Received 19 June 2021; revised 23 July 2021; accepted 6 August 2021. Date of publication 24 August 2021; date of current version 10 September 2021.

Digital Object Identifier 10.1109/OJCOMS.2021.3106274

FPV Video Adaptation for UAV Collision Avoidance

SIMRAN SINGH[1] (Member, IEEE), HEE WON LEE[2], TUYEN X. TRAN[3] (Member, IEEE), YU ZHOU[3], MIHAIL L. SICHITIU[1] (Member, IEEE), ISMAIL GÜVENÇ[1] (Fellow, IEEE), AND ARUPJYOTI BHUYAN[4] (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606, USA
[2]Data and Information Technology Center, Samsung Electronics, New York, NY 10014, USA
[3]Department of Network Analytics and Automation, AT&T Research Labs, Bedminster, NJ 07921, USA
[4]Wireless Security Institute, Idaho National Laboratory, Idaho Falls, ID 83402, USA

CORRESPONDING AUTHOR: S. SINGH (e-mail: ssingh28@ncsu.edu)

This work was supported in part by the AT&T Labs, Bedminster, NJ, USA, through the National Science Foundation under Grant CNS-1453678 and Grant CNS-1738093, and in part by the Idaho National Laboratories (as part of Laboratory Directed Research Development Program through DOE Idaho Operations Office) under Contract DEAC07-05ID14517.

ABSTRACT First person view (FPV) technology for unmanned aerial vehicles (UAVs) provides an immersive experience for pilots and enables various personal and commercial applications such as aerial photography, drone racing, search and rescue operations, agricultural surveillance, and structural inspection. While real time video streaming from a UAV and vision-based collision avoidance strategies have been studied in literature as separate topics, in this paper we tackle collision avoidance in FPV scenarios, taking into account network delays and real time video parameters. We present a theoretical model for obstacle collisions that considers the current communication channel conditions, the real time video parameters, and the UAV's position relative to the closest obstacle. A video adaptation algorithm is then designed, using this metric, to tune the FPV video resolution, number of re-transmission attempts, and the modulation scheme to maximize the probability of avoiding collisions. This algorithm also takes into account specific latency constraints of the application. This video algorithm was evaluated in various scenarios and its ability to respond to both distances to the obstacle as well as the communication channel conditions was demonstrated. It was found that, for the considered scenarios, the performance of the proposed adaptive algorithm was, on an average, 58.63% higher than the closest non-adaptive one in terms of maximizing the probability of avoiding collision. Such collision avoidance strategies could be used to make UAV FPV applications safer and more reliable.

INDEX TERMS Collision avoidance, FPV, safety, UAV, video adaptation.

I. INTRODUCTION

FIRST person view (FPV) technology provides the vantange point on an unmanned aerial vehicle (UAV) to a ground pilot, allowing the pilot to manoeuvre the UAV based on the video feed from the UAV's camera. The applications of UAV FPV technology are spread across the spectrum ranging from pure leisure, such as UAV racing and aerial photography, to critical services such as search and rescue operations in adverse environments. Other applications include agricultural inspections, live broadcasting, and remote area patrol. Infrastructure to support such UAV applications, such as the design of drone corridors, is being studied extensively and put to practice [1], [2]. When piloting a UAV remotely for such applications, avoiding collisions with obstacles is a fundamental requirement. This requires the FPV video feed to be adapted such that the pilot is able to recognize the obstacle with sufficient time to stop the UAV before collision.

The performance of existing FPV systems vary considerably in the quality of the video being transmitted, as well as the delay between the capture device and display. The best systems adapt the streaming parameters to the current conditions. This video feed adaptation has to be performed dynamically since the quality of the air to ground (AG) wireless link from the UAV to the pilot may vary with both time and position, due to factors such as the presence of scatterers, obstructions in the line of sight path, and Doppler effect due to the UAV's mobility. The effects of ground reflections, shadowing due to obstructions, path loss due to altitude, and antenna beam orientation on the wireless coverage for UAVs was studied in [3], [4], when using ground base stations with uptilted antennas for UAVs, in conjunction with downtilted antennas for ground users. A 3GPP study [5] reported that UAVs experienced a higher handover failure rate and a longer handover interruption time than terrestrial users. To avoid collisions, this video adaption has to be performed with the specific objective of maximizing the probability of recognizing an obstacle. This requires an understanding of the response of the human visual recognition system to objects of various dimensions, along with knowledge of the trade-offs between video resolution and video latency. As determined by the video-codec, higher video resolutions may result in more data being transmitted over the link, a larger transmission time, and an increase in latency.

This trade-off is also influenced by the flight scenario. For instance, when the obstacle is farther away from the UAV (and, therefore, it appears smaller in the video feed), it is desirable to increase the video resolution to allow the pilot to see the finer details and spot the obstacle. When the obstacle is closer, on the other hand, it may be desirable to minimize latency at the expense of video resolution such that the pilot is able to recognize the obstacle quickly before collision. The optimum balance between video resolution and latency is also determined by the UAV's velocity. For example, at a constant distance of the UAV from the obstacle, higher resolutions are preferred when the UAV is moving at a lower velocity towards the obstacle, as the UAV has more time before it collides with the obstacle. In this case, higher resolutions would allow the pilot to recognize the obstacle while it is still far away and the lower velocity allows the pilot sufficient time to stop the UAV before collision. Hence, to make the ever-growing field of UAV FPV applications safer, it is important to study the design of collision-avoidance video adaptation algorithms that take into account the complex interplay between the factors described above.

We analyze this scenario and derive a collision avoidance metric that is a function of the UAV's orientation with reference to the nearest obstacle, the real-time communication channel conditions, and the FPV feed parameters. The key feature of this metric is that it captures the trade-offs inherent in adapting the FPV stream to avoid collision. A simple yet effective strategy is then proposed to adapt the UAV FPV feed to optimize this metric and thereby maximize the probability of avoiding collision. In contrast to existing commercial FPV systems [6] that provide various video quality configurations without being responsive, our proposed video adaptation is dynamic. Further, unlike other video adaption algorithms in the industry and the academia [7], [8] that maximize objectives such as the quality of experience of a teleconferencing user, our algorithm focuses on maximizing the probability of avoiding collision. Additionally, our strategy may be tailored to the latency requirements of different applications, allowing the UAV to switch between these. For instance, while also minimizing the chances of collisions, the video adaptation strategy may be configured to meet the stringent low latency requirements of FPV racing or the more relaxed latency requirements of leisure applications such as aerial photography.

This algorithm does not require a priori knowledge of the UAV's intended trajectory nor information about the location of obstacles in the environment. Rather, the algorithm relies on current and past sensory inputs from the UAV's autopilot, camera, and communication modules to gauge the UAV's velocity, distance to closest obstacle, and the current communication channel conditions, respectively. This makes the algorithm generic such that it can also be deployed in unknown environments, rather than being optimized for pre-determined trajectories in known environments. This distinguishes our work from studies that focus on the problem of obstacle avoidance in conjunction with trajectory tracking [9], where the trajectory that the UAV has to navigate is known completely or partially, for example when the UAV is used to spray an agricultural field or survey infrastructure. We also evaluate this video adaptation algorithm for various latency constraints, and demonstrate its ability to respond to changes in channel conditions as well as distance to nearest obstacles.

