User Perceive Realism of Machine Learning-based Drone Dynamic Simulator

Damitha Sandaruwan[1], Nihal Kodikara[2], Piyumi Radeeshani[3],
K.T.Y. Mahima[4], Chathura Suduwella[5], Sachintha Pitigala[6] and Mangalika Jayasundara[7]
University of Colombo School of Computing,
35 Reid Ave, Colombo 00700, Sri Lanka[1,2,3,4,5]
Department of Statistics Computer Science, University of Kelaniya
Sri Lanka[6,7]

***Abstract*—The drone will be a commonly used technology by a significant portion of society, and simulating a given drone dynamic will be an essential requirement. There are drone dynamic simulation models to simulate popular commercial drones. In addition, there are many Newtonian and fluid dynamics-based generic drone dynamic models. However, these models consist of many model parameters, and it is impracticable to evaluate the required model parameters to simulate a custom-made drone. A simple method to develop a machine learning-based dynamic drone simulation model to simulate custom- made drones mitigates the issues mentioned above. Specifically, the authors' research is associated with the development of a machine learning-based drone dynamic model integrated with a virtual reality environment and validation of the user-perceived physical and behavioural realism of the entire solution. A figure of eight manoeuvring patterns was used to collect the data related to drone behaviour and drone pilot inputs. A Neural Network-based approach was employed to develop the machine learning-based drone dynamic model. Validations were done against real-world drone manoeuvres and user tests. Validation results show that the simulations provided by machine learning are accurate at the beginning and it decreases the accuracy with time. However, users also make mistakes/misjudgments while perceiving the real-world or virtual world. Hence, we explored the user perceive motion prediction accuracy of the simulation environment which is associated with the behavioural realism of the simulation environment. User tests show that the entire simulation environment maintains substantial physical realism.**

***Keywords*—Drone; simulation; machine learning; drone dynamics; virtual reality**

I. Introduction

Drone is an unmanned aircraft controlled by a ground base station, used primarily to carry out air-based missions, such as surveillance, transportation and entertainment [1], [2]. Historically drones were first used by the military for ground surveying and spying missions, and they were called Unmanned Aerial Vehicles (UAVs) [3], [4].

Nowadays the term drone is widely used for quadcopters, vertical takeoff and land (VTOL) type miniaturized aircraft, with four propeller blades, pushing air downwards to maintain itself on air, while moving back and forward using the thrust created by the same propellers [5]. Such drones come in many sizes ranging from a few inches up to several feet and are used in many applications including for toys, entertainment, photography, land surveillance and scientific research [1], [6], [7], [8].

According to the Stanford University Intelligent Systems Laboratory and National Aeronautics and Space Administration (NASA), within the next few years, Low-Altitude airspace will be congested with Low-Altitude Remotely Piloted Aircrafts [9]. This will be a common technology used by a significant portion of society. Due to the maturing of the technology, it requires a skilled and qualified human resource to use the technology and train pilots for specific tasks and missions. At the same time, it allows individuals and various organizations to build custom-made drones at their own pace.

Therefore, drone pilot training for custom-made drones via simulations is essential. It provides the knowledge and practical skills that are necessary to safely and efficiently operate unmanned aircraft for commercial and non-commercial use [10], [11]. Hence, drone pilot training simulators will be a very important requirement. Depending on the safety-critical level of the drone operation, there is a requirement of simulates the drone operations prior to the real real-world operation. This type of simulation supports the identification of potential disasters due to the manoeuvring capability of the drone and the pilot. Hence, drone simulators with required realism will be essential in future [1].

There are solutions to simulate real-time drone dynamics, such as the DJI drone simulator. These simulators are capable of simulating the bundled series of drones belonging to a particular vendor. There are many proposed generalized drone dynamic simulation models and most of these models are based on Newtonian dynamics and fluid dynamics. Simulating custom-built drones with these generalized drone dynamics is a highly challenging task. It requires evaluating model parameters related to the custom build drone and it needs many experiments to be conducted with the given ideal conditions, domain-specific knowledge and a wide range of practical issues. A machine learning-based drone dynamic model avoids most of the above issues. There is significant value to explore the possibility of developing a machine learning-based drone dynamic model and incorporating it with a Virtual Reality (VR) environment to simulate a custom-built drone. It is highly essential to evaluate the realism level of such a drone dynamic model and the VR environment. Overall realism of a virtual environment can be expressed with physical realism and behavioural realism. Behavioural realism expresses the accuracy of dynamic activities such as motion predictions. Physical realism expresses the physical infrastructure of the simulated environment [12], [13], [14], [15].

The remainder of this paper is structured as follows: State-of-the-art drone simulation models are critically evaluated in Section II. Section III discusses initial research initiatives such as data collection techniques. The development processes of the machine learning model and VR environment for the proposed dynamic drone simulator are covered in Sections IV and V, respectively. Section VI presents the experimental and validation results of the simulator under different criteria. Finally, Section VII, concludes the paper along with prospective research directions.

II. Related Works

There are many proposed drone dynamic simulation models and most of these models are based on Newtonian dynamics and fluid dynamics [16], [17]. These models were used to simulate drones for different applications and comparisons of different simulation models were reported [18], [19]. However, the evaluation of required model parameters and simulations of existing drones is not discussed.

There are commercial simulators that could use for the simulation of drones. For example, the DJI Assistant 2 software is a drone simulator provided by DJI used to simulate selected DJI drones [20], [21]. The DJI Assistant 2 is programmed/programmable to simulate drones with offline remote control data. Real Flight drone/flight simulator [22], Simpro drone simulator [23], Liftoff by Immersion RC [24] and HELI-X professional R/C flight simulator [25] are some of the reviewed commercial drone simulators which can be employed to simulate particular commercial drone.

There are many research and development works carried out by employing machine-related theories in drone-related ICT solution developments [26], [27], [28], [29], [30], [31], [32]. However, most of these applications are related to designing and developing autonomous drones and target tracking in outdoor/indoor environments.

Jemin et al. presented a method to control a quadrotor with a neural network trained using reinforcement learning techniques [33]. They demonstrated the performance of the trained policy both in simulation and with a real quadrotor. The trained policy shows outstanding performance and remains computationally cheap simultaneously. Furthermore, it shows many other advantages of neural network policies that are not limited to their versatility. This experiment is limited to a small space covering approximately a 2m x 2m x2m controlled area.

