Information-based Supervised Learning of In-proximity Effects for 3D Distance Estimation and Collision Avoidance

Jacob M. Anderson and Kam K. Leang†

***Abstract*—In-proximity effects (IPE) in 3D, specifically in-ground, in-ceiling, and in-wall effects, experienced by a rotary-wing aerial robot as it flies near obstacles are leveraged for obstacle distance estimation and collision-free motion control. Onboard motor commands and inertial measurement unit (IMU) signals are processed to enable the robot to essentially "feel" the presence of nearby obstacles through aerodynamic interactions. The physics of IPE, along with Shannon information, are used to tailor the input space and train a deep neural network (DNN) to estimate the distance to ground, ceiling, and wall features. Simulation and physical experimental results demonstrate reliable and robust obstacle detection and collision avoidance with a median distance estimation accuracy of 93.35%, 89.22%, and 90.67% for ground, ceiling, and wall, respectively. This new form of "sensing" is useful in environments with fog, smoke, dust, rain, or snow, where traditional proximity sensors and vision-based systems struggle to detect obstacles and determine distance.**

***Index Terms*—Aerial Systems: Perception and Autonomy, Deep Learning Methods, Motion and Path Planning**

I. Introduction

COLLISION-FREE flight, within close proximity to and around obstacles or through confined spaces, requires accurate obstacle distance sensing and precise motion control. This is especially true for rotary-wing aerial robots such as quadcopters, where complex aerodynamic interactions between the rotor wake and nearby obstacles induce in-proximity effects such as in-ground, in-ceiling, and in-wall effects [1], [2]. These near-obstacle effects can lead to an increased rate of collisions. On-board sensors such as camera, sonar, and LiDAR are often used to detect the presence of obstructions to keep the robot at a safe distance for collision-free navigation [3]. However, such sensors are bulky, heavy, and their weight limits flight time and performance. Additionally, detecting and avoiding obstacles is more difficult in challenging conditions that contain smoke, dust, fog, rain, and snow. These conditions limit and hinder the performance of traditional proximity sensors, affecting the ability of the aerial robot to navigate safely [4].

We introduce a new method for estimating obstacle distance that leverages the physics of in-proximity effects (IPE), specifically in-ground, in-ceiling, and in-wall effects, along with supervised learning and information theory. Our method only uses motor commands and on-board inertial measurement unit (IMU) signals to enable a rotary-wing aerial robot to estimate the distances of nearby obstacles for collision-free navigation (see Fig. 1). Using the underlying IPE physics, it is shown that a set of deep neural networks (DNNs) can process the on-board signals to estimate the distance to ground, ceiling, and wall for motion control. In fact, as the robot approaches these features, changes in airflow above, below, or between the rotors lead to local changes in air pressure as illustrated in Fig. 2(a-c). Thus, the aerial robot experiences changes in thrusts and moments that affect its behavior. The experimental results in Fig. 2(d) and (e) show that the required motor power decreases as the robot approaches a ground and ceiling feature, respectively. Similarly, the difference in motor power between the rotors closest to and farthest from a wall feature also changes with wall distance (see Fig. 2(f)). We also show that the Shannon information of the vehicle behavior has decreased sensitivity to expected behavior variation due to aerodynamic stochasticity and increased sensitivity to direct behavioral changes due to IPEs. By including these information terms in the DNN input space, the distance estimation accuracy is greatly improved.

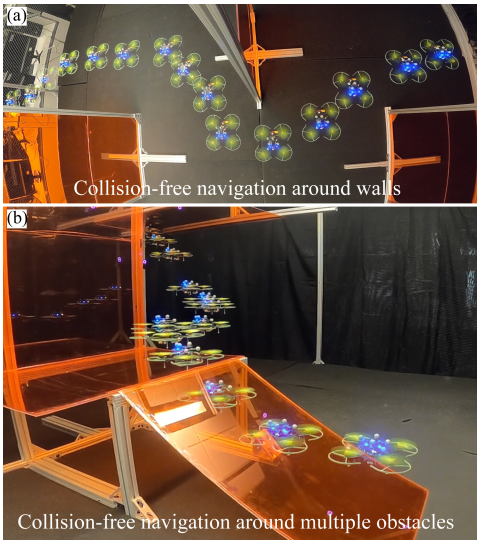

Fig. 1: Quadcopter robot navigating through maze-like environments by estimating obstacle distance via IPE: (a) zigzagging around three walls and (b) flying up a ramp, detecting and flying around a wall, and then avoiding a small ceiling feature.

Figure 3 illustrates our concept, where on-board motor commands and IMU signals are processed by an information-based DNN to estimate obstacle distance for collision avoidance. Our results show accuracies of 93.35%, 89.22%, and 90.67% to estimate the distance to ground, ceiling and wall, respectively. The inclusion of the information terms improves not only the repeatability of the DNN training processes but also the ability to generalize to situations not accounted for within the training sets. Using our method, a quadcopter robot is able to navigate its way through maze-like environments. For example, Fig. 1(a) shows the robot zig-zagging through three vertical walls without collisions. Figure 1(b) shows the robot flying up a ramp, detecting and going around a wall, and then

Manuscript received: October 8, 2025; Revised: January 3, 2026; Accepted: January 18, 2026. This paper was recommended for publication by Editor Aniket Bera upon evaluation of the Associate Editor and Reviewers comments. Authors are with the Dynamic Autonomous Robotics (DARC) Lab, Dept. of Mechanical Eng., Univ. of Utah. They are also members of the Univ. of Utah Robotics Center. †Corresponding author, e-mail: kam.k.leang@utah.edu. Digital Object Identifier (DOI): see top of this page.

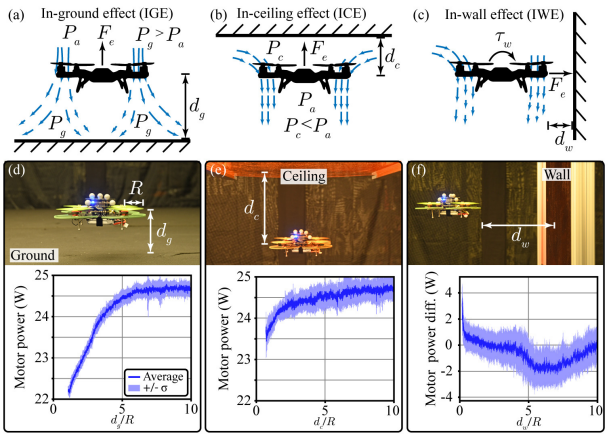

Fig. 2: (a)-(c) In-ground, in-ceiling, and in-wall effects. Differences in pressure (P_x relative to the atmospheric pressure P_a) induce forces F_e and moments τ_w that impact vehicle behavior. (d)-(f) IPEs observed through measured motor power as a function of the normalized distance (d_x/R) between the rotor (with radius R) and obstacle.

