision based on whether the agent is currently overshooting or undershooting the desired position (as indicated by the sign of the distance in (2)). Therefore, the regularizer for each axis is defined as:

$$\mathcal{L}_{reg,k} = -\alpha_{reg} \cdot \delta_k \cdot g_{RL}(x) \tag{3}$$

where α_{reg} is the weight of the regularizer, δ_k it is a binary variable $\{-1, 1\}$ indicating whether we are currently overshooting or undershooting the target position and $g_{RL}(x)$ is the agent's output. Then, the overall loss is calculated by simply adding the regularizer for both axis to the PPO loss.

III. Experimental Evaluation

We conducted experiments using both a simulated environment, i.e., for supervised learning and validation of the proposed DRL approach, as well as a real UAV. For the simulated experiments, we employed Webots [20]. For supervised learning, we collected 500 episodes with velocities and ground truth positions. We also experimentally found that the IMU measurement is biased depending on the vehicle's velocity and it is always underestimated. Therefore, we estimate the velocity bias in the simulation environment as:

$$v/(1 + 1/(1 + c * |v|)), \tag{4}$$

where c is an IMU-depended factor and v is the ground truth velocity. For the supervised learning model, we used $c = 5$, while for the evaluation we used in all cases $c = 2$ to simulate the drift that can occur due to hardware changes.

The IMU was pooled with a frequency of 25Hz, while each episode has a total length of 10 seconds. We also used an MLP with two hidden layers as a backbone, with 12 neurons each with the `tanh` activation. Then, the network culminates in two branches that output corrections for each dimension. In every branch, there are two extra trainable parameters, which are used for shifting and scaling the output of the network. We found that when we re-train the network with new data from alternative sensors, the convergence succeeds more quickly due to these variables, which allows for promptly shifting and scaling the output without refitting all the weights of the backbone. Similar results, yet for input normalization, have also been reported in the literature [21]. The network receives a one-second time frame of velocities, i.e., 25 measurements

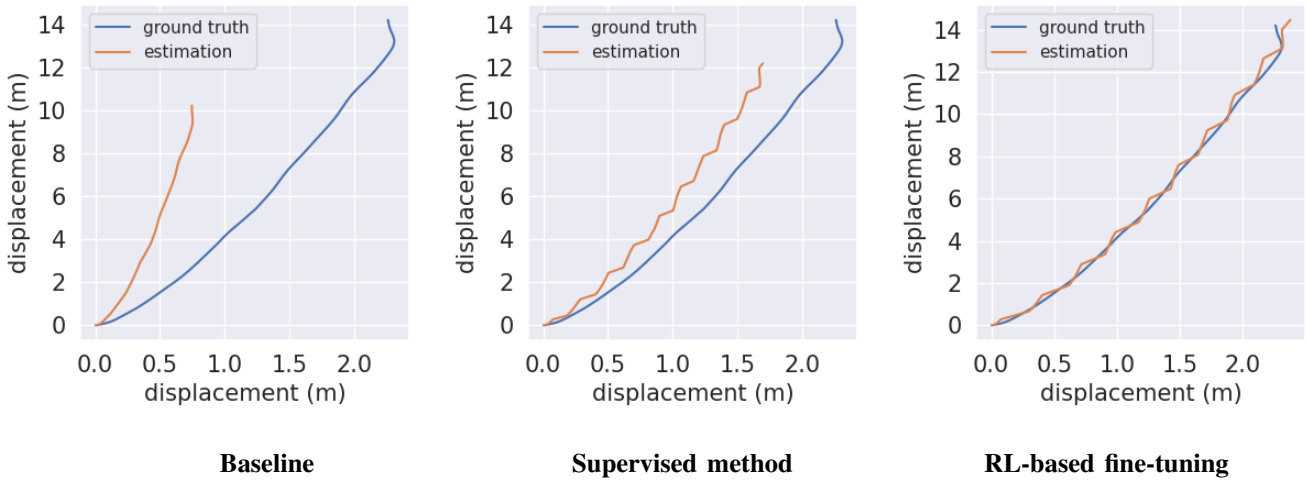

Fig. 1. Comparison between baseline (Euler's method) (left), supervised training (middle) and RL-based fine-tuning (right). Each axis corresponds to the displacement of a UAV in the 2d space when flying on a pre-determined course.

along each of the two axes, and returns two corrections, one of each axis. The mean squared error was used for training the supervised model using the Adam optimizer and a learning rate of 3×10^{-2}. The optimization ran for 10 epochs with a batch size of 512, while the learning rate was reduced when the learning process approached a plateau (the reduction factor was set to 0.1 and patience to 10). For the DRL setup, we used the same configuration as the previous model for the actor, while an MLP with 2 hidden layers with 12 neurons each was used as a critic. The models were trained for 30 epochs with a learning rate of 9×10^{-5} and 5×10^{-3} in actor and critic accordingly. PPO algorithm was used for fitting the DRL agent, while the clipping factor was set to 0.15.