Osman Çakira and Tolga Yükselb [34] developed a neural network-based controller for quadrotors. In this study, neural network control of quadrotors is aimed to obtain an artificial intelligence-based drone controller and the results show that neural network controllers achieve satisfactory trajectory tracking results. This experiment is also limited to a small space that approximately covers 5m x 3m x3m controlled area.

Jeong, Baek and Lee propose a prediction model of the vehicle trajectory [35]. Their approach is not based on physics-based motion models and there are no kinematic and dynamic models, laws of physics and fluid dynamics. They employed a Deep Neural Network that takes as input vehicle velocity, acceleration, yaw rate, steering, and road curvature. The authors discussed the advantages of employing deep neural network-based predictions to avoid several potential issues in similar physics-based vehicle simulation requirements.

Jeong et al. proposed Deep Neural Network (DNN) that considers preprocessed vehicle velocity, acceleration, yaw rate, steering, and road curvature as the input layer and eventually reaches the output layer via multiple hidden layers. DNN uses an activation function with a function called a rectified linear unit (ReLU) [36], [37]. However, a ReLU function is not a perfect match to be used in this study because the final outputs of our DNN model include expected future lateral movements, which can be negative. Hence, Leaky ReLU was employed for the activation function, which is slightly different from the original ReLU [35]. The results of the proposed study confirm the feasibility of Deep Neural Network-based long-term trajectory prediction for vehicles driving on roads with a certain level of road conditions such as varying curvature.

Jackson et al. employed the machine-learning technique to design and develop a dynamic model of a rotorcraft [38]. Their key justification for this approach is that widely used physical-law-based and substantially accurate rotorcraft dynamic models need to be simplified to make real-time motion predictions. Hence, it leads to motion prediction errors compared to the real vehicle. In their current work, machine-learning techniques are employed to train a rotorcraft dynamic model to predict the dynamic on-axis motion responses such as pitch rate, roll rate and yaw. The employed machine learning was designed with a Gaussian Process (GP) non-linear autoregressive model [39]. They have proven that the machine-learning approach can be successfully utilized to predict the on-axis motions of a rotorcraft. The obtained level of accuracy is generally higher than the physics-based dynamic models.

Punjani proposed a helicopter dynamic modelling method with a Rectified Linear Unit (ReLU) Network Mode [40]. The reasons for selecting this approach are helicopter has a complex dynamic system with rigid body dynamics with aerodynamics, engine dynamics, vibration and other factors such as manoeuvring patterns. They described several baseline models and shows that the helicopter dynamics with ReLU significantly outperformed other considered baseline models. Furthermore, It improves acceleration prediction over state-of-the-art methods and they presented performance gains techniques with hyperparameters fine-tuning. They selected a range of manoeuvres such as forward/sideways flight, vertical sweeps, inverted vertical sweeps, stop-and-go, flips, loops, turns, circles, dodging, orientation sweeps, orientation sweeps with motion, gentle freestyle and aggressive freestyle.

Considered, baseline models are based on Linear Acceleration Model and it serves as a direct state-of-the-art performance baseline. Validations, efficacy investigations, compare and contrast among baseline models and the ReLU-based model were done by using data obtained from the Stanford Autonomous Helicopter Project [41]. Following Fig. 2 presents observed and predicted accelerations in the up-down direction for selected three different aerobatic manoeuvres. It shows that the baseline Linear Acceleration model performs poorly compared to the novel ReLU Network Model [40].

Sandaruwan et al. proposed a machine learning-based approach to simulate drone dynamics related to the figure of

Eight Manoeuvering pattern [1]. Authors have obtained satisfactory results and proven that the machine-learning approach can be successfully utilized to predict drone motions. However, they have not published the evaluation of the developed drone dynamic model and simulation environment.

Zhang et al. investigated users' situational awareness of virtual outdoor, virtual indoor, real-world indoor, and real-world outdoor environments. They considered distance judgment for their experiments and identified potential error factors and other considerations [42]. They also investigate the impact of users' real-world scene awareness on distance judgment in the same simulated scene in a virtual environment. Their results suggest that both the virtual and real-world environments have an impact on distance judgment in VR which affects the users' ability to perceive the realism of VR environment [42].

Ziemer et al. also explored the order in which people experience real and virtual environments that influence their distance estimates. They also identified potential error factors, error percentages and other considerations in estimating distance in real and virtual environments. Their results provide facts and figures to measure users' situational awareness of the virtual and real scenes [43]. They presented the importance of presence and reality judgment in the VR environment and aspects that contribute to creating a person's reality judgment in a given scenario in the VR environment. They discuss the problem of "How do people decide whether something is real or not?". Their work aims to design a self-report measure that assesses both constructs [44].

Sandaruwan et al. conducted several research studies to design and deploy a Maritime VR Environment [14], [13]. They discuss essential factorsconsiderations related to validating the user perception of the physical & behavioural realism of a maritime VR environment. They employed several techniques & methods to validate the physical & behavioural realism of the maritime VR environment. Moreover, they used techniques such as comparing real vehicle trials and simulated vehicle trials, simulation of simple possible scenarios and user tests [15]. However, those techniques & methods are applicable to most VR environments with dynamic vehicle models.

By considering the above-reviewed literature and the main objective of designing and developing a machine learning-based drone dynamic model for outdoor simulation of a custom-built drone, the following actions were executed: 1) Deep Neural Network-based approach with Leaky ReLU or any other appropriate activation function. 2) The figure of eight manoeuvring pattern-based outdoor data collection to build the machine learning model. 3) Conduct evaluation/validation such as comparing real drone trials and simulated drone trials, simulation of simple possible scenarios and user tests to measure the user perceive realism of the develop machine learning-based drone dynamic model.

Analysis of Newtonian dynamic and fluid dynamic-based analytical solution used by the researchers [16], [17], [18], [19] shows that takeoff & landing of the drone is more complicated and needs more accurate complex dynamics. In addition, real-world wind flows with a turbulent effect make the situation further complicated. Hence, this research does not focus on the takeoff & landing of the drone, and all experiments and tests were carried out in a calm outdoor environment with negligible wind effect.