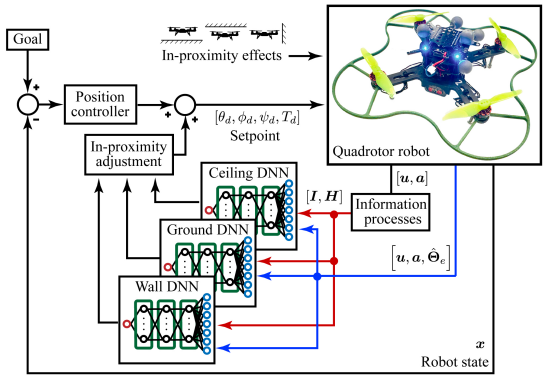

Fig. 3: Collision avoidance control scheme using estimated distance of detected obstacles from IPE-informed DNN.

avoiding a small ceiling feature. Because motor commands and IMU signals are used, our method can be readily applied to nearly any rotary-wing platform just through a software update, regardless of the robot configuration or sensor payload capabilities. No new hardware or mechanical design is needed.

While many researchers have focused on modeling IPEs [5]–[7] and designing controllers to account for IPE-induced disturbances [8], [9], there is limited effort on leveraging IPEs for obstacle distance estimation and collision avoidance. In-ground effects (IGEs) were first used to estimate relative ground proximity through changes in the motor inputs required to maintain a vehicle's position [10]. Empirical relations between motor input and vehicle position were used for efficient trajectory planning [11]. An unscented Kalman filter and a classifier were used to estimate in-wall-effect induced forces and torques [12]. Ground obstacles were avoided with reinforcement learning [13]. Finally, wall obstacles were avoided using an extended Kalman filter (EKF) and a ducted-rotor shroud to amplify the in-wall effects [14].

Table I compares the capabilities of our approach to prior works [10]–[14], where we can handle all three IPE effects (ground, ceiling, and wall). Even though faster performance has been achieved for detecting walls [14] through specialized

TABLE I: Comparison of capabilities of prior work and proposed approach for ground (G), ceiling (C), and wall (W).

	Prox. Detect.	**Distance Est.**	**Collision Avoid.**
Powers et al. [10]	G	N/A	N/A
van Dam, van Kampen [13]	G	N/A	G
Gao et al. [11]	G/C	G/C	G
McKinnon, Schoellig [12]	W	N/A	N/A
Ding et al. [14]	W	W	W
Our Method	**G/C/W**	**G/C/W**	**G/C/W**

rotor shroud designs, our work strictly focuses on processing onboard signals to estimate IPE in multiple directions. Thus, our work advances the state-of-the-art by the following novelty and contributions: (1) leveraging IPE and supervised learning to estimate the proximity of obstacles within a 3D environment; (2) incorporating Shannon information in the DNN input space to improve estimation accuracy; and (3) demonstrating collision-free navigation that utilizes the estimated obstacle distance as feedback. Our approach for obstacle detection is versatile and can be easily adapted to various multirotor configurations to augment sensing capabilities in challenging conditions and improve the robot's overall performance.

II. Ground, Ceiling, and Wall In-proximity Effects

The study of in-ground effects in helicopters dates back to the 1950s [15]. More recent research has explored in-ceiling effects (ICE) [11] and in-wall effects (IWE) [12], [14] for multirotor robots. For IGE, as shown in Fig. 2(d), up to 10% less motor power is required to hover when the vehicle approaches ground [5], [8], [16]. This is due to the rotor downwash being obstructed and forced to disperse radially against the ground surface [2]. Thus, the vehicle is '*pushed*' away as it approaches the ground surface. Similarly for ICE [7], [17], less motor power is required to hover near a ceiling feature (see Fig. 2(e)). Thus, ICE '*sucks*' the vehicle into the ceiling and it is observed to be approximately 50% less than IGE for the same vehicle. Both are often modeled by [15]

$$\frac{T(z)}{T_\infty} = \left[1 - \left(\frac{R}{4z}\right)^2\right]^{-1}, \tag{1}$$

where T_∞ is the out-of-proximity effect thrust, R is the rotor radius, and z is the ground-to-vehicle distance. Recently, models have been proposed to account for different multirotor configurations and rotor-blade geometry, for example for IGE [5]

$$\frac{T(z)}{T_\infty} = C_a e^{-C_b z/R} + 1, \tag{2}$$

while for ICEs, the thrust ratio is shown to be [18]

$$\frac{T(z)}{T_\infty} = \frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{f_t(\theta)}{8(z/R)^2}}. \tag{3}$$

Finally, for IWE, the experimental results in Fig. 2(f) show that the motor-power difference increases as the wall distance d_w decreases, where the percent change between the in-proximity and out-of-proximity zones is approximately 7%. For ducted rotors, it has been shown that IWE increases the

magnitude of a horizontal force [14], [19] and torque acting on the vehicle [6]. Thus, the vehicle is *pulled* into the wall and this behavior can be modeled by an external force F_e and a torque τ_w into the wall (see Fig. 2(c)). From momentum theory and conservation of energy, horizontal thrust is given by [14]

$$F_h = \gamma_h(d_w/R)(\sqrt{2\rho A}P_A)^{\frac{2}{3}}, \tag{4}$$

where ρ is the air density, A is the area of the rotor disk, P_A is the aerodynamic power of the spinning rotor, and $\gamma_h(\cdot)$ is empirically fit and varies with the distance to a wall. It is noted that the torque on the vehicle, which is experimentally shown to develop due to IWEs, is not accounted for in existing IWE models. The horizontal thrust, however, is shown to increase as d_w decreases. In Fig. 2(f), the dip in the response over $d_w = [5, 10]$ is believed to be caused by the combined wall-ground interaction pushing the robot away from the wall surface, therefore requiring a roll towards the wall to maintain position. More studies are needed to test this hypothesis.

The measured results in Fig. 2(d)-(f) show that motor commands contains information about IPEs. These findings, along with the fact that data must be gathered to facilitate empirical fittings for existing IPE models, motivate our approach to utilize on-board signals, combined with supervised learning, to estimate the distance to obstacles for collision avoidance.