First, we evaluated the proposed method in simulation using Webots. The experimental comparison between Euler's method with no corrections, the supervised method trained on a model with $c = 5$, and the proposed RL-based fine-tuning of the supervised model are shown in Fig. 1. As demonstrated, using DRL to fine-tune the model trained in simulation to adjust the actual characteristics of a specific UAV leads to significant improvements. Based on these observations we evaluated the ability of the proposed data-efficient RL approach compared to the baseline Euler method. The results are reported in Table I where we report the mean squared error (MSE), mean distance (MD), mean positional error (MPE), and absolute trajectory error (ATE) between the ground truth displacement and the one estimated by the DRL models. These results indicate that the proposed method can improve DRL agents' performance when training under a constrained number of episodes (i.e., 30 episodes). Note that as the duration of an episode increases, the error still accumulates. Nonetheless, the proposed method manages to significantly reduce all the error metrics compared to the baseline. Finally, we also validated the proposed method using data collected from a DJI Mavic mini 2 UAV using different velocities. The results reported in Table II again confirm that for a wide range of different speeds the proposed method still leads to better performance.

TABLE I
DRL FINE-TUNING EVALUATION ON WEBOTS USING A DISTRIBUTION SHIFT SCENARIO (c CHANGES FROM 5 TO 2).

	10 secs		
metrics	baseline	supervised	**proposed**
MSE	12.297	2.975	**0.101**
MD	3.438	1.692	**0.286**
MPE	1.714	0.931	**0.212**
ATE	2.008	1.073	**0.251**
	100 secs		
MSE	1225.922	312.100	**18.832**
MD	34.437	17.444	**3.816**
MPE	18.376	141.05	**2.053**
ATE	21.014	10.696	**7.613**

TABLE II
DRL FINE-TUNING EVALUATION USING A DJI MAVIC 2 UAV. THE PERCENTAGE OF ESTIMATED DISTANCE COVERED TO THE TRUE DISTANCE (MPE, %) IS REPORTED FOR DIFFERENT FLYING SPEEDS.

Vel. (m/s)	0.1	0.3	1.1	1.4	2.2	2.8
baseline (%)	46.58	86.20	87.51	94.46	96.06	95.70
proposed (%)	**80.59**	**98.14**	**106.89**	**99.31**	**102.98**	**98.14**

IV. CONCLUSIONS

In this paper, we proposed a data-efficient DRL approach for improving inertial-based navigation for a UAV. The proposed method employed a two-stage pipeline: in the first stage, a backbone is trained using supervised learning, while in the second stage a data-efficient DRL-based approach for fine-tuning is used. We demonstrated that the proposed method can indeed allow for improving inertial-based navigation, focusing on cases where the IMUs used in UAVs can have different characteristics requiring UAV-specific fine-tuning using a very small number of real episodes.

REFERENCES

[1] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Thomas Lagkas, and Ioannis Moscholios, "A compilation of uav applications for precision agriculture," *Computer Networks*, vol. 172, pp. 107148, 2020.

[2] Ebtehal Turki Alotaibi, Shahad Saleh Alqefari, and Anis Koubaa, "Lsar: Multi-uav collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55817–55832, 2019.

[3] Natthawat Boonyathanmig, Sarun Gongmanee, Prachaya Kayunyeam, Piyavat Wutticho, and Sethakarn Prongnuch, "Design and implementation of mini-uav for indoor surveillance," in *Proceedings of the 9th International Electrical Engineering Congress (iEECON)*, 2021, pp. 305–308.

[4] Salah Sukkarieh, Eduardo Mario Nebot, and Hugh F Durrant-Whyte, "A high integrity imu/gps navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 572–578, 1999.

[5] Songlai Han and Jinling Wang, "Integrated gps/ins navigation system with dual-rate kalman filter," *GPS Solutions*, vol. 16, no. 3, pp. 389–404, 2012.

[6] "GPS Accuracy," https://www.gps.gov/systems/gps/performance/accuracy/, Accessed: 2022-07-21.

[7] Patrick Henkel, Ulrich Mittmann, and Michele Iafrancesco, "Real-time kinematic positioning with gps and glonass," in *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1063–1067.

[8] Michel Jaboyedoff, Thierry Oppikofer, Antonio Abellán, Marc-Henri Derron, Alex Loye, Richard Metzger, and Andrea Pedrazzini, "Use of lidar in landslide investigations: a review," *Natural Hazards*, vol. 61, no. 1, pp. 5–28, 2012.

[9] Frederick G Fernald, "Analysis of atmospheric lidar observations: some comments," *Applied Optics*, vol. 23, no. 5, pp. 652–653, 1984.

[10] Dinh Van Nam and Kim Gon-Woo, "Solid-state lidar based-slam: A concise review and application," in *Proceedings of th IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2021, pp. 302–305.

[11] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M Khairi, and Vijayabaskar Kasi, "Reviews on various inertial measurement unit (imu) sensor applications," *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.

[12] Billur Barshan and Hugh F Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 328–342, 1995.

[13] Martin Brossard, Axel Barrau, and Silvere Bonnabel, "Rins-w: Robust inertial navigation system on wheels," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2068–2075.

[14] Santiago Cortés, Arno Solin, and Juho Kannala, "Deep learning based speed estimation for constraining strapdown inertial navigation on smartphones," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018, pp. 1–6.

[15] Sachini Herath, Hang Yan, and Yasutaka Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3146–3152.

[16] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 31, 2018.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al., "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[20] Olivier Michel, "Cyberbotics ltd. webots™: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 5, 2004.

[21] Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis, "Deep adaptive input normalization for time series forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3760–3765, 2019.