III. Preliminaries and Data Collection

Four-rotor drones are a subset of multirotor systems and these drones use four rotors to keep them flying. A popular example of these multirotor drones is the widely used Phantom drone made by the SZ DJI Technology Co. Ltd [45]. Drone movements are controlled by a handheld radio controller. The drone changes its position and orientation based on the given radio controller inputs. This research focuses on a machine learning-based four-rotor drone dynamic model to build the relationship between the drone pilot's radio controller inputs and the drone's position and orientation. The Phantom drone by the SZ DJI Technology Co Ltd was selected for the experiments and it has six degrees of freedom motions (Three rotational motions & three translational motions).

DJI Phantom quadcopter drone comes with four propulsors that enable vertical takeoff and landing. It has four key controllable variables which move the drone in the 3D space [45], named throttle, pitch, roll & yaw. Fig. 1 illustrates a radio controller with key controllable variables. Based on the key controllable variable inputs, the drone changes its position and orientation.

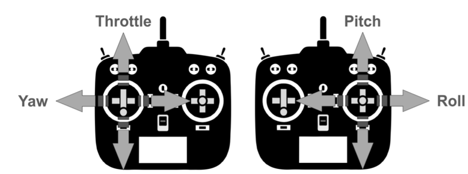

Fig. 1. A radio controller with key controllable variables.

Movement in the horizontal frame is achieved by tilting the platform with the different thrusts of the motors. Vertical movement is achieved by changing the total thrust of the motors.

DJI Phantom 4 [45] consists of many inbuilt sensors such as accelerometers, proximity sensors, GPS and GLONASS. That sensor assists with precision flying, precise hovering and much more. In addition, it provides a flight log consisting of flight position and orientation data & radio controller input data with the frequency of 10 sample points per second (10Hz). DJI Phantom 4 Pro [46] drone maintains 1.5-meter position accuracy with inbuilt GPS sensors & it can be enhanced up to 1cm accuracy by configuring external sensors.

The required machine learning-based drone dynamic model must predict the drone's position and orientation against the radio controller input. Hence, the required data can be categorized into two main categories as given below:

- Drone pilot inputs are entered via the drone Radio Controller (RC) and represent the given inputs, such as throttle and rudder values that are responsible for the drone position and orientation changes.
- The Drone's position and orientation change with time, and all other relevant sensor information vary with time (E.g. Battery level, accelerations, velocities).

If all manoeuvres are performed with fully charged batteries and the battery level is within 97% and 100%. Then it reduced the potential performance variation of the drone due to the battery power variations. The recorded data set consists of over forty parameters that vary with time. After examining the time-varying fields of the raw data set, the following observations were made:

- Certain sets of parameters directly imply the other set of available parameters. E.g. Latitude, Longitude and Altitude present the position of the drone in 3D space. All data stamps were recorded with a constant time gap. VelocityX ,VelocityY and VelocityZ also present the position of the drone in 3D space.
- There are certain sets of parameters that are constant during the experiment or indicate negligible variation. E.g. GpsCount, GpsLevel Battery Power(%), Battery Voltage, Battery Voltage Deviation, Battery Cell Voltages.
- There are other sets of parameters that are static due to the selected settings of the drone or provide Tips and warnings. E.g. App Tip, App Warning, App Message and Flight Mode.

Hence, after considering the above fact extracting the required data and selected key parameters of the data set are described below:

- Time (seconds): Time elapsed since the power-up.
- RcAileron: Rc signals for roll.
- RcElevator: Rc signals to control the horizontal pitch attitude of the drone.
- RcRudder: Rc signals to control the yaw of the drone.
- RcThrottle: Rc signals control the engine's speed and indicate how fast or slow the drone's movement.
- x,y,z: Cartesian conversion of Longitude and Latitude.
- Orientation: Bearing of the head of the drone in degrees.

The above parameters describe the drone pilot input via the radio controller & resultant position and orientation of the drone. Hence, the above parameters were taken into consideration to develop a machine-learning drone dynamic model.

IV. Development of Machine Learning Model

This research focuses on a machine learning-based drone dynamic model based on a subset of artificial intelligence. Machine learning consists of several subsections. Deep learning is one of the subsets of machine learning in which artificial neural networks adapt and learn from a large amount of data [47].

Machine learning models/methods or learnings are based on what it has learned only. Neural network structures/arrange algorithms in layers of fashion that can learn and make intelligent decisions on their own [48]. Neural networks are more suited to solve complex machine-learning problems. Neural networks can learn and model the relationships between inputs and outputs that are nonlinear and complex. This can be used to generalize input-output relationships, and reveal hidden relationships, patterns and predictions. It supports modelling highly volatile time series data and capable of predictions [49]. Neural networks require much more data than traditional Machine Learning algorithms to complete the model development. Depending on the requirement, building a customized neural network model that is perfectly suited. However, it takes more time compared to the traditional ML algorithm. A neural network consumes a longer time to train rather than a traditional machine learning model. It requires continuous computational resources, depending on the architecture of the neural network and the size of the data [50].

This research deals with complex rapidly changing input data set and output data set (Drone pilot's radio controller inputs and drone's position and orientation). The relationships between inputs and outputs are nonlinear. A single drone pilot trial consists of thousands of data points and the entire data set consists of over 200 thousand data points. A continuous computational resource is not a vital issue with the available technological infrastructure. Hence, an Artificial Neural Network (ANN) based approach was selected.

An artificial neural network has parameters that cannot be directly estimated from the data. This type of model parameter is referred hyperparameter. No analytical solution is available to calculate appropriate values for hyperparameters [51]. An artificial neural network has many hyperparameters. However, two key hyperparameters are the number of layers and nodes in each hidden layer. It controls the entire architecture/topology of the artificial neural network. In addition, there are other hyperparameters such as activation function, the number of epochs, batch size, learning rate, Mini-batch size, and Learning rate, which are identified as other potential hyperparameters [52].

Several frameworks and libraries have been developed in the last few years to fulfil machine learning-related necessities. Industry and academia use various frameworks and libraries to expedite the neural network-based model development, training and good results. Hence, model development and training have become easier. Based on the star ratings on Github, and our similar project experiences in the field, TensorFlow [53] was selected as the most effective and easy-to-use framework and library.