III. Physics-informed Information-based DNN for Obstacle Distance Estimation

A. Deep Neural Network Structure

A deep neural network is used to estimate the: (1) intensity of the IPE and (2) distance to the obstacle surfaces, $\hat{d}_o$. The chosen DNN is a simple fully-connected feedforward network using rectified linear units (ReLU) activation (Fig. 3). Each network consists of only 3 hidden layers with 24 neurons. During training, the mean-squared-error loss is evaluated, and dropout stochastic regularization is applied.

Motivated by the underlying physics of IPEs and the results shown in Fig. 2, the primary inputs to the DNNs are the motor powers that relate to thrust and torque on the vehicle, *i.e.*,

$$\boldsymbol{\alpha}[k] = \begin{cases} [u_1[k], u_2[k], u_3[k], u_4[k]] & \text{(ground/ceiling)} \\ [\tau_{1,2}[k], \tau_{3,4}[k], \tau_{1,4}[k], \tau_{2,3}[k]] & \text{(wall)} \end{cases}. \tag{5}$$

B. DNN Information-Based Inputs to Handle Stochasticity

Due to the turbulent nature of the aerodynamic interactions, there is a high level of stochasticity associated with the DNN inputs, $\boldsymbol{\alpha}$. To enable the DNNs to better handle the stochasticity, the DNN input space will include the information of the inputs. In order to justify, in a theoretical sense, why these inputs are useful for the stochastic process, we first define the Shannon information of a variable β as [20]

$$I(\beta) = -\log(p(\beta)), \tag{6}$$

where the marginal probability of β, assuming flight with no obstacles, is determined relative to a normal distribution

$$p(\beta) = \mathcal{N}(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(\beta-\mu)^2}{2\sigma^2}}. \tag{7}$$

The distribution $p(\beta)$ is formed by fitting data gathered during test flights, including both cases where the robot hovers at a set position and where a trajectory is followed at speeds up to 0.0175 m/s without the presence of obstacles. Next, we assume that the base measured input β varies from its expected mean value by Δ; thus, the change in the variable can be written as

$$\Delta\beta = |(\mu + \Delta) - \mu| = \Delta. \tag{8}$$

However, the change in the information of the variable with respect to Δ, using the marginal probability Eq. (7), is

$$\Delta I(\beta) = |I(\mu + \Delta) - I(\mu)| = \frac{\Delta^2}{2\sigma^2}. \tag{9}$$

Thus, for a Gaussian baseline, the change in information is proportional to the squared deviation from the mean, Δ, normalized by the squared standard deviation, σ. This makes it less sensitive to small fluctuations caused by the stochasticity of the aerodynamic interactions compared to the standard inputs (Eq. (8)), while amplifying statistically unlikely deviations caused by direct IPEs. These information-based input features are deterministic transforms of the measured signals, and therefore, due to the data processing inequality, do not increase the information content available to the DNN. Instead, they introduce an additional coordinate that expresses each measurement as a normalized deviation from a nominal, far-from-obstacle, operating distribution. Since IPEs induce shifts in the distribution relative to the nominal hover behavior (see Fig. 2), this representation provides better-conditioned inputs for a finite ReLU network to learn the mapping to obstacle distance, improving the robustness to stochastic variability.

To provide a notion of temporal change to the networks, the expectation of the information, $\tilde{H}$, over a time window w_h,

$$\tilde{H}(\beta[k]) = -\sum_{j=k-w_h}^{k} p(\beta[j]) \log(p(\beta[j])), \tag{10}$$

is used as an additional DNN input. This new *pseudo entropy* quantity is analogous to Shannon entropy except that not every possible outcome of the event is captured by the summation. Both the information I and the pseudo entropy $\tilde{H}$ provide better conditioned inputs relative to the expected system stochasticity. To the authors' best knowledge, information-based terms have not been used to assist in identifying IPEs but are important as aerodynamic interactions are highly stochastic processes.

IV. Simulation Study: Estimating Ceiling Distance

To initially understand the performance of the DNN distance estimator subject to different inputs (motor power signals and the information terms), a simulation and physical experiment were performed for in-ceiling effect. The simulation uses a basic six-DOF rigid body model of a quadrotor [21], with a mass of 1.25 kg, 0.265-m arm length, and inertial parameters (I_{xx}, I_{yy}, I_{zz}) = (0.0232, 0.0232, 0.0468) kg·m^2. The ICE behavior is modeled by Eq. (1), which has been shown to accurately model ICE for multirotor systems [17]. Figure 4(a) shows the simulation setup, where the quadrotor moves vertically between normalized distances $d_c/R \in [1.2, 6.30]$ from a ceiling obstacle, following a sinusoidal trajectory over a 5-minute period (see

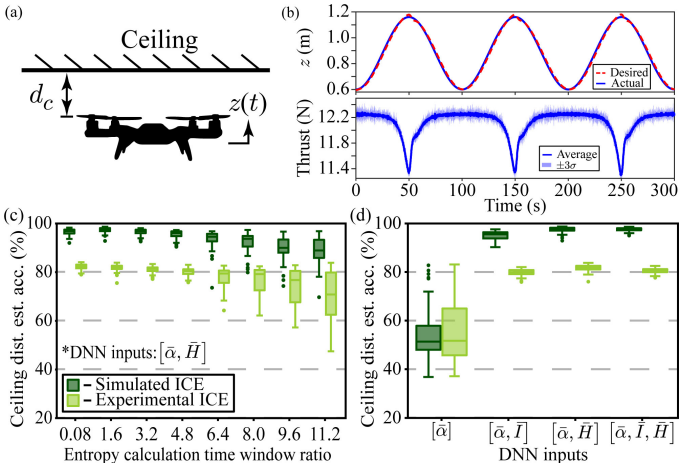

Fig. 4: (a) Simulation setup for training/estimating ceiling distance. (b) Training trajectory and input. (c) DNN accuracy estimates: (c) for thrust and thrust pseudo entropy, with different time windows, and (d) trained with different combinations of thrust, Shannon information, and pseudo entropy inputs.

Fig. 4(b)). Twenty simulation trials were performed with flight data split 70-30 between training and test sets for each DNN.