The contributions of our work can be summed up as follows.

1. We study UAV FPV video adaptation with the specific goal of collision avoidance for UAVs, based on recognizing obstacles in the FPV feed. To the best of the authors's knowledge, this is the first study on this topic, which combines UAV movement, wireless communications, and collision avoidance.

2. We develop a detailed theoretical model for obstacle detection using FPV, which combines UAV mobility, location of obstacles, video encoding, modulation schemes, re-transmission strategies, and human visual perception. While we assume motion-JPEG (MJPEG) encoding, our model may be easily modified to include other encoding schemes by modifying the equation to calculate the encoded frame size.

3. Leveraging this model, we derive a safety metric that represents the probability of collision in a specific scenario.

4. We propose a simple FPV adaptation algorithm, which is guided by this safety metric. The performance of the algorithm is compared with non-adaptive algorithms for a simple scenario.

The rest of this paper is organized as follows. A comparison with related work is presented in Section 1 and the theoretical model of the UAV FPV system is introduced in Section III. Building on this model, in Section IV we derive the probability of avoiding collisions. Section V then formulates the video adaptation problem and presents the results using a greedy algorithm. Finally, Section VI provides concluding remarks.

TABLE 1. Comparison with related work.

Reference	Video Adaptation	Obstacle Detection	Applicable to FPV	Video Adaptation Goal	Obstacle Detection Technique
[7]	✓	×	×	Minimize end to end video delay	Not applicable (N/A)
[9]	×	✓	×	N/A	Sensors
[10]	✓	×	×	Maximize quality of experience	N/A
[11]	✓	×	×	Real-time surveillance	N/A
[12]	✓	×	×	Two possible modes: maximize robustness or throughput	N/A
[13]	×	✓	×	N/A	Ultrasonic and infrared sensors
[14]	×	×	×	N/A	Optical flow sensors and stereo vision
[15]	×	×	×	N/A	Optical flow sensors
[16]	×	✓	×	N/A	Changes in obstacle feature point in consecutive frames.
[17]	×	✓	×	N/A	Thermal images
[6]	×	×	✓	N/A	N/A
[18]	×	×	✓	N/A	N/A
[19]	×	×	✓	N/A	N/A
This work	✓	✓	✓	Maximize probability of avoiding collision	FPV

II. RELATED WORK

Existing work on video adaptation, and obstacle detection and avoidance is reviewed in this section, and contrasted with our contribution. The problem of real-time video transmissions from UAVs is gradually receiving attention in the academia. The research in [10] utilized a deep reinforcement learning (RL) agent to obtain the optimal bit-rate selection strategy under varying channel conditions with the aim of maximizing the general quality of experience, rather than for a specific application. Video adaptation for the specific application of UAV surveillance was studied in [11] and [12]. The work in [11] presented a cross-layer rate-adaptive algorithm meant for UAV surveillance applications. In [12], video adaptation was performed by transmitting a number of layered video streams from the UAV. However, collision detection and avoidance was not addressed in any of these studies.

Obstacle detection, defined as the process of detecting proximity to objects that could potentially impede motion and cause a collision, has been extensively studied in literature for UAVs, using both sensors [13], [14] and image processing techniques [15], [16], [17], [20]. In [13], 16 infrared sensors and 12 ultrasonic sensors were employed to provide 360° coverage for obstacle detection, while [14], [15], [21] leveraged optical flow sensors to estimate the distance to obstacles. Optical flow was supported by stereo vision techniques in [14] whereas [16] relied on a single monocular camera, and estimated the distance to obstacles by calculating the change in size of feature points of obstacles in consecutive frames. The work in [22] similarly detected obstacles based on monocular vision feature points while additionally proposing an obstacle-avoidance method. Optimal algorithms to detect obstacles from thermal infrared images captured by a camera at the UAV were presented in [17]. However, none of the above work attempted to adapt the video transmission parameters to maximize the probability of detecting obstacles from the UAV FPV feed.

The problem of obstacle avoidance has received considerable attention in literature, wherein an on-board auto-pilot is used to manoeuvre the UAV along collision-free trajectories [23], [24], [25], [26]. However, these approaches assume the knowledge about the UAV's trajectory in terms of certain coordinates the UAV has to navigate through. Collision avoidance in multi-UAV swarms has also been studied using techniques such as cooperative formation control [27], [28], [29] and swarm algorithms [30]. However, these approaches require some form of communication and cooperation among the UAVs.

UAV FPV systems that are commercially available also have their shortcomings; for instance, they do not sense and adapt to the channel conditions. The DJI FPV [6] system is a popular commercial option, which provides two video modes- low latency and high video quality, but it is not adaptive, i.e., the system does not switch automatically between these modes as a function of the channel conditions. Digital FPV video transmitters such as Connex Prosight HD [18] (designed for drone racing) or Insight SE [19] employ fixed video resolutions, up to 720p and 1080p respectively, and are also not adaptive. Hence, to the best of our knowledge, literature lacks a study on approaches for video adaptation for UAV FPV applications with the goal of avoiding collisions.

We address this gap by presenting a dynamic video adaptation algorithm that tunes the parameters of a real-time FPV video stream with the goal of allowing a pilot to avoid obstacles. Specifically, for each image frame captured at the UAV's camera, this algorithm selects the video resolution, and number of re-transmission attempts, and modulation scheme, taking into account the current communication channel conditions (in terms of channel error rate and transmission rate), physical variables (the UAV's velocity and distance to nearest obstacle), and application requirements (such as constraints on latency), in order to maximize the probability of avoiding collision with the closest obstacle. A comparison of our work with related work is presented in Table 1. Next, we present the theoretical model, wherein for a given scenario, the probability of avoiding collision is derived. The pilot can be (conceptually, or actually)