TensorFlow is a full-fledged open-source deep learning framework designed and developed by Google. It was initially released in 2015 and it comes with documentation, training support, scalability options and support for different platforms. In addition, TensorFlow is associated with flexible, comprehensive community resources, libraries, frameworks and tools that facilitate developing and deploying machine learning solutions [53]. Keras [54] is a high-level neural network library that runs on top of TensorFlow. Further,Keras supports building high-level API to be used for easily building and training models. Keras is a built-in Python. Keras is an open-source software framework that provides a Python interface for designing and developing artificial neural networks.

On top of TensorFlow and Keras, a sequential machine-learning model was developed and tuned to produce optimum results. The developed model has three layers, consecutively 56 and 112 nodes in the first and second hidden layers and one node in the output layer. The activation function for all three layers is linear. The loss is calculated using the mean squared logarithm error while using the Adam optimizer.

V. Development of VR Environment

According to the 3D graphics rendering pipeline and commonly used game engine architectures, this type of 3D drone simulation environment consists of a real-time computational drone motion prediction module, configuration module (Configure environment, drone drill, etc.), The visual rendering engine, sound generation engine and seamless display system/head-mounted display [55]. Real-time motion prediction carries out by the proposed machine learning model which considers user interactions and defined environmental conditions to predict real-time drone motions. The visual rendering engine considers the predicted state of the drone-requested view of the virtual environment (first-person view, third-person view, etc.) to generate the relevant scenery. Finally, the user can see the generated visual through a seamless display system or head-mounted display. Integration of a sound generation engine brings more realism to the solution. A high-level structure of the virtual environment is illustrated in Fig. 2.

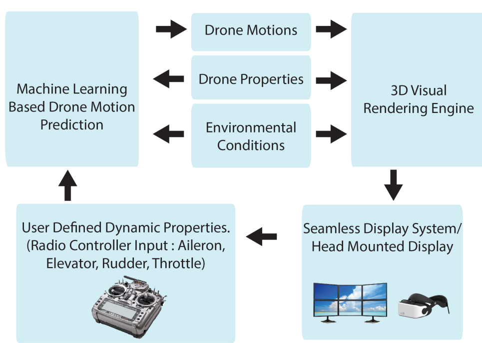

Fig. 2. The high level structure of the 3D virtual environment

There are many open-source, free and commercial frameworks, libraries and engines to develop 3D VR environments [56], [57], [58], [59]. Mairaj, et al. carried out comprehensive literature revive related to the design and development of drone simulators and analyzed available frameworks, libraries and engines [60]. Based on the Mairaj, et al. review, authors' own experience gained during the last ten years [61], GitHub Star rating and other reviews, Microsoft Aerial Informatics and Robotics Simulation (AirSim) [62], [63] open-source robotics simulation platform was selected to design and develop the proposed 3D virtual environment to simulate drone dynamics.

AirSim is developed for AI research to experiment with deep learning, computer vision and reinforcement learning algorithms for autonomous vehicles. However, AirSim provides APIs to retrieve data and control vehicles in a platform-independent way. Hence, several modules of AirSim were decoupled and several API facilities, such as retrieve data and control vehicles were slightly modified to develop the proposed 3D drone simulation environment [62]. Fig. 3 presents the high-level architecture of the modified Microsoft AirSim simulation platform which was used to develop the 3D drone simulation environment.

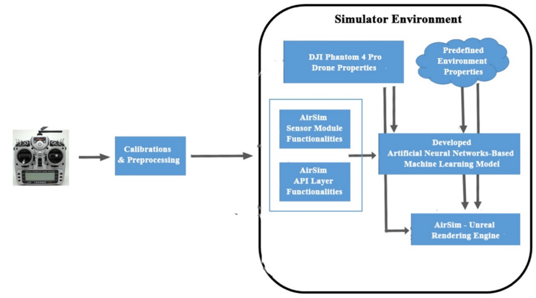

Fig. 3. High-level architecture of the modified microsoft airsim simulation platform

AirSim simulation platform was configured with an Unreal environment because the Unreal Marketplace has several environments available that can be used to generate realistic 3D scenaries easily. As illustrated in Fig. 3, AirSim physics engine was replaced with a developed machine learning-based drone dynamic model. The selected functionalities of the AirSim sensor module and AirSim API layer were reused to connect the real physical radio controller and virtual environment. However, a real Fr-Sky Taranis radio transmitter & receiver was used to capture drone pilot inputs and feed input stream to develop a machine learning-based drone dynamic model in the simulated environment. Hence, a separate calibration/preprocessing module was developed to align/map Fr-Sky Taranis radio transmitter inputs and DJI Phantom 4 Pro drone radio controller inputs. Fig. 4 presents the Fr-Sky Taranis radio transmitter/radio controller, which can be configured to align with DJI Phantom 4 Pro drone radio controller.

Fig. 4. Fr-sky taranis radio transmitter/radio controller

Fig. 5 presents the design and development of the proposed machine learning-based drone dynamic simulation environment. It uses an Intel Core i7 CPU with 3.6 Hz, GTX 1070 graphic card and Ram 32 GB to run the entire solution-inducing Unreal rendering engine and machine learning model. 32-inch screen connected to visualize the rendered image stream, and it makes continuous sensation for the users (Drone pilots) who interact with the virtual environment via Fr-Sky Taranis radio controller. Depending on the user's/drone pilot's preference instated of the screen, a head-mounted display such as Oculus Quest-2 can be connected with the solution.

VI. Experimental Results and Validation

The entire simulated environment can be validated under different criteria. As discussed before, simulation represents

Fig. 5. Developed machine learning based drone dynamic simulation environment.

a real-world scenario with assumptions, limitations and simplifications. Comparison of the simulated results against its actual real-world scenario is one of the validation methods employed in this type of research and development work. This can be done by considering comparisons such as positions and orientations with the timestamps. The user test is another vital validation technique to validate the user's perception of the simulated environment. In this technique, the virtual environment is validated based on the observations of pilots with much drone experience.

Some of these validation techniques are quantitative, while others are qualitative. For example, quantitative validation methods can be used to validate activities such as the accuracy of the motion predictions and qualitative validation methods can be used to validate components such as user perception enhancement and ecological validity of the simulated environment.