The results for the distance estimation accuracy versus entropy time windows are shown in Fig. 4(c). The window lengths are between 2 to 280 measurements, normalized by the robot's vertical translational time constant. For each window length, 200 DNNs were trained. The average estimated ceiling distance over the test set is calculated by

$$\bar{d}_a = \frac{1}{k_f}\sum_{k=0}^{k_f}\left(1-\left[|d_o[k]-\hat{d}_o[k]|/d_{max}\right]\right), \tag{11}$$

where d_o and $\hat{d}_o$ are the true and estimated distance to an obstacle, respectively, d_{max} is an upper bound on the DNN prediction range based on Fig. 2(e), and k_f is the final trial timestep. The DNNs using a pseudo entropy window length of 40 measurements (a time window ratio of 1.6) resulted in the highest median accuracy and the smallest variability between trained DNNs. This window range will be used for any network that uses the pseudo entropy as an input. When the pseudo-entropy window deviated from 40 measurements, the median accuracy decreased and the variability within the configuration's estimation accuracy increased greatly. This decrease in performance is believed to be due to the information content continuing to be averaged down as the pseudo entropy calculation window increases, leading to the diminished ability to identify IPE from the signal as it is increasingly filtered.

Next, the DNNs were trained using the inputs: (1) $[\bar{\alpha}]$, (2) $[\bar{\alpha}, \bar{I}]$, (3) $[\bar{\alpha}, \bar{H}]$, and (4) $[\bar{\alpha}, \bar{I}, \bar{H}]$, where each input is averaged across all motors. The results of the average estimated ceiling distances are shown in Fig. 4(d), where 200 DNNs were trained for each configuration. The DNNs using only the average motor power achieved a median accuracy of 51.32% with an interquartile range (IQR) of 10.06%. However, the DNNs with information-based inputs performed better, with an average median accuracy improvement of 45.67%. The IQR, on average, reduces to 15.6% of the IQR when only motor power is used. This result illustrates the benefit of using information-theoretic

TABLE II: ε_{min} **results from ASO testing on DNNs using different inputs for ceiling distance estimation.**

Simulated/Experimental ICE

DNN inputs	$[\bar{\alpha}]$	$[\bar{\alpha}, I]$	$[\bar{\alpha}, \tilde{H}]$	$[\bar{\alpha}, I, \tilde{H}]$
$[\bar{\alpha}]$	-	0.99/1.00	0.99/0.99	0.99/1.00
$[\bar{\alpha}, I]$	0.01/0.00	-	1.00/0.89	0.99/1.00
$[\bar{\alpha}, \tilde{H}]$	0.01/0.02	0.01/0.52	-	1.00/0.49
$[\bar{\alpha}, I, \tilde{H}]$	0.01/0.01	0.02/0.67	0.52/1.00	-

transforms of the vehicle behavior as additional inputs to the DNN. Even though the information terms are direct transforms of the motor power, they encode a sense of the expected vehicle behavior, process stochasticity, and have increased sensitivity to large IPE-induced changes. This supports the expected outcome consistent with the previously derived theoretical justification (Eq. (9)) for using information-based inputs. The effects of stochasticity are reduced on the measured vehicle behavior and the supervised learning, and thus the DNN is able to more accurately identify obstacle proximity.

To determine whether the results with information terms were statistically significant, the almost stochastic order (ASO) test [22] is used. The ASO test calculates the ratio, $\varepsilon_{W_2} \in [0,1]$, to which the stochastic order between two cumulative distribution functions (CDFs) is violated using

$$\varepsilon_{W_2}(D_1, D_2) = \frac{\int_{\mathscr{V}_\chi}(D_1^{-1}(t) - D_2^{-1}(t))^2 dt}{(W_2(D_1, D_2))^2}. \tag{12}$$

The constants D_1 and D_2 represent CDFs of the average estimation accuracies attained using two different DNN architectures, $\mathscr{V}_\chi$ is the violation set where the stochastic order is being violated between the two CDFs, and $W_2(D_1, D_2)$ is the univariate l_2-Wasserstein distance. An upper bound to the violation ratio can then be formulated, giving the minimal ε_{W_2} value to ensure stochastic dominance, as

$$\varepsilon_{min}(D_1, D_2, \alpha_t) = \varepsilon_{W_2}(D_1, D_2) - \sqrt{\frac{n_1+n_2}{n_1 n_2}}\hat{\sigma}_{1,2}\Phi^{-1}(\alpha_t).$$

Here, n_1 and n_2 represent the number of samples in the D_1 and D_2 CDFs, respectively, $\hat{\sigma}_{1,2}$ is the estimated variance term, α_t is the level of significance, and Φ^{-1} is an inverse normally distributed CDF. A DNN is determined to be stochastically dominant over another when the CDF of its accuracy scores for a test set compared to scores from another DNN architecture result in an $\varepsilon_{min} < \tau$. A significance level of $\alpha_t = 0.05$ is used (before adjusting for any pair-wise comparisons using the Bonferroni correction) and $\tau = 0.2$ is chosen based on a balance between Type I and Type II error rejection properties [22].

Table II shows the ASO results where each ε_{min} is the result of the DNN configuration left of the cell compared to the one listed above. It is noted that batches of 200 DNNs per input configuration were used because through a bootstrapped power analysis approach, batches of 200 DNNs were shown to result in a power greater than 0.8 and any further increase in the number of DNNs trained had a diminished effect on the

uncertainty reduction of the ASO test. The ASO results show that the information terms improve estimation accuracy with statistical significance. However, the prediction performance between the different DNNs that use information-based inputs showed no statistically significant differences. Thus, the pseudo entropy will be further evaluated as a DNN input, but the Shannon information input will be the primary information-based input for improving obstacle detection performance. It is pointed out that calculating information is computationally efficient and does not require a history of data.

To validate the results of the simulated ceiling obstacle distance estimation, a set of DNNs was trained on experimental data from trials using a physical quadrotor robot (Fig. 3). The accuracy distributions of DNNs trained on the physical data and using the same input configurations as the simulation case are shown in Fig. 4(c) and Fig. 4(d). When the experimental data is used, the prediction accuracy is consistently lower than that seen in the simulation results. However, the variation of the accuracy for both the different entropy calculation windows and for each different input configuration closely follows the trends seen in the simulation results. The lower accuracy is expected as the measured quadrotor parameters are less ideal and contain more variation than the simulated case. This leads to larger spreads in the inputs provided to the DNNs corresponding to each true obstacle distance seen during training, causing less certainty in the final distance estimation. Again, the DNN accuracy distribution results show that by including Shannon information or Shannon entropy terms as inputs, the prediction accuracy increases, and the training repeatability increases. The ASO results also show that the accuracy improvements gained are statistically significant.

These initial simulation and experimental results for the ICE case provide evidence that a DNN can estimate the distance to an obstacle based on IPEs observed only through the motor power inputs. Also, by including information-based terms, the accuracy of the DNN distance estimates and the repeatability of the training process are greatly improved. Next, we extend the DNN structure to include IMU and EKF signals to experimentally estimate ground, ceiling, and wall distances, and use this information to demonstrate collision avoidance.