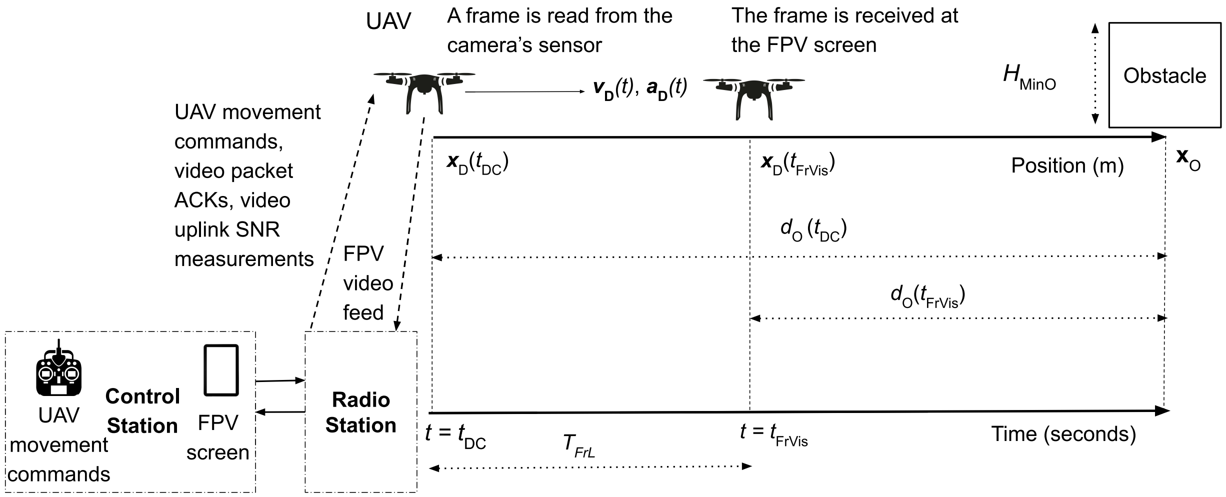

UAV
A frame is read from the camera's sensor
The frame is received at the FPV screen
H_{MinO}
Obstacle
$\mathbf{v}_{\mathrm{D}}(t)$, $\mathbf{a}_{\mathrm{D}}(t)$
UAV movement commands, video packet ACKs, video uplink SNR measurements
FPV video feed
$\mathbf{x}_{\mathrm{D}}(t_{\mathrm{DC}})$
$\mathbf{x}_{\mathrm{D}}(t_{\mathrm{FrVis}})$
Position (m)
$\mathbf{x}_{\mathrm{O}}$
$d_{\mathrm{O}}(t_{\mathrm{DC}})$
$d_{\mathrm{O}}(t_{\mathrm{FrVis}})$
UAV movement commands
Control Station
FPV screen
Radio Station
$t = t_{\mathrm{DC}}$
T_{FrL}
$t = t_{\mathrm{FrVis}}$
Time (seconds)

FIGURE 1. An overview of the scenario under consideration, showing the UAV, the obstacle, the RF controller, and the FPV screen. Relevant variables are also shown.

replaced by an obstacle avoiding algorithm based exclusively on processing the video at the remote control station.

III. THEORETICAL MODEL OF THE UAV FPV SYSTEM

In this section, we develop the theoretical model of the UAV FPV system. First, various functional blocks of the system are introduced, at the UAV and at the control station. Then, we mathematically describe the UAV's motion and the journey of an image frame as it is captured at the UAV, encoded and transmitted over the AG link to the radio station, forwarded from the radio station to the control station, decoded, and finally displayed. We then model the response of human visual system to obstacles of different size in this decoded frame. An expression for the probability distribution of frame latency is derived, which is then utilized in Section IV to derive the probability of avoiding collision. Upper case notations are used to denote parameters that may vary across FPV systems but are constant within a given FPV system. Lower case notations denote variables, calligraphic notations denote sets, and boldface notations denote vectors. Time instants are represented using the lower case while time intervals using the upper case.

As shown in Fig. 1, consider a UAV in flight, equipped with a camera at front that captures and transmits video in real time over a wireless link to a radio station, which forwards the video to the control station. The radio station and the control station may be housed in separate hardware units, in which the video feed is forwarded from the radio to the control station via a back-haul network and/or the Internet. Alternately, the radio and control stations may be co-located within the same physical entity. We assume that the link between the radio and control stations has sufficient capacity, while the wireless link between the UAV and the radio station is subject to impairments and is the main data rate bottleneck. The control station decodes, acknowledges, and displays the live video feed on a display screen, which we term the *FPV screen*. This FPV screen could be a screen on the UAV's remote control unit, a separate monitor, a smart device, or (augmented reality) FPV goggles. By viewing this FPV screen, the pilot manoeuvres the UAV along its intended path by sending commands from the control station to change the UAV's yaw, pitch, roll, and throttle. If the pilot detects the presence of an obstacle in time, it takes action to avoid the collision (by breaking, or dodging). In this work, we address the problem of adapting the FPV video feed such that the probability of avoiding collision is maximized.

Relevant functional components of the UAV and the control station are shown in Fig. 2. The stream of images captured from the UAV's camera are processed by the *Depth Estimator* module, which generates a depth pixel map using a technique such as that presented in [31]. Based on this depth map, the distance to the closest obstacle is calculated.

Acknowledgements to the received packets and the signal to noise ratio (SNR) of the UAV's transmitter (which we term the uplink SNR), as measured at the radio station, are sent from the control station to the UAV. Based on the SNR measurements, the *FPV Controller* chooses an appropriate modulation scheme, and thereby determines the transmission rate as well as the packet error rate. The UAV's transmitter re-transmits unacknowledged packets after their time-out interval. The *FPV controller* also receives estimates of the UAV's velocity and acceleration from the autopilot. It processes all the above mentioned inputs, along with the requirements of the FPV application and decides the video resolution and number of re-transmission attempts for each captured image frame. Accordingly, the *Video Encoder and Transmitter* block processes and transmits these captured frames to the ground station.

A. ENCODING AND TRANSMISSION OF THE FPV VIDEO

We now model the encoding, transmission, and reception of the UAV FPV video feed. At any given time instant t,

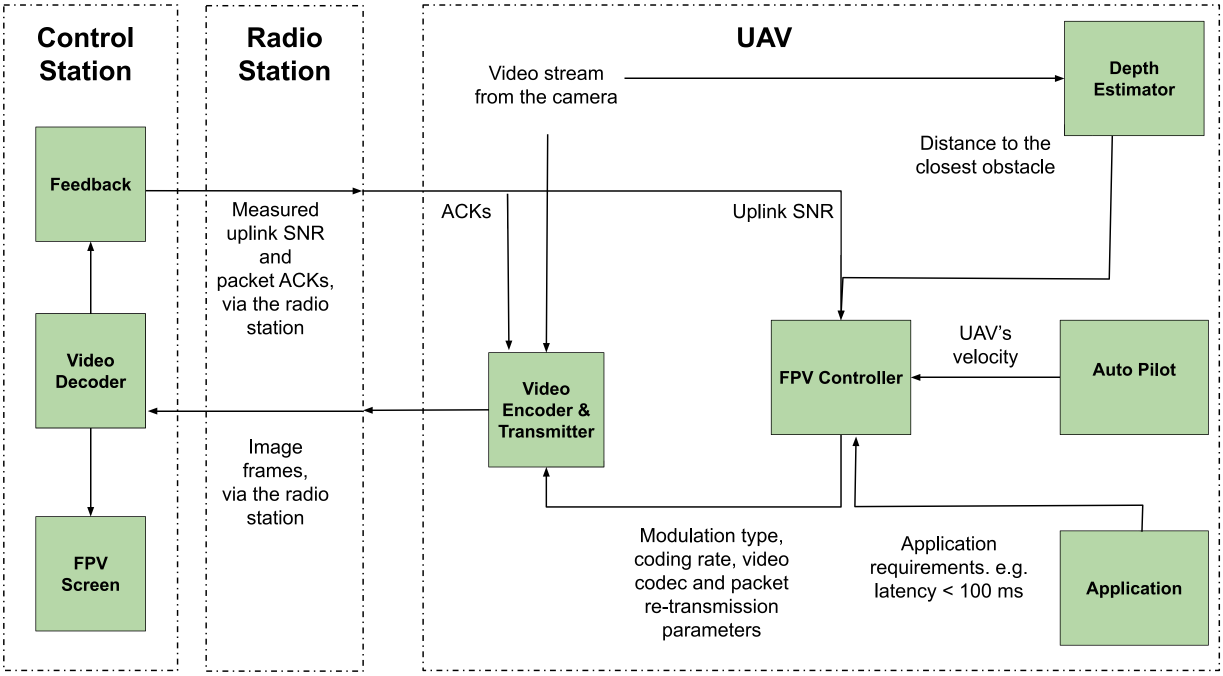

Control Station
Radio Station
UAV
Feedback
Video Decoder
FPV Screen
Measured uplink SNR and packet ACKs, via the radio station
Image frames, via the radio station
Video stream from the camera
ACKs
Uplink SNR
Depth Estimator
Distance to the closest obstacle
Video Encoder & Transmitter
FPV Controller
UAV's velocity
Auto Pilot
Modulation type, coding rate, video codec and packet re-transmission parameters
Application requirements. e.g. latency < 100 ms
Application