The validation process of the proposed virtual environment was divided into the following four segments:

- Carry out short-term motion predictions with the developed machine learning model: In this approach, real-world scenarios were simulated in the simulated environment and investigate the accuracy of the predictions against the radio controller input variations.
- Carry out long-term predictions with simple possible scenarios and investigate the acceptability of the obtained results.
- Carry out long-term motion predictions: In this approach, real-world scenarios were simulated in the simulated environment and compare-contrast the predictions against the real-world scenario.
- User tests: Compare the real-world user perception and the user perception of the VR solution with drone pilots who have much experience with drones. This can be used to compare the numerical and user-perceived accuracy and ecological validity of the VR solution.

As mentioned in the related work, we analyzed previously carried out research work such as [13], [14], [15], [42], [43], [44]. The analysis shows that users' awareness of the location of a real-world outdoor scene or virtual-world outdoor scene consists of percentage-based errors ranging from 10% to 25%.

As motioned above, the developed machine learning model predicts the drone's position and orations against the radio controller inputs throttle, pitch, roll & yaw. Due to the assumptions and limitations, it works under negligible wind effects while drones perform forward and lateral movements. Under the simulation of short-term motion predictions, a single prediction was made by considering the drone's current position, oration and controller inputs throttle, pitch, roll & yaw. We considered several figures of eight shape drone trials and considered the drone's actual position, orientation and ratio controller inputs, then predicted the drone's position and orientation after 0.1 seconds. Then closely investigate the effect of the factors on prediction error. Fig. 6 illustrates the Actual positions of the drone and predicted positions of the drone.

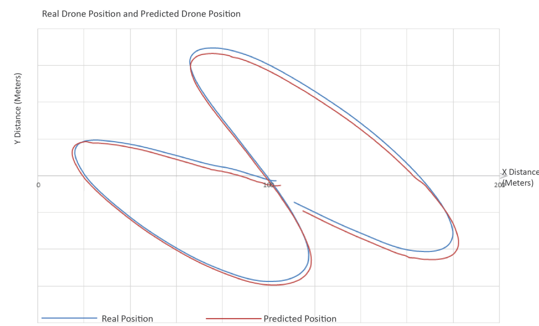

Fig. 6. Actual positions of the drone and predicted positions of the drone.

Fig. 7 illustrates the variation of the prediction error of the developed machine learning model against the time. It shows that the variation of the error is within 0.5 meters, and it is required to investigate the factors that affect the error.

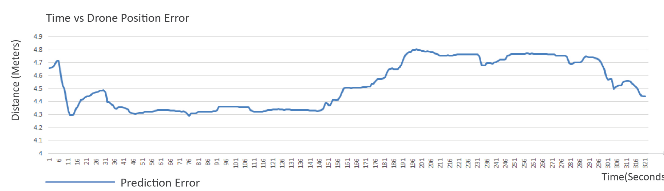

Fig. 7. Variation of the prediction error of the machine learning model against the time.

Table I presents the correlation between radio controller inputs (RcAileron, RcElevator, RcRudder, RcThrottle) and the machine learning model's prediction error. Further, it shows that the prediction error highly depends on the RcRudder-radio controller input.

Fig. 8 illustrates the acceleration error of the machine learning model against time. Fig. 9 depicts the magnified segment of Fig. 8 that illustrates the acceleration error of the machine learning model against time.

Fig. 8 and 9 show that most of the time, acceleration error is significantly less. Compared with the rudder variation pattern and acceleration error, it shows that the acceleration error rapidly increases with the rudder. However, smaller rudder variations in the developed machine-learning model

TABLE I. CORRELATION BETWEEN CONTROLLER INPUTS (RCAILERON, RCELEVATOR, RCRUDDER, RCTHROTTLE) AND PREDICTION ERROR

Correlation between Prediction Error and Radio Controller Inputs	Correlation Coefficient
Correlation between Prediction Error and RcAileron	0.003580433
Correlation between Prediction Error and RcElevator	0.369202804
Correlation between Prediction Error and RcRudder	0.989092821
Correlation between Prediction Error and RcThrottle	0.242975470

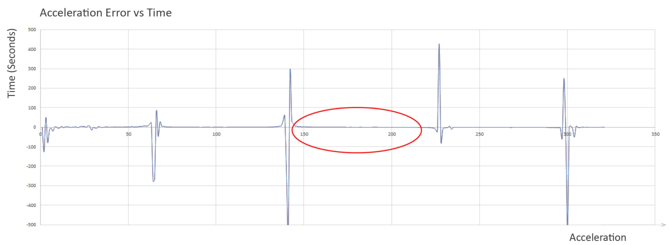

Fig. 8. Acceleration error of the machine learning model against time.

provide substantial accuracy. As previously mentioned, a real-world outdoor scene or virtual-world outdoor scene consists of percentage-based errors ranging from 10% to 25%. Hence, the developed model and its short-term motion prediction results provide encouraging results to perform long-term predictions and further validate the model.

A. Long-Term Drone Motion Predictions with Simple Possible Scenarios

As explained above (Subsection: Design and development of the VR solution), the Fr-Sky Taranis radio transmitter & receiver was connected to the solution and make enabled user interactions via an actual radio controller. Fig. 5, presents the designed and developed machine learning-based drone dynamic simulation environment. Three experienced pilots were exposed to the VR environment and asked to perform simple drills such as the figure eight type manoeuvre, manoeuvre along a straight line and circular manoeuvre. Their responses were assigned to a typical Likert scale with a five-point agreement scale related to simulation results produced by the solution. The five points of the Likert scale are strongly agreed, agree but no idea, disagree, and strongly disagree. The overall Likert scale results show that the three experienced pilots agreed with the results produced by the designed and developed simulation environment.

B. Long-Term Drone Motion Predictions and Validate against Real-World Scenario

Under the assumptions and limitations of the drone dynamic model, a simulation of long-term motion predictions was made. We consider several figures of eight-shape drone trials for this validation. We considered the drone's initial settled/stabled position, orientation in the outdoor environment and ratio controller inputs, then predicted the drone's position, and orientation continuously. Then closely investigate the deviations between the actual drone trial results and predicted results. Fig. 10 and Fig. 11 illustrate the actual trajectory of the drone and the predicted trajectory of the drone.