V. Robot Platform and Test Environment

Physical experiments were conducted on a custom-designed, 3D-printed quadrotor robot with 3-inch propellers (see Fig. 3). The robot weighs 134 g, has footprint of 11.5 cm $\times$ 11.5 cm, and hovers for 14 min. with a 2s 1100 mAh battery. The Bitcraze Crazyflie 2.0 flight controller has an onboard IMU with an EKF to estimate the robot's state using motion capture position and attitude measurements [23]. Attitude and position PID control run at 500 Hz and 20 Hz, respectively.

Experiments are carried out in a 4.9 m (16 ft) $\times$ 3.7 m (12 ft) $\times$ 2.1 m (7 ft) flight volume equipped with 22 motion capture cameras (OptiTrack Flex 13 cameras) running at 120 Hz to record the robot's position and attitude. A ground station computer (GSC) runs Ubuntu 18.04 and the Robot Operating System (ROS) Melodic. The GSC controls the robot through a wireless radio link. All DNN tasks (such as training), including collision avoidance, are performed on the GSC through a ROS node architecture. Acrylic glass sheets were used as obstacles.

Recorded motor input and IMU signals are low-pass filtered to minimize noise. The filter is implemented using the forward-backward filtering algorithm [24]. Alternatively, when implementing trained DNNs for real-time obstacle distance estimation, a discrete infinite impulse response filter is used

$$y[k] = \alpha_d u[k] + (1-\alpha_d) y[k-1], \tag{13}$$

where y and u are the filtered and raw input signal, respectively, and $\alpha_d \in [0,1]$ is related to the cutoff frequency by

$$f_c = \frac{\alpha_d}{(1-\alpha_d)2\pi} f_s. \tag{14}$$

The sampling frequency for all experiments is $f_s = 20$ Hz.

VI. Experimental Study: Estimating Ground, Ceiling, and Wall Proximity

A. Expanded DNN Input Space

In addition to the information terms, the DNN inputs are further expanded with IMU and attitude estimation error to estimate physical ground, ceiling, and wall obstacle distances. These inputs are believed to improve the estimation accuracy by encoding additional vehicle behavior. The onboard IMU signals are translational acceleration $\boldsymbol{a} = [a_x, a_y, a_z]$ and rotational velocity $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$. The attitude error is determined by an EKF, $\hat{\boldsymbol{\Theta}}_e = [\hat{\phi} - \phi_m, \hat{\theta} - \theta_m, \hat{\psi} - \psi_m]^T$, where the estimated roll, pitch, and yaw are denoted by $\hat{\phi}$, $\hat{\theta}$, and $\hat{\psi}$, respectively. The measured attitude, denoted by a subscript m, comes from the on-board IMU or measured with a motion capture system. Finally, the six different DNN input configurations for evaluation are: $I_1 = [\boldsymbol{\alpha}]$, $I_2 = [\boldsymbol{\alpha}, \boldsymbol{a}, \boldsymbol{\omega}]$, $I_3 = [\boldsymbol{\alpha}, \boldsymbol{a}, \boldsymbol{\omega}, \hat{\boldsymbol{\Theta}}_e]$, $I_4 = [\boldsymbol{\alpha}, \boldsymbol{a}, \boldsymbol{\omega}, \boldsymbol{I}]$, $I_5 = [\boldsymbol{\alpha}, \boldsymbol{a}, \boldsymbol{\omega}, \hat{\boldsymbol{H}}]$, and $I_6 = [\boldsymbol{\alpha}, \boldsymbol{a}, \boldsymbol{\omega}, \boldsymbol{I}, \hat{\boldsymbol{\Theta}}_e]$. It is noted that this approach is vehicle-specific. However, to generalize the process, for example, to larger airframes, higher flight speeds, or to outdoor disturbances, nondimensional analysis is needed to understand the key features for the DNN inputs. Future work will focus on generalization of this approach.

B. DNN Training Details

For each type of obstacle, 200 DNNs were trained for each separate input configuration. Flight data was gathered from the physical quadrotor robot as it flew a sinusoidal trajectory toward and away from each obstacle. The training process is stochastic due to the initialization of weights and dropout regularization for each network. Therefore, to assess the performance of networks trained under each input configuration, the average accuracy over the test set for each of the trained 200 networks using each input configuration is compared.

C. Results and Discussion

Examples of the DNN estimated distances trained on ground, ceiling, and wall, using inputs I_4, I_6, and I_5, respectively, are shown in Fig. 5(a), 5(b), and 5(c). The ground and ceiling distance estimation results are directly compared to estimates developed by Gao *et al.* [11], where they used the power model $\bar{\alpha} = a\,(d_o/R)^b + c$, with the coefficients a, b, and c determined through optimization to fit the measured motor data. The model was then inverted to estimate ground and ceiling distances. The results show that the average ground estimation accuracy is similar between the DNN estimator and the power model

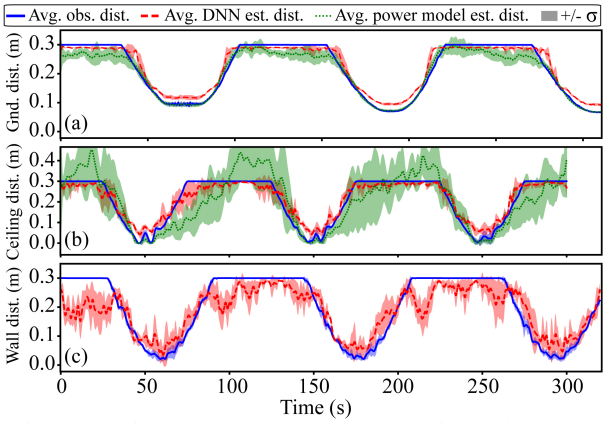

Fig. 5: Comparison between true, DNN estimated, and power model [11] estimated distances over six trials for (a) ground, (b) ceiling, and (c) wall cases where the DNNs were trained on I4, I6, and I5, respectively.

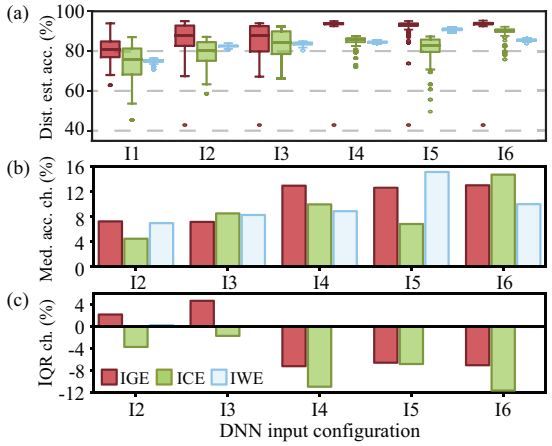

Fig. 6: (a) Accuracy of DNN estimation of ground, ceiling, and wall distance versus input configuration. Change in the (b) median DNN prediction accuracy and (c) variability for each input configuration and obstacle type.