FIGURE 2. Functional blocks of the FPV collision avoidance system.

let the three-dimensional (3D) coordinates of the current position be denoted as $\mathbf{x}_{\mathrm{D}}(t) = \{x_{\mathrm{Dx}}(t), x_{\mathrm{Dy}}(t), x_{\mathrm{Dz}}(t)\}$, and the 3D vector representing the current velocity as $\mathbf{v}_{\mathrm{D}}(t) = \{v_{\mathrm{Dx}}(t), v_{\mathrm{Dy}}(t), v_{\mathrm{Dz}}(t)\}$. $\mathbf{x}_{\mathrm{D}}(t)$ is calculated as:

$$\mathbf{x}_{\mathrm{D}}(t) = \int_{0}^{t} \mathbf{v}_{\mathrm{D}}(t)dt \tag{1}$$

Pixels are read from the charge-coupled device (CCD) of the UAV's camera as an image frame. We assume that the video codec at the UAV uses an adaptation strategy to decide the frame resolution of the captured frame at any given instant t. Let this frame resolution be denoted as the set $\{w_{\mathrm{Fr}}(t), h_{\mathrm{Fr}}(t)\}$, where $w_{\mathrm{Fr}}(t)$ is frame's width and $h_{\mathrm{Fr}}(t)$ is the frame's height, both in pixels. We assume that the FPV feed is encoded using motion JPEG (M-JPEG), i.e., each raw image frame is encoded as an JPEG image. The compression process is frame-by-frame (unlike H.264). This ensures that the decoding of each frame is independent of previously received frames, and errors in the reception of a frame are not carried over to the next frames. The JPEG quality factor is a parameter which determines the quality and size of compressed. We assume that the FPV operator chooses a quality factor, which results in a compression factor of C_{JPG}. The size of the encoded JPEG frame is then calculated as:

$$n_{\mathrm{JPG}}(t) = C_{\mathrm{JPG}} w_{\mathrm{Fr}}(t) h_{\mathrm{Fr}}(t) N_{\mathrm{Color}}, \tag{2}$$

where N_{Color} is the color depth of the frame in bits. Studies have shown that compression factors of upto 0.1 are practically imperceptible to the human visual system [32], [33], [34]. This encoded JPEG frame is then assumed to be split into packets of maximum size $n_{\mathrm{P}}(t)$. The number of such packets is calculated as:

$$n_{\mathrm{Pkt}}(t) = \mathrm{ceil}\left(\frac{n_{\mathrm{JPG}}(t)}{n_{\mathrm{P}}(t)}\right) \tag{3}$$

where ceil is the ceiling function (i.e., round up function). We further assume that each packet incurs an overhead of N_{Ov} bits. The total number of bits transmitted is thus:

$$n_{\mathrm{Fr}}(t) = n_{\mathrm{JPG}}(t) + n_{\mathrm{Pkt}}(t) N_{\mathrm{Ov}}, \tag{4}$$

where $n_{\mathrm{Fr}}(t)$ represents the total number of bits transmitted for the image frame that was captured at time instant t. It is assumed that all the packets representing a frame are transmitted consecutively. In this work, for simplicity, we do not consider the case wherein the frame is partially received, i.e., we assume that all packets representing a frame have to be successfully decoded and displayed.

It is assumed that the frame encoding and transmission processes described above are performed in separate threads at the UAV's computing resource. Assume that a frame has just been transmitted at time instant t, then let $T_{\mathrm{InterFr}}(t)$ denote the time interval until the next frame transmission, i.e., $T_{\mathrm{InterFr}}(t)$ denotes the inter-frame interval at time t. Let $t_{\mathrm{n}}^{\mathrm{Fr}}$ represent the time instant at which the n^{th} frame is transmitted. Then, $t_{\mathrm{n}}^{\mathrm{Fr}}$ may be calculated recursively based on the time of transmission of the previous frame, $t_{\mathrm{n\text{-}1}}^{\mathrm{Fr}}$:

$$t_{\mathrm{n}}^{\mathrm{Fr}} = t_{\mathrm{n\text{-}1}}^{\mathrm{Fr}} + T_{\mathrm{InterFr}}\left(t_{\mathrm{n\text{-}1}}^{\mathrm{Fr}}\right). \tag{5}$$

Let $\mathcal{T}_{\mathrm{Fr}}(t)$ represent the set of all time instants from $t = 0$ to t at which a frame is transmitted from the UAV. Then,

$$\mathcal{T}_{\mathrm{Fr}}(t) = \left\{t_{1}^{\mathrm{Fr}}, t_{2}^{\mathrm{Fr}}, \ldots, t_{n}^{\mathrm{Fr}}\right\}, \tag{6}$$

such that $t^{\text{Fr}}_{n-1} + T_{\text{InterFr}}(t^{\text{Fr}}_{\text{n-1}}) < t$, and $t^{\text{Fr}}_{n} + T_{\text{InterFr}}(t^{\text{Fr}}_{\text{n}}) > t$. The cardinality of this set, denoted by $|T_{\text{Fr}}(t)|$, represents the number of frames that are transmitted from the UAV before time t. The inter-frame interval determines the frames per second (FPS) of the FPV feed. We note that the FPS of the feed may vary with time as the inter-frame interval may be uneven.

B. UAV TO RADIO STATION AG PROPAGATION MODEL

AG propagation models have been studied extensively in literature [35]. Our proposed FPV system adapts to channel conditions by relying on the feedback of acknowledgements and SNR measurements from the control station to the UAV *FPV controller*. For demonstration purposes, we use the Ricean AG propagation model derived in [36], which assumed UAVs at low altitudes in a semi-urban region. The model is expressed as a combination of an altitude-dependent, log-distance path loss model, which represents the dominant component, and an altitude-dependent random variable, which represents the scattered component that causes fading. Based on this model, the SNR at time instant t, which we denote as $\gamma(t)$, is calculated as:

$$\gamma(t) = \frac{K_1(h(t)) - 10K_2(h(t))\log_{10}\left(\sqrt{d_{\text{R}}^2(t) - h^2(t)}\right) - X_{\text{R}}(h(t))}{BN_0}, \tag{7}$$

where $h(t)$ is the altitude of the UAV at time instant t, $d_{\text{R}}(t)$ is the distance of the UAV from the radio station at time instant t, $K_1(h(t))$ represents the relative path-loss intercept and depends on the altitude as well as the transmission power, $K_{\text{PLE}}(h(t))$ is the altitude-dependent path-loss exponent, N_0 represents the noise power spectral density, B represents the total signal bandwidth, and $X_{\text{R}}(h(t))$ is the Rician fading random variable at altitude $h(t)$, characterized by the parameter $K_{\text{R}}(h(t))$. $K_{\text{R}}(h(t))$ is calculated as:

$$K_{\text{R}}(h(t)) = K_{\text{R1}} + K_{\text{R2}}h(t) + K_{\text{Y}}, \tag{8}$$

where K_{Y} is a zero-mean Gaussian random variable, with standard deviation δ_Y. The value of δ_Y is calculated in [36], for the specific environment under consideration.