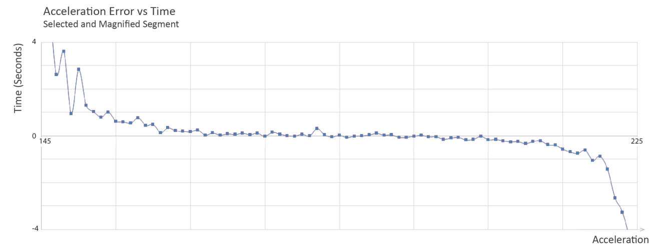

Fig. 9. Magnified segment of the previous graph that presents the acceleration error against time.

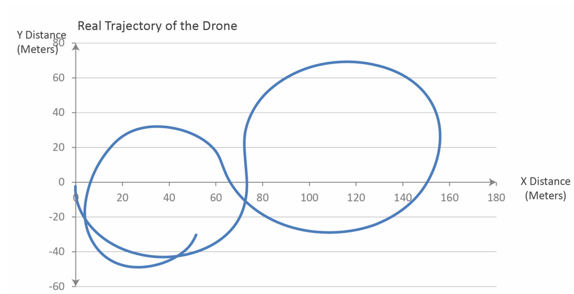

Fig. 10. Actual trajectory of the drone.

The above results show that the predictions are deviated/deviating with the time, and initial position predictions are closer to the actual position of the drone. As we discussed above, users perceive the position of an outdoor drone or users perceive the position of a drone in an outdoor VR environment is accoladed with a percentage-based error ranging from 10% to 25%. We consider the average of this error and define error merging for the predicted results. Fig. 12 illustrates the initial segment of the predicted drone trajectories and real drone trajectory with probable user perceive rejoin of the drone.

C. User Tests

The developed machine learning-based drone dynamic simulation environment needs to be validated to determine the immersive feeling "sense of being there" or "how users perceive" in the VR environment. The most common method of measuring this presence or "sense of being there" is to use questionnaires [64], [65]. Questionnaires give subjective measurement, and in most questionnaires, participants' responses to each question are assigned to a numerical scale [14]. Finally, the immersive feeling "sense of being there" in the VR solution can be reflected as a percentage. In the research validation phase, our primary focus is on subjective measurements. Hence, the user test was designed with a questionnaire. It targets three experienced drone pilots, including the drone pilot involved in the data gathering/recording pace, and the user test can be summarized as follows:

Simulate known conditions and record experienced drone pilots' responses. First, concerning each participant's response (feeling about the simulated scenario), qualitative properties of the simulated scenario will be assigned to a numerical scale (Likert scale) as follows [66]. Next, the quantitative properties of the simulated scenario will be directly recorded. Finally, the deviation from the expected value will be calculated. Under

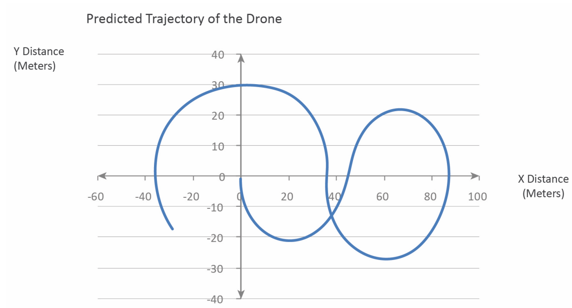

Fig. 11. Predicted trajectory of the drone

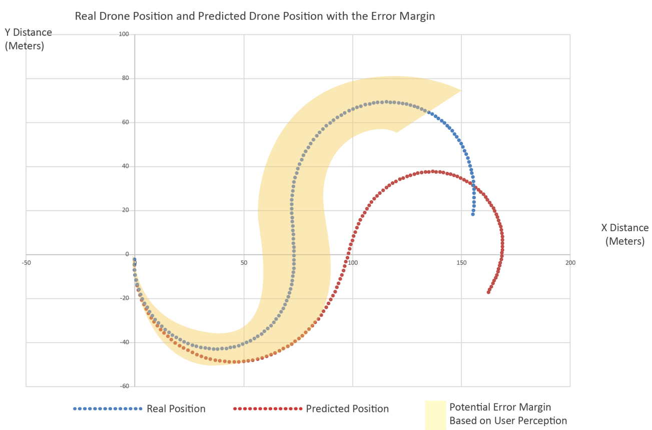

Fig. 12. Initial segment of the predicted drone trajectory and real drone trajectory with probable user perceive rejoin of the drone.

the simulated scenarios, many responses were recorded and selected questions/statements are given below:

- Approximated travel distance $\approx$ 300m-400m $\rightarrow$ What is the approximated travel distance?
- What is the current velocity?
- Asked to perform figure eight manoeuvre. What is the approximate size of the performed figure eight manoeuvre?
- Compared to the real-world figure eight manoeuvre with Phantom 4 Pro, the drones' response to the radio controller during that figure eight manoeuvre carried out in the simulated environment is realistic.
- Used radio controller: Ease of Use/ realism level compared to the real-world Phantom 4 Pro radio controller is it perfect/realistic?
- How many vehicles were there on the ground surface?
- Presence (the user's sense of immersion or "being within" the environment) is perfect.
- Three drone pilots were exposed virtual environment with large flat screen-based visualization. Do you recommend HMD or any other visualization method?

The actual and simulated scenarios' qualitative and quantitative properties were compared throughout this user test.

TABLE II. Selected Results of the Carried Out User Tests

Simulated Scenario	Pilot Response Numerical Indicator
Physical realism of the entire VR environment (Input Methods/ Controllers/Output methods/ Visualizations)	1.4 (from -2 to 2)
Realism of the Drone Maneuverability	1.0 (from -2 to 2)
Deviation of the size of the performed trial/covered area during the circular/ figure eight maneuvers	20%
Deviation of travel distance judgment	25%
Deviation of speed judgment	15%
Awareness and information gathering capability	1.4 (from -2 to 2)
Presence (immersion or "being within" the environment)	1.0 (from -2 to 2)

The main focus is to identify spatial awareness (The user's implicit knowledge of his position and orientation within the environment - during and after travel), information gathering (the user's ability to actively obtain information from the environment - during travel), and accuracy of the drone motion prediction against the real-world situation. Table II presents selected results of the carried out user tests.