(94.0% vs. 94.5%). However, for the ceiling case, the DNN performed significantly better on average (91.2% vs. 82.4%), with considerably lower variation between trials. It is noted that no model was compared against for the wall case because existing models are specific to a vehicle configuration [14]. The results of distance estimation accuracy are shown in Fig. 6(a), along with the change in median estimation accuracy, and the spread of the accuracy for each input compared to I_1 results in Fig. 6(b) and 6(c). The ASO ε_{min} values associated with the performance of each input are shown in Table III. The baseline median accuracies for I_1, for ground, ceiling, and wall, are 81.04%, 74.35%, and 75.51%, respectively. In Fig. 6(b), as IMU data, EKF error, and/or information-based inputs are introduced, the median accuracy increases in all cases. However, each of the best median accuracies came from input with information-based terms. For ground with I_4 the accuracy is 93.35%; for ceiling with I_6 the accuracy is 89.22%; and for wall with I_5 the accuracy is 90.67%. The ASO results in Table III show that for each of these DNN configurations, the prediction accuracy performance is statistically significant over DNNs using the other inputs. Furthermore, when input with information-based terms (I_4-I_6) is used, the obstacle distance prediction accuracy IQR was reduced on average by 6.96%, 9.84%, and $\approx$ 0.00%, for the ground, ceiling, and wall cases, respectively, compared to the IQR achieved by the baseline DNNs. This reduced range of the prediction accuracy leads to increased repeatability of the DNN training process. For the IQR range of the DNNs using non-information-based inputs (I_1-I_3), the wall distance accuracy IQR on average is 9.87% lower than the ground and ceiling distance accuracy IQR. This difference is equivalent to the largest reduction of IQR achieved by either the ground or ceiling DNNs when the inputs I_4-I_6 were used. This may explain why adding information-based inputs failed to result in any substantial IQR decrease for the wall detection DNNs as the baseline accuracy estimate performance was already far more repeatable.

D. Generalization

The trained DNNs associated with inputs I_1, I_3, I_4, I_5, and I_6 were subjected to multiple obstacles as shown in Fig. 7(a), (c), and (e) to test their ability to generalize. These inputs were chosen because they achieved the most accurate obstacle distance estimation from the previous experiment. For each trajectory, 10 trials were conducted, each trial lasted approximately 4.5 minutes. The accuracy results are shown in Fig. 7(b), (d), and (f), and divided into zones, where Zone 1, 3, and 5 mimic the trained behavior but the position relative to the edges of the obstacle changes. Zones 2 and 4 are regions where the IPE is near-constant and strongly felt, while Zone 6 and 7 are areas where the robot transitions from strong IPE to weak IPE, or vice versa. Finally, Zone 8 and 9 combine motion in both perpendicular and horizontal motion relative to the obstacles, with the first is in front of the obstacles and the latter is beyond the obstacle. Many of the zones include movements that are not captured within the training data sets.

For all cases, the results show the DNNs using information-based inputs (I_4, I_5, or I_6) achieved a higher median accuracy than the baseline case and, with the exception of the wall case, also resulted in a lower accuracy spread. The average decrease in the median accuracy for ground, ceiling, and wall is 1.41%, 9.68%, and 11.2%, respectively. The ground detection, which resulted in the highest median training accuracies, also proved to be the most robust when the environment changes. Similarly, wall detection had the lowest median training accuracy and was shown to be the most sensitive to environmental changes.

Finally, the combined results for each zone and input configuration are shown in Fig. 7(g) and 7(h). The information-based input I_4, I_5, and I_6 are grouped. In general, information-based inputs generalized better to new environments within the training sets. However, when the robot operated very close to the obstacles, input I_1 that captures thrust or torque through motor inputs performed similarly or better than using information-based inputs in Zone 2 and 4. These zones show that the IPE is relatively strong; therefore, the motor inputs are sufficient to accurately estimate obstacle proximity. When the IPE is weaker, additional input terms are needed to deal with low signal-to-noise ratio and measurement uncertainty.

VII. Collision Avoidance Demonstrations

The DNN obstacle distance estimation is used as a feedback signal for closed-loop collision avoidance. The control structure

TABLE III: ε_{min} **results of almost stochastic order testing on DNNs using different input configurations.**
Ground/ceiling/wall distance estimation

Input configuration	I1	I2	I3	I4	I5	I6
I1	–	1.00/0.98/0.99	1.00/0.99/0.99	1.00/0.98/1.00	1.00/1.00/0.99	0.98/0.98/0.99
I2	0.30/0.05/0.01	–	0.48/0.99/0.99	1.00/0.99/1.00	1.00/1.00/1.00	1.00/0.99/0.68
I3	0.47/0.04/0.01	1.00/0.02/0.01	–	1.00/0.86/1.00	1.00/0.06/1.00	1.00/0.99/0.55
I4	0.06/0.03/0.01	0.16/0.03/0.01	0.09/0.30/0.00	–	0.00/0.00/1.00	1.00/1.00/0.77
I5	0.07/0.01/0.01	0.17/0.31/0.01	0.10/1.00/0.00	1.00/1.00/0.01	–	1.00/0.98/0.00
I6	0.04/0.02/0.26	0.11/0.01/1.00	0.09/0.01/1.00	0.66/0.01/1.00	0.60/0.07/1.00	–

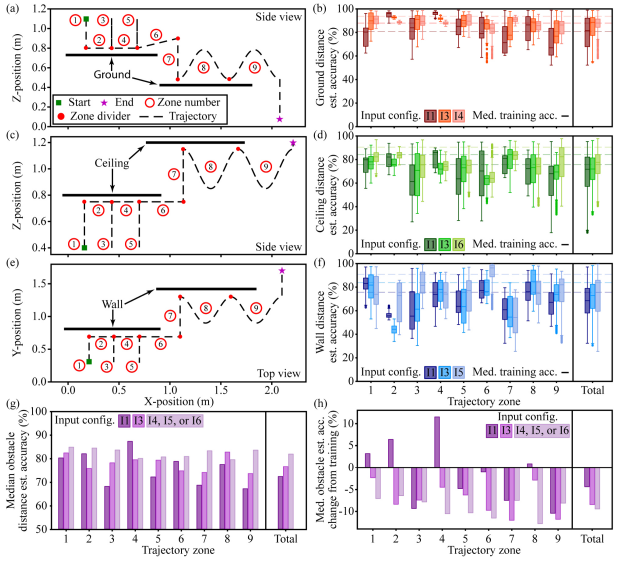

Fig. 7: Multiple obstacle configurations and trajectories for (a) ground, (c) ceiling, (e) and walls. DNN distance prediction accuracies versus zone for (b) ground, (d) ceiling, and (f) wall. (g) Combined prediction accuracy versus input configuration. (h) Median percent change from the training accuracy versus zone and input configuration.