Based on the SNR measurements fed back from the control station, the *FPV Controller* chooses an appropriate M quadrature amplitude modulation (QAM) scheme, where $M(t)$ is the instantaneous modulation order selected by the *FPV controller*. We assume that Gray code bit mapping is used. Since the signal bandwidth is B, the symbol rate is $\frac{B}{2}$, and the net bit transmission rate is hence:

$$r(t) = \frac{B\log_2(M(t))}{2}. \tag{9}$$

The bit error rate, which we denote as $e_{\text{b}}(t)$ is then calculated as [37]:

$$e_{\text{b}}(t) = \frac{4}{\log_2(M(t))}\left(1 - \frac{1}{\sqrt{(M(t))}}\right) \times \sum_{k=1}^{\sqrt{M(t)}/2} Q\left((2k-1)\sqrt{\frac{3B\gamma(t)\log_2(M(t))}{r(t)(M(t)-1)}}\right), \tag{10}$$

and the packet error rate, $e(t)$, as:

$$e(t) = (1 - e_{\text{b}}(t))^{n_P(t)}. \tag{11}$$

C. RE-TRANSMISSION AND RECEPTION OF THE FPV VIDEO

We present the frame re-transmission model, based on which, an expression for frame latency is derived. Assume that a frame is read from the camera's sensor by the video codec at time instant t_{DC}. Let the time instant at which this frame is visible at the FPV screen be denoted as t_{FrVis}. We term the time difference $(t_{\text{FrVis}} - t_{\text{DC}})$ as frame latency, $T_{\text{FrL}}(t)$.

The time taken to process the frame to generate the encoded frame is denoted as T_{ENC}, and the time taken to transmit the encoded frame as single or multiple packets as $T_{\text{TX}}(t)$. $T_{\text{Tx}}(t)$ is the time elapsed between the time instant at which the first bit is sent out and the time instant at which the last bit of a frame is sent out over the air, assuming that all packets representing a frame are transmitted back to back. We can calculate $T_{\text{TX}}(t)$ as:

$$T_{\text{Tx}}(t) = \frac{n_{\text{Fr}}(t)}{r(t)}, \tag{12}$$

The time taken for a packet bit to propagate over the air to the radio station is denoted as $T_{\text{P}}(t)$. This propagation delay, $T_p(t)$, is a function of the distance of the UAV from the radio station, $d_{\text{R}}(t)$, and is calculated as:

$$T_p(t) = \frac{d_{\text{R}}(t)}{c}, \tag{13}$$

where c is the speed of light. $d_{\text{R}}(t)$ is calculated as:

$$d_{\text{R}}(t) = |\mathbf{x}_{\text{D}}(t) - \mathbf{x}_{\text{R}}|, \tag{14}$$

where $\mathbf{x}_{\text{R}}$ represents the location of the radio station, which is assumed to be stationery. $T_p(t)$ varies as a function of the UAV's position. For a packet transmitted at time t, we consider its propagation delay to be as calculated in (13), and ignore the variation in propagation delay due to UAV motion *during* this packet's transmission. The networking delay encountered by the bit as it is forwarded from the radio station to the control station is denoted as T_{N}. The network delay for *any* link consists of five components: processing delay, queuing delay, transmission delay, and medium access delay, of which queuing and medium access delays are the only two components that vary with time. Since we assume that the link between the radio and the control station has sufficient capacity, we neglect queuing delays. For simplicity, the medium access delay is also assumed to be constant, and thus the network delay is assumed to be constant at all time instants.

The time taken to generate the decoded frame is denoted as T_{DEC}. All components of frame latency are shown in Fig. 3.

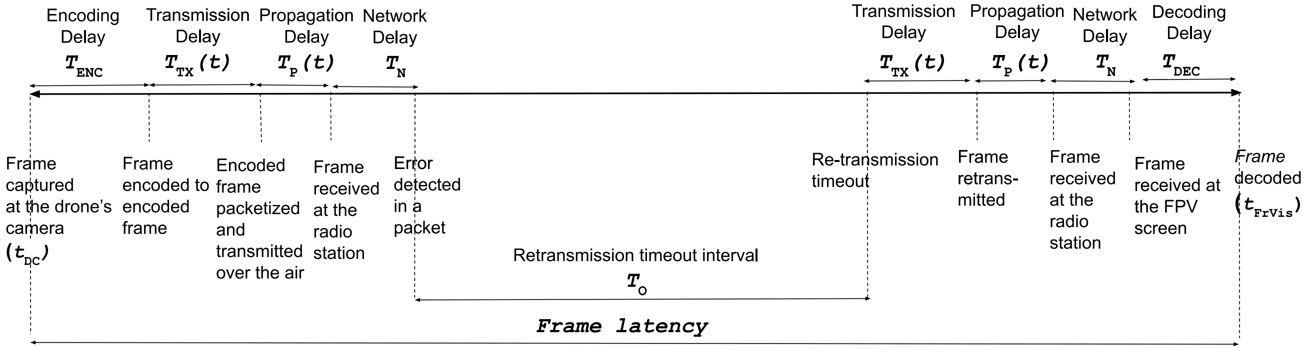

Encoding Delay T_{ENC}
Transmission Delay $T_{TX}(t)$
Propagation Delay $T_P(t)$
Network Delay T_N
Transmission Delay $T_{TX}(t)$
Propagation Delay $T_P(t)$
Network Delay T_N
Decoding Delay T_{DEC}
Frame captured at the drone's camera (t_{DC})
Frame encoded to encoded frame
Encoded frame packetized and transmitted over the air
Frame received at the radio station
Error detected in a packet
Re-transmission timeout
Frame retrans-mitted
Frame received at the radio station
Frame received at the FPV screen
Frame decoded (t_{FrVis})
Retransmission timeout interval T_O
Frame latency

FIGURE 3. Components of frame latency with a single re-transmission.

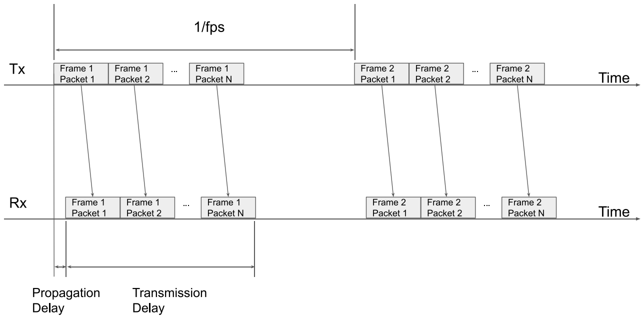


FIGURE 4. All packets representing a frame are transmitted back to back, without waiting for individual packet acknowledgements. Re-transmissions of unsuccessful packet transmissions, if any, are performed after all packets of that frame are transmitted.