VII. Conclusions and Future Work

The overall objective of the presented work is to propose a machine learning-based drone dynamic model and VR environment that can simulate existing drones without domain-specific knowledge and sophisticated laboratory infrastructure. Under selected circumstances, the proposed solution's accuracy and user-perceived accuracy were evaluated, and the authors were able to get promising results. The following conclusions and recommendations can be made based on the entire research.

- The proposed and developed machine learning model and its evaluation experiments were carried out by using Commodity-Off-The-Shelf hardware.
- If an accurate location tracker is available, the same procedure can be followed, and a similar machine-learning model can be developed for any existing drone.
- Short-term predictions of the proposed and developed machine learning model are within the user-perceived accuracy of both the real-world outer door scene and the virtual world outer door scene.
- The accuracy of the predictions mainly depends on the rudder variation. If the rate change of the rudder is more significant, then predictions of the proposed and developed machine learning model are less.
- Long-term predictions of the proposed and developed machine learning model are within the user-perceived accuracy of both the real-world outer door scene and the virtual world outer door scene for a limited period, and it gradually deviates with time.
- However, user test results show that the experience drone pilots agree with the simulated drone's physical realism and manoeuvrability. Moreover, there is a deviation between the user perceived position/speed of the simulated drone and the actual position/speed of the simulated drone.

- According to the evaluation results, early-stage predictions provided by the proposed and developed drone simulator are substantially accurate with rudder variations. Hence, drone piloting drills/missions/exercisers that require a short period (less than 25 seconds) for the entire activity can be simulated with substantial user perceive realism, behavioural realism and physical realism.

There is physical and behavioural realism in the proposed machine learning-based drone simulation environment. However, a wide range of further research work can be carried out to improve existing physical and behavioural realism. Some of the most critical and selected future research works are described below.

- Perform a wide range of drone manoeuvres and collect larger data sets that cover more drone dynamics and enhance behavioural realism by increasing the accuracy of the machine learning-based drone dynamic model.
- Design and develop a cylindrical or spherical display system, connect head-mounted display (HMD) and carry out experiments to enhance the physical realism and user perceive accuracy.

REFERENCES

[1] D. Sandaruwan, M. Jayasundara, N. Kodikara, and S. Pitigala, "Machine learning based approach to simulate drone dynamics related to figure of eight maneuvering pattern," *European Journal of Computer Science and Information Technology*, vol. 7, no. 5, pp. 16–25, 2019.

[2] K. Nonami, "Research and development of drone and roadmap to evolution," *Journal of Robotics and Mechatronics*, vol. 30, no. 3, pp. 322–336, 2018.

[3] J. F. Keane and S. S. Carr, "A brief history of early unmanned aircraft," *Johns Hopkins APL Technical Digest*, vol. 32, no. 3, pp. 558–571, 2013.

[4] J. D. Blom, "Unmanned aerial systems: A historical perspective," 2010, available online at: https://www.armyupress.army.mil/Portals/7/combat-studies-institute/csi-books/OP37.pdf.

[5] A. Intwala and Y. Parikh, "A review on vertical take off and landing (vtol) vehicles," *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 2, no. 2, pp. 187–191, 2015.

[6] Parrot Drone SAS, "Parrot ar.drone 2.0 elite edition," 2018, available online at: https://www.parrot.com/global/drones/parrot-ardrone-20-elite-edition.

[7] DJI Enterprise, "Drone solutions for a new generation of work," 2018, available online at: https://www.dji.com/phantom-4.

[8] 3D Robotics Inc, "Scale your drone operations with enterprise atlas," 2018, available online at: https://3dr.com.

[9] W. Hoffman, "Drones will need air traffic control, stanford and nasa will build it," inverse innovation," 2016, available online at: https://www.inverse.com/article/13951-drones-will-need-air-traffic-control-stanford-and-nasa-will-build-it.

[10] T. Kelly, "The booming demand for commercial drone pilots," 2017, available online at: https://www.theatlantic.com/technology/archive/2017/01/drone-pilot-school/515022/.

[11] Pilot Institute, "How much salary can a drone pilot expect?" 2020, available online at: https://pilotinstitute.com/drone-pilot-salary/.

[12] Det Norske Veritas (DNV), "Standard for certification no. 2.14 maritime simulator systems," 2011.

[13] D. Sandaruwan, N. Kodikara, C. Keppitiyagama, R. Rosa, M. Jayawardena, and P. Samarasinghe, "User perception of the physical & behavioral realism of a maritime virtual reality environment," in *2012 UKSim 14th international conference on computer modelling and simulation*. IEEE, 2012, pp. 172–178.

[14] D. Sandaruwan, "Computer based ship model to predict real-time ship motion characteristics for perception enhanced virtual environment," 2015.

[15] D. Sandaruwan, N. Kodikara, C. Keppitiyagama, R. Rosa, K. Dias, R. Senadheera, and K. Manamperi, "Low cost immersive vr solutions for serious gaming," in *Algorithmic and Architectural Gaming Design: Implementation and Development*. IGI Global, 2012, pp. 407–429.

[16] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, "Dynamics modelling and linear control of quadcopter," in *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE, 2016, pp. 498–503.

[17] C. Caceres, D. Amaya, and J. M. Rosario, "Simulation, model and control of a quadcopter ar drone 2.0," *dynamics*, vol. 8, p. 10, 2016.

[18] F. Sabatino, "Quadrotor control: modeling, nonlinearcontrol design, and simulation," 2015.

[19] C. MASSÉ, O. GOUGEON, D.-T. NGUYEN, and D. SAUSSIÉ, "Modeling and control of a quadcopter flying in a wind field: A comparison between lqr and structured h∞ control techniques," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 1408–1417.

[20] Spark Pilots, "Dji assistant 2 ver 1.1.2," 2018, available online at: https://sparkpilots.com/threads/dji-assistant-2-ver-1-1-2.11976/.

[21] DJI, "Dji download center," 2018, available online at: https://www.dji.com/downloads/softwares/assistant-dji-2.

[22] Horizon Hobby, LLC, "Realflight 8 horizon hobby edition," 2018, available online at: https://www.realflight.com.

[23] DroneSim Pro, "Dronesim pro drone simulator," 2019, available online at: https://www.dronesimpro.com/.