is shown in Fig. 3, where the outer loop, in the absence of obstacles, generates attitude and thrust setpoint $[\phi_d, \theta_d, \psi_d, T_d]$ to track a goal x_g position. When nearby obstacles are present, the DNN output augments the setpoint signal to the robot, enabling it to avoid the obstacle. As shown in Fig. 3, each DNN produces a distance estimate $\hat{d}_g$, $\hat{d}_c$, or $\hat{d}_w$, for ground, ceiling, or wall, respectively. The in-proximity adjustment process then determines the intensity with which the setpoint is modified to avoid the obstacle. The adjustment mimics potential field (PF) through a linearly increasing piecewise function based on the predicted distance to an obstacle $\hat{d}_o$:

$$v_o = \begin{cases} \beta_o(r_o + s_o) & \hat{d}_o < r_o \\ \beta_o(r_o + s_o - \hat{d}_o) & r_o \leq \hat{d}_o \leq (r_o + s_o) \\ 0 & \hat{d}_o > (r_o + s_o) \end{cases} \tag{15}$$

Here, $v_{o,(x,y,z)}$ is the resultant velocity that affects the next desired position, while r_o and s_o are threshold distances and β_o is a scaling factor that changes the maximum intensity of the PF. The closer the estimated distance, the stronger the effect of the PF to repel the robot away from the obstacle.

The collision-avoidance scheme is first applied to avoid ground, ceiling, and wall obstacles as shown in Fig. 8(a), (b), and (c), respectively. First, the robot begins at $\boldsymbol{x}_s$ and moves toward the goal position $\boldsymbol{x}_g$, with an attractive PF while repelling away from any DNN-detected obstacles. Next, the obstacles are removed and the same algorithm is run, where the robot again traverses between the start and goal positions as shown by the red trajectory. The results in Fig. 8(a)-(c) demonstrate successful collision avoidance.

Finally, the demonstration in Fig. 8(d) shows that the robot, running the same collision avoidance scheme with simultaneous IGE, ICE, and IWE estimation (running at 12.5 Hz), can traverse a more complex environment that includes a ramp with a level platform, connected to a wall feature, followed by a ceiling obstacle. These features were not part of the training data set. The demonstration utilizes two goal positions (see Fig, 8(d)), $\boldsymbol{x}_{g,1}$ and $\boldsymbol{x}_{g,2}$, where the transition between them occurs when the robot's $x-y$ planar position error from the first goal decreases below a threshold distance, given by $\sqrt{(x-x_{g,x})^2+(y-x_{g,y})^2} < 0.03$ m. As shown in Fig. 8(d), the robot starts at the bottom of the ramp, detects the ramp feature and traverse up the ramp without collision. At the top of the ramp, it detects the ground and wall features and traverses around towards the goal. As it moves to the goal, it detects and avoids the ceiling. Running the same algorithm without obstacles the robot traverses from the start location directly to Goal 1, then directly to Goal 2 as shown in Fig. 8(e).

VIII. Conclusions

Simulations and physical experimental results demonstrated the ability to use information-based supervised learning to process onboard motor commands and IMU signals to estimate the distance of the ground, ceiling, and wall. Reliable obstacle detection with a median proximity estimation accuracy of 93.35%, 89.22%, and 90.67% for the ground, ceiling, and walls, respectively, was achieved. Demonstrations showed the successful detection and avoidance of nearby obstacles in a 3D environment using this new form of "sensing". This method, while effective, is currently limited to slow to moderate vehicle movements to ensure limited angular offset relative to obstacle surfaces. As the vehicle speed and body angle offset increase, the IPEs manifest differently. Accounting for these changes and enabling greater travel speeds for IPE-based collision avoidance remains an area for future work.

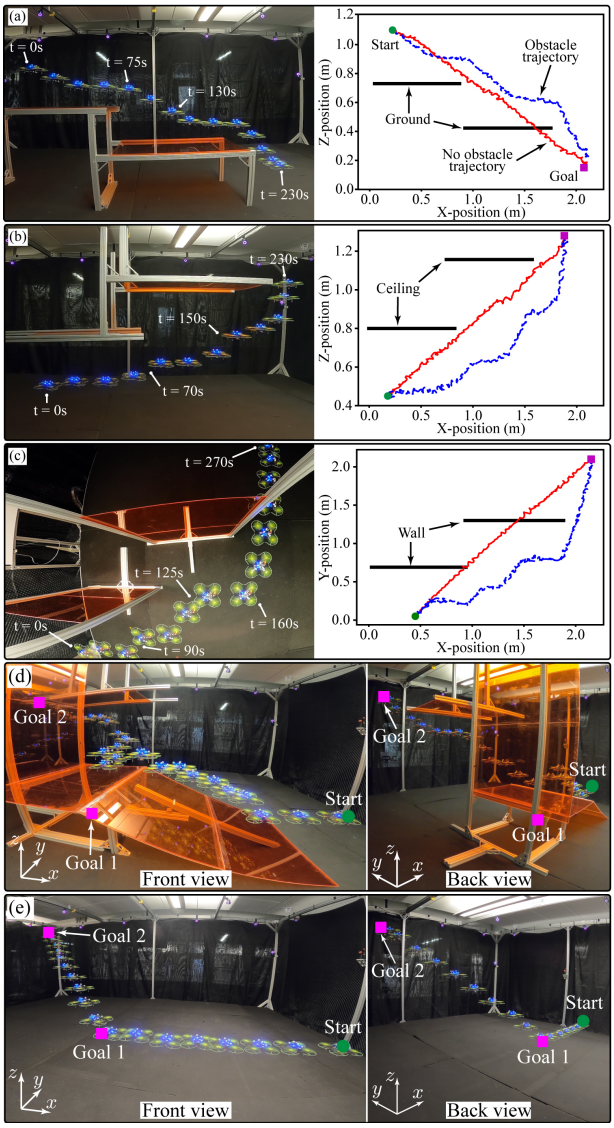

Fig. 8: Collision avoidance using DNN distance estimation as feedback: (a) ground (b) ceiling, and (c) wall features. No-obstacle trajectory in red and collision-avoidance trajectory in blue. (d) Time-evolving snapshots for simultaneous IGE, ICE, and IWE collision avoidance. (e) Behavior without obstacles.