It is assumed that all the packets representing a frame are transmitted back to back, and Fig. 4 shows this packet transmission mechanism, where the packet propagation delay, over the air, needs to be accounted for only once rather than for each packet. The video transmitter waits for a timeout interval T_O before re-transmitting any packets that were not acknowledged, with a maximum of $n_{MRT}(t)$ such re-transmission attempts. The next frame is transmitted after $n_{MRT}(t)$ attempts or earlier, if and when all the packets of the current frame are successfully acknowledged. To minimize chances of collision in certain scenarios, the collision avoidance algorithm may also choose not to perform re-transmissions, in which case packets of the latest encoded frame are transmitted as soon as they are available, without waiting for any acknowledgements. We model this re-transmission process as a Markov decision process (MDP) and derive the probability distribution of frame latency.

The re-transmission process can be modelled as an MDP with states denoted as s_i, where state s_i corresponds to i packets of the current frame being transmitted in the next transmission attempt, where $i \in [0, n_{\mathrm{Pkt}}(t)]$. The next state for s_i is the set of all states s_j such that $j \leq i$. Transition from state i to state j implies that, out of the i packets transmitted, j packets were in error. Further, transition to state s_0 implies that all packets from the previous state were successfully transmitted. Fig. 5 shows selected states and transitions for this MDP. The transition probability from any state s_i to s_j, denoted as $\mathbb{P}_{i,j}$ can be calculated as below:

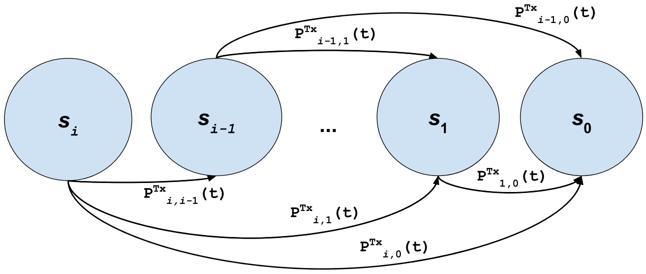


FIGURE 5. For the frame transmission Markov decision process, selected states and associated valid transitions are shown. Transition probabilities are denoted above the transitions.

$$\mathbb{P}_{i,j}(t) = \binom{i}{j}(e(t))^j(1-e(t))^{i-j}\ , \tag{15}$$

where $e(t)$ is the packet error rate in (11). We note that the transmission process of a frame ends after a maximum of $n_{MRT}(t)$ such transitions, or after reaching state s_0. Then, packets of the next frame are transmitted. The time interval after which the MDP transitions from a state s_i to *any* other state is calculated as:

$$T_i(t) = T_O + \frac{i(n_P(t) + N_{Ov})}{r(t)}. \tag{16}$$

For simplicity, we assume that all packets are of the same size. Let $\mathcal{S}$ denote an ordered set of MDP states that could be encountered in the successful transmission of this frame, where successful frame transmission implies that this ordered set ends at state s_0. We term $\mathcal{S}$ as a successful transmission sequence. Let $\mathcal{S}[k]$ identify the k^{th} state in this sequence. The probability that a given successful transmission sequence, $\mathcal{S}$, occurs is calculated as:

$$\mathbb{P}_{\mathcal{S}} = \prod_{k=1}^{|\mathcal{S}|} \mathbb{P}_{\mathcal{S}[k],\mathcal{S}[k+1]}, \tag{17}$$

and the frame latency associated with this transmission sequence is calculated as:

$$T_{\mathcal{S}}(t) = T_{\mathrm{ENC}} + \sum_{i=1}^{|\mathcal{S}|} T_i(t) + T_{\mathrm{P}}(t) + T_{\mathrm{N}} + T_{\mathrm{DEC}}, \quad (18)$$

Let $\mathcal{S}_\alpha$ denote the set of all such successful transmission sequences, where $\mathcal{S}_\alpha[m]$ identifies the m^{th} sequence. This set can be calculated by starting from the initial state, and enumerating through each possible next state until the maximum number of re-transmission attempts are reached, building transmission sequences in the process, one state at a time. For the final re-transmission attempt, s_0 is assumed to be the only valid next state for each transmission sequence, to ensure that only successful transmission sequences are generated. The PDF of frame latency is computed as follows:

$$\begin{aligned}&\mathbb{P}(T_{\mathrm{FrL}}(t) = T_{\mathcal{S}}(t)) = \mathbb{P}_{\mathcal{S}}(t), \ \forall\, \mathcal{S} \in \mathcal{S}_\alpha \\ &\mathbb{P}(T_{\mathrm{FrL}}(t) = \infty) = 1 - \sum_{m=1}^{|\mathcal{S}_\alpha|} \mathbb{P}_{\mathcal{S}_\alpha[m]}, \end{aligned} \quad (19)$$

where $T_{\mathrm{FrL}} = \infty$ represents the case where all packets of a frame are not successfully transmitted.

Let the cumulative distribution function (CDF) of frame latency be denoted as $\mathbb{F}_{\mathrm{FrL}}(t, \tau)$, which can be written as:

$$\mathbb{F}_{\mathrm{FrL}}(t, \tau) = \mathbb{P}(T_{\mathrm{FrL}}(t) < \tau), \quad (20)$$

where $\mathbb{F}_{\mathrm{FrL}}(t, \tau)$ represents the probability that a frame that is captured from the drone's CCD at instant t experiences a frame latency less than τ, and is hence displayed at the FPV screen before time instant $t + \tau$. The computational complexity of calculating this probability distribution is $\mathcal{O}(n_{\mathrm{Pkt}}(t)^{n_{\mathrm{MRT}}(t)})$. This distribution can be pre-computed offline as a function of possible channel SNR values, video resolutions, re-transmission attempts, and modulation schemes, and then a look-up table may be used during online video adaptation. In this assumed model, the distribution of frame latency varies with time as per the channel conditions, with the major source of randomness being the variability in the transmission time, which depends on the instantaneous transmission data rate and error rate. T_{FrL} represents the latency of a frame from the time instant at which it is captured at the drone's camera to the time instant it is visible at the FPV screen, and captures the *freshness* of the feed, i.e., the metric represents the delay between the stream of images visible in the FPV screen and stream of images as seen by the UAV's camera. The image that the pilot sees at the FPV screen is guaranteed to have been captured at the UAV's camera no later than $T^{\star}_{\mathrm{FrL}}$ seconds ago with a probability of $\mathbb{F}_{T_{\mathrm{FrL}}}(T^{\star}_{\mathrm{FrL}})$.