[24] Immersion RC, "Liftoff – the drone race simulator," 2019, available online at: https://www.immersionrc.com/fpv-products/liftoff-drone-race-simulator.

[25] Heli-X, "Professional r/c flight simulation," 2015, available online at: https://www.heli-x.info/cms/.

[26] K. Amer, M. Samy, M. Shaker, and M. ElHelw, "Deep convolutional neural network based autonomous drone navigation," in *Thirteenth International Conference on Machine Vision*, vol. 11605. SPIE, 2021, pp. 16–24.

[27] P. T. Jardine, S. Givigi, and S. Yousefi, "Parameter tuning for prediction-based quadcopter trajectory planning using learning automata," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2341–2346, 2017.

[28] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield, "Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4241–4247.

[29] P.-H. Chen and C.-Y. Lee, "Uavnet: an efficient obstacel detection model for uav with autonomous flight," in *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*. IEEE, 2018, pp. 217–220.

[30] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[31] R. P. Padhy, S. Verma, S. Ahmad, S. K. Choudhury, and P. K. Sa, "Deep neural network for autonomous uav navigation in indoor corridor environments," *Procedia computer science*, vol. 133, pp. 643–650, 2018.

[32] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," *arXiv preprint arXiv:1611.04201*, 2016.

[33] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[34] O. Çakir and T. Yüksel, "Neural network control for quadrotors," *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 31, no. 1, p. 191–200, May 2017.

[35] D. Jeong, M. Baek, and S.-S. Lee, "Long-term prediction of vehicle trajectory based on a deep neural network," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2017, pp. 725–727.

[36] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[37] J. Brownlee, "A gentle introduction to the rectified linear unit (relu)," 2019, available online at: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

[38] R. D. Jackson, M. Jump, and P. L. Green, "Predicting on-axis rotorcraft dynamic responses using machine learning techniques," *Journal of the American Helicopter Society*, vol. 65, no. 3, pp. 1–12, 2020.

[39] J. Requeima, W. Tebbutt, W. Bruinsma, and R. E. Turner, "The gaussian process autoregressive regression model (gpar)," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1860–1869.

[40] A. Punjani and P. Abbeel, "Machine learning for helicopter dynamics models," Ph.D. dissertation, University of California, Berkeley, 2014.

[41] Stanford University, "The stanford university autonomous helicopter," 2010, available online at: http://heli.stanford.edu.

[42] J. Zhang, X. Yang, Z. Jin, and L. Li, "Distance estimation in virtual reality is affected by both the virtual and the real-world environments," *i-Perception*, vol. 12, no. 3, p. 20416695211023956, 2021.

[43] C. J. Ziemer, J. M. Plumert, J. F. Cremer, and J. K. Kearney, "Estimating distance in real and virtual environments: Does order make a difference?" *Attention, Perception, & Psychophysics*, vol. 71, no. 5, pp. 1095–1106, 2009.

[44] R. M. Baños, C. Botella, A. Garcia-Palacios, H. Villa, C. Perpiñá, and M. Alcaniz, "Presence and reality judgment in virtual environments: a unitary construct?" *CyberPsychology & Behavior*, vol. 3, no. 3, pp. 327–335, 2000.

[45] DJI Technology company, "Phantom 4 specs," 2016, available online at: https://www.dji.com/phantom-4/info.

[46] DJI Technology Company, "Phantom 4 pro specs," 2016, available online at: https://www.dji.com/phantom-4-pro/info.

[47] J. Vrana and R. Singh, "Nde 4.0—a design thinking perspective," *Journal of nondestructive evaluation*, vol. 40, no. 1, pp. 1–24, 2021.

[48] Priya Pedamkar, "Differences between machine learning vs neural network," available online at: https://www.educba.com/machine-learning-vs-neural-network/.

[49] SAS Institute Inc, "Artificial neural networks: What they are & why they matter," 2022, available online at: https://www.sas.com/en_us/insights/analytics/neural-networks.html.

[50] R. Bhatia, "When not to use neural networks," 2018, available online at: https://medium.datadriveninvestor.com/when-not-to-use-neural-networks-89fb50622429.

[51] M. Kuhn, K. Johnson *et al.*, *Applied predictive modeling*. Springer, 2013, vol. 26.

[52] P. Sharma, "Improving neural networks – hyperparameter tuning, regularization, and more (deeplearning.ai course #2)," 2018, available online at: https://www.analyticsvidhya.com/blog/2018/11/neural-networks-hyperparameter-tuning-regularization-deeplearning/.

[53] Tensorflow.org, "Tensorflow," 2015, available online at: https://github.com/tensorflow/tensorflow.

[54] Keras, "Keras api reference," (n.d.), available online at: https://keras.io/api.

[55] R. S. Ferguson, *The Principles of Quantum Mechanics*. CRC Press, 2014.

[56] Epic Games, "The world's most open and advanced real-time 3d creation tool," (n.d.), available online at: https://www.unrealengine.com/en-US/.

[57] OGRE, "Open source 3d graphics engine," (n.d.), available online at: https://www.ogre3d.org.

[58] Unity, "Unity real-time development platform," (n.d.), available online at: https://unity.com.

[59] VDrift, "Vdrift," (n.d.), available online at: http://vdrift.net.

[60] A. Mairaj, A. I. Baba, and A. Y. Javaid, "Application specific drone simulators: Recent advances and challenges," *Simulation Modelling Practice and Theory*, vol. 94, pp. 100–117, 2019.

[61] D. Sandaruwan, "Computer based ship model to predict real-time ship motion," Ph.D. dissertation, University of Colombo, Colombo, 2015.

[62] Microsoft, "Airsim," 2017, available online at: https://microsoft.github.io/AirSim/.

[63] Microsoft Research, "Aerial informatics and robotics platform," (n.d.), available online at: https://www.microsoft.com/en-us/research/project/aerial-informatics-robotics-platform/.

[64] B. There, "Concepts, effects and measurements of user presence in synthetic environment," 2003.

[65] T. Tsiatsos, K. Andreas, and A. Pomportsis, "Evaluation framework for collaborative educational virtual environments," *Journal of Educational Technology & Society*, vol. 13, no. 2, pp. 65–77, 2010.

[66] S. McLeod, "Likert scale definition, examples and analysis," 2019, available online at: https://www.simplypsychology.org/likert-scale.html.