REFERENCES

[1] A. Matus-Vargas, G. Rodriguez-Gomez, and J. Martinez-Carranza, "Ground effect on rotorcraft unmanned aerial vehicles: A review," *Intelligent Service Robotics*, vol. 14, no. 1, pp. 99–118, 2021.

[2] D. J. Carter, L. Bouchard, and D. B. Quinn, "Influence of the ground, ceiling, and sidewall on micro-quadrotors," *AIAA Journal*, vol. 59, no. 4, pp. 1398–1405, 2021.

[3] J. N. Yasin, S. A. S. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Unmanned aerial vehicles (UAVs): Collision avoidance systems and approaches," *IEEE Access*, vol. 8, pp. 105 139–105 155, 2020.

[4] J. W. Starr and B. Y. Lattimer, "Evaluation of navigation sensors in fire smoke environments," *Fire Techn.*, vol. 50, no. 6, pp. 1459–1481, 2014.

[5] X. He and K. K. Leang, "Quasi-steady in-ground-effect model for single and multirotor aerial vehicles," *AIAA journal*, vol. 58, no. 12, pp. 5318–5331, 2020.

[6] S. A. Conyers, "Empirical evaluation of ground, ceiling, and wall effect for small-scale rotorcraft," Ph.D. dissertation, Univ. of Denver, CO, 2019.

[7] I. Franzetti, R. Parin, E. Capello, M. J. Rutherford, and K. P. Valavanis, "Ground, ceiling and wall effect evaluation of small quadcopters in pressure-controlled environments," *J. of Int. Robotic Syst.*, vol. 110, no. 3, p. 125, 2024.

[8] X. He, G. Kou, M. Calaf, and K. K. Leang, "In-ground-effect modeling and nonlinear-disturbance observer for multirotor unmanned aerial vehicle control," *Journal of dynamic systems, measurement, and control.*, vol. 141, no. 7, p. 071013, 2019.

[9] D. Lee, A. Awan, S. Kim, and H. J. Kim, "Adaptive control for a VTOL UAV operating near a wall," in *2012 AIAA Guidance, Navigation, and Control Conference*, Conference Proceedings.

[10] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, *Influence of aerodynamics and proximity effects in quadrotor flight*. Springer International Publishing, 2013, pp. 289–302.

[11] S. Gao, C. D. Franco, D. Carter, D. Quinn, and N. Bezzo, "Exploiting ground and ceiling effects on autonomous UAV motion planning," in *Int. Conf. on Unmanned Aircraft Systems*. IEEE, 2019, pp. 768–777.

[12] C. D. McKinnon and A. P. Schoellig, "Estimating and reacting to forces and torques resulting from common aerodynamic disturbances acting on quadrotors," *Robotics and Auton. Systems*, vol. 123, p. 103314, 2020.

[13] G. van Dam and E.-J. Van Kampen, "Obstacle avoidance for quadrotors using reinforcement learning and obstacle-airflow interactions," in *AIAA Scitech 2020 Forum*, Conference Proceedings, p. 2249.

[14] R. Ding, S. Bai, K. Dong, and P. Chirarattananon, "Aerodynamic effect for collision-free reactive navigation of a small quadcopter," *npj Robotics*, vol. 1, no. 1, p. 2, 2023.

[15] I. Cheeseman and W. Bennett, "The effect of the ground on a helicopter rotor in forward flight," *Techn. Report 3021, Aeron. Res. Council*, 1955.

[16] S. A. Conyers, M. J. Rutherford, and K. P. Valavanis, "An empirical evaluation of ground effect for small-scale rotorcraft," in *IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 1244–1250.

[17] ——, "An empirical evaluation of ceiling effect for small-scale rotorcraft," in *IEEE Int. Conf. on Unman. Aircraft Syst.*, 2018, pp. 243–249.

[18] L. Yiliang, K. Zi, L. Huadong, G. Yuzhe, L. Daochun, and Z. Shiwei, "Analysis and modeling of the aerodynamic ceiling effect on small-scale propellers with tilted angles," *Aerospace Science and Technology*, vol. 147, p. 109038, 2024.

[19] H. Li, X. Dong, Y. Gao, Y. Liu, D. Bie, D. Li, J. Xiang, and Z. Tu, "Numerical analysis on the ducted propeller aerodynamics in sidewall-ground effect," *Physics of Fluids*, vol. 36, no. 12, p. 125148, 2024.

[20] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[21] R. W. Beard, "Quadrotor dynamics and control," *Brigham Young University*, vol. 19, no. 3, pp. 46–56, 2008.

[22] D. Ulmer, C. Hardmeier, and J. Frellsen, "Deep-significance: Easy and meaningful significance testing in the age of neural networks," in *Int. Conf. on Learning Rep. [ML Eval. Standards Workshop]*, 2022.

[23] M. W. Mueller, M. Hehn, and R. D'Andrea, "Covariance correction step for Kalman filtering with an attitude," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2301–2306, 2017.

[24] F. Gustafsson, "Determining the initial states in forward-backward filtering," *IEEE Trans. on Signal Proc.*, vol. 44, no. 4, pp. 988–992, 1996.

Jacob M. Anderson received his B.S. degree in mechanical engineering from the Univ. of Utah in 2020. He is currently pursuing a Ph.D. degree in mechanical engineering at the University of Utah. His research interests include unmanned aerial vehicles, control systems, deep learning methods, information theory, and autonomy.

Kam K. Leang (M'02) received the B.S. and M.S. degrees in mechanical engineering from the Univ. of Utah, Salt Lake City, in 1997 and 1999, respectively, and the Ph.D. degree from the Univ. of Washington, Seattle, in December 2004. He is a Professor in the Mechanical Engineering Department and member of the Robotics Center at the University of Utah. His current research focuses on control, estimation, and motion planning for robotics-based emergency response and environmental monitoring. He currently serves as an Associate Editor for Mechatronics (Elsevier) and ASME Letters in Dynamic Systems and Control. He has been involved with conference organization and editorialship, including the American Control Conference (ACC), IEEE Int. Conference on Robotics and Automation (ICRA), and IEEE/ASME Int. Conference on Advanced Intelligent Mechatronics (AIM). He is also a Fellow of ASME and Senior Member of IEEE.