D. OBJECT RECOGNITION MODEL

After presenting the models for frame encoding, transmission, and reception, we present the model for obstacle recognition in the decoded frame, assuming a human visual system. We assume that the FPV pilot defines a parameter, H_{MinO}, which is the minimum dimension of the obstacle to be detected and avoided. Given the focal length of the UAV's camera, L_{Foc}, and the height of the camera's sensor, H_{Sen}, the height of the obstacle in pixels, denoted as $h_{\mathrm{O}}(t)$, in the decoded frame is calculated as:

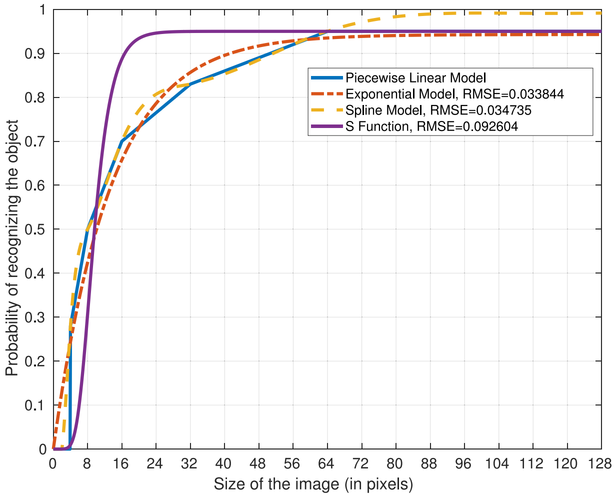


FIGURE 6. Various fits to the reference model of $\mathbb{P}_{\mathrm{OR}}(t)$, the probability of recognizing the object as a function of the size of the object in pixels.

$$h_{\mathrm{O}}(t) = \frac{H_{\mathrm{MinO}} \times h_{\mathrm{Fr}}(t) \times L_{\mathrm{Foc}}}{H_{\mathrm{Sen}} \times d_{\mathrm{O}}(t)}. \quad (21)$$

We denote the probability of recognizing the obstacle from the frame captured at instant t as $\mathbb{P}_{\mathrm{OR}}(t)$. The variation in $\mathbb{P}_{\mathrm{OR}}(t)$ as a function of the dimension of the obstacle ($h_{\mathrm{O}}(t)$) was studied in [38], wherein a piece-wise linear model was used to fit the data. This is shown in Fig. 6 as the solid plot. As this plot has sharp discontinuities, we evaluated other smoother options to fit the data: a spline function, an exponential function, and the Richard's curve [39].

$$\mathbb{P}_{\mathrm{OR}}(t) = \frac{1}{\left(1 + 0.001 e^{-0.35(h_{\mathrm{O}}(t) - 8.35)}\right)^{1000}}. \quad (22)$$

Using this Richard's curve fit, the variation in $\mathbb{P}_{\mathrm{OR}}$ with distance for various obstacle dimensions is shown in Fig. 7. Full high definition (HD) image frames were assumed with $h_{\mathrm{Fr}} = 1080$ pixels, the focal length was assumed to be 0.021 m and the height of the sensor as 0.0045 m.

IV. COLLISION AVOIDANCE

Building on the theoretical foundation developed in Section III, in this section we calculate the instantaneous probability of the UAV avoiding collision with an obstacle, which we denote as $\mathbb{P}_{\mathrm{AC}}(t)$. We then provide insight on the relation between all relevant variables, and how they may be tuned to minimize the probability of collision.

A. COLLISION AVOIDANCE REGIONS

After the frame is displayed on the FPV screen at time instant t, the pilot takes a non-negligible amount of time to process the image, recognize the obstacle, and to respond,

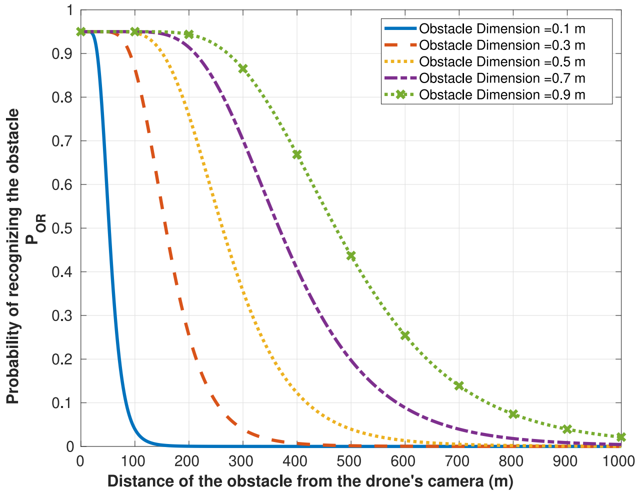


FIGURE 7. Probability of recognizing the object as a function of distance from the UAV's camera, when using a 1080p video stream, and the Richard's curve fit. Plots are shown for various obstacle dimensions.

if needed, by pushing the appropriate commands on the RF controller. Let this be time duration be denoted as T_{VisResp}. After a duration equal to the propagation delay, the command reaches the UAV, which then executes it. Since the command is transmitted at time instant $t + T_{\text{VisResp}}$, the propagation delay is $T_{\text{P}}(t{+}T_{\text{VisResp}})$. In this time duration $T_{\text{VisResp}}{+}T_{\text{P}}(t{+}T_{\text{VisResp}})$, the distance traveled by the UAV is given by:

$$\int_{t}^{t+T_{\text{VisResp}}+T_{\text{P}}(t+T_{\text{VisResp}})} \mathbf{v}_{\text{D}}(t)\text{d}t. \tag{23}$$

For the duration taken for the command to reach the UAV after it has been transmitted, and hence, also for the entire duration of the integral, $T_{\text{P}}(t + T_{\text{VisResp}})$ is assumed to be constant. Let $d_{\text{BD}}(t)$ denote the braking distance, and let $\mathbf{a}_{\text{BD}}$ denote the braking deceleration, where the braking distance is the distance traveled by the UAV from the position where it starts decelerating at the rate of $\mathbf{a}_{\text{BD}}$ to the position where it comes to a complete stop. Then, the braking distance can be calculated using the equations of motion as:

$$d_{\text{BD}}(t) = \frac{\mathbf{v}_{\text{D}}^2(t)}{2\mathbf{a}_{\text{BD}}}. \tag{24}$$

Based on the above discussion, if at time instant t the distance of the UAV to the obstacle is below a threshold, i.e., if

$$d_{\text{O}}(t) < d_{\text{BD}}(t) + \int_{t}^{t+T_{\text{VisResp}}+T_{\text{P}}(t+T_{\text{VisResp}})} \mathbf{v}_{\text{D}}(t)\text{d}t, \tag{25}$$

then the UAV cannot avoid collision just by braking. If the UAV's environment allows it, the pilot may still be able to avoid collision by swerving around the obstacle. However in this work, we assume that it is necessary for the pilot to keep all obstacles outside the braking distance of the UAV to guarantee safety. We term the region as defined by (25) as the *collision inevitable* region. Then, it follows that the *collision avoidable* region is the region where

$$d_{\text{O}}(t) > d_{\text{BD}}(t) + \int_{t}^{t+T_{\text{VisResp}}+T_{\text{P}}(t+T_{\text{VisResp}})} \mathbf{v}_{\text{D}}(t)\text{d}t. \tag{26}$$

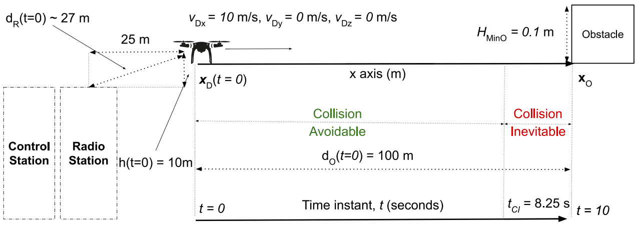


FIGURE 8. The scenario considered as an example to illustrate the collision avoidance regions.

We also note that if the UAV has a non-zero acceleration, then the boundaries of these collision regions change with time. Further, if the UAV has a constant acceleration $\mathbf{a}_{\text{D}}(t)$, then the integral in (25) and (26) becomes:

$$\begin{aligned}&\int_{t}^{t+T_{\text{VisResp}}+T_{\text{P}}(t+T_{\text{VisResp}})} \mathbf{v}_{\text{D}}(t)\text{d}t \\ &\quad = \mathbf{v}_{\text{D}}\big(T_{\text{VisResp}} + T_{\text{P}}\big(t + T_{\text{VisResp}}\big)\big) \\ &\qquad + 0.5\mathbf{a}_{\text{D}}(t)\big(T_{\text{VisResp}} + T_{\text{P}}\big(t + T_{\text{VisResp}}\big)\big)^2. \end{aligned} \tag{27}$$

These collision regions are illustrated in Fig. 8 for a specific scenario, wherein an obstacle is at a distance of 100 m away from the UAV, and the UAV is traveling at a speed of 10 m/s towards the obstacle. t_{CI} denotes the time instant at which the UAV is at a distance of d_{BD} from the obstacle, such that collision is inevitable for all $t > t_{\text{CI}}$ if the UAV is moving towards the obstacle. Assuming a braking acceleration of 4 m/s^2 and a T_{VisResp} of 0.25 s, the braking distance is then 12.5 m and t_{CI} for this scenario is approximately 8.25 s, neglecting propagation delay for simplicity.

B. SAFETY METRICS

We derive the expression for $\mathbb{P}_{\text{AC}}(t)$ in this section. The probability of avoiding collision at time instant t can be calculated by considering the probability of recognizing the obstacle in the frame that is visible at the FPV screen at time t, and by calculating the probability of arrival of previously transmitted frames at the FPV screen by time instant t. It is assumed that the FPV screen displays the latest received frame. The expression for the probability of avoiding collision is derived as:

$$\begin{aligned}\mathbb{P}_{\text{AC}}(t) = \sum_{n=0}^{|T_{\text{Fr}}(t)|} \Bigg\{ &\mathbb{F}_{\text{FrLat}}\big(t = t_n^{\text{Fr}},\ \tau = t - t_n^{\text{Fr}}\big)\mathbb{P}_{\text{OR}}(t)\big(t = t_n^{\text{Fr}}\big) \\ &\times \prod_{m=n+1}^{|T_{\text{Fr}}(t)|} \big(1 - \mathbb{F}_{\text{FrLat}}\big(t = t_m^{\text{Fr}},\ \tau = t - t_m^{\text{Fr}}\big)\big)\Bigg\},\end{aligned} \tag{28}$$

for all $t < t_{\text{CI}}$.

All the previously transmitted frames are considered when calculating the probability of avoiding collision at any given time instant t. As such, the impact of an individual frame is embedded in expression (28). To guide the video adaptation algorithm in choosing the parameters for a particular frame, a measure of the contribution of an individual frame to avoiding collision is needed. Such a measure may be approximated as the probability of the event that the frame is received by the pilot while the UAV is in the collision avoidable region, and the pilot is able to recognize the obstacle in this received frame. Let this measure for the n^{th} frame be denoted as $\mathbb{P}_{\text{AC, n}}(t)$. For a frame read at time instant t, we first derive the time taken by the UAV to reach the boundary of the *collision avoidable* and *collision inevitable* regions assuming that the UAV continues at moving with velocity $\mathbf{v}_{\text{D}}(t)$ and acceleration $\mathbf{a}_{\text{D}}(t)$. Let this time interval be denoted as $T_{\text{C,n}}$ for the n^{th} frame. The assumption on UAV's motion may be removed if the UAV has a priori knowledge of its trajectory.

Then, the distance of the UAV from the obstacle at time instant $t + T_{\text{C,n}}$ can be calculated as:

$$d_{\text{O}}\big(t + T_{\text{C,n}}\big) = d_{\text{O}}(t) - \Big(\mathbf{v}_{\text{D}}(t)T_{\text{C,n}} + 0.5\mathbf{a}_{\text{D}}(t)T_{\text{C,n}}^2\Big), \quad (29)$$

while the UAV's velocity at the same time instant is:

$$\mathbf{v}_{\text{D}}\big(t + T_{\text{C,n}}\big) = \mathbf{v}_{\text{D}}(t) + \mathbf{a}_{\text{D}}(t)T_{\text{C,n}}. \quad (30)$$

At this time instant of $t + T_{\text{C,n}}$, the UAV is at the braking distance from the obstacle, i.e.,

$$d_{\text{O}}\big(t + T_{\text{C,n}}\big) = \frac{\mathbf{v}_{\text{D}}^2\big(t + T_{\text{C,n}}\big)}{2\mathbf{a}_{\text{BD}}} + \int_t^{t+T_{\text{VisResp}}+T_{\text{P}}\left(t+T_{\text{C,n}}+T_{\text{VisResp}}\right)} \times \mathbf{v}_{\text{D}}\big(t + T_{\text{C,n}}\big)\mathrm{d}t. \quad (31)$$

$T_{\text{C,n}}$ may be calculated by substituting the expression for $d_{\text{O}}\ (t + T_{\text{C,n}})$ from (31) in (29), and by using the expression for $\mathbf{v}_{\text{D}}(t + T_{\text{C,n}})$ given in (30). This results in the quadratic equation (32), shown at the bottom of the page, the roots of which yield $T_{\text{C,n}}$.

The probability that the n^{th} frame is received within the collision avoidable region is $\mathbb{F}(t, T_{\text{C,n}})$. By incoporating the probability of recognizing the obstacle in the n^{th} frame, we can mathematically approximate the contribution of this n^{th} frame to avoiding collision as:

$$\mathbb{P}_{\text{AC, n}}(t) = \mathbb{F}\big(t = t_n^{\text{Fr}}, \tau = T_{\text{C,n}}\big) \times \mathbb{P}_{\text{OR}}\big(t = t_n^{\text{Fr}}\big). \quad (33)$$

When a constraint on frame latency is specified, such that the latency of FPV feed is required to be less than T_{FrL}^* with a guarantee of K_{G}, then the contribution of an individual frame to avoiding collision is calculated as:

$$\mathbb{P}_{\text{AC, n}}(t) = \mathbb{F}\big(t = t_n^{\text{Fr}}, \tau = T_{\text{FrL}}^*\big) \times \mathbb{P}_{\text{OR}}\big(t = t_n^{\text{Fr}}\big). \quad (34)$$

To meet this constraint, the FPV feed has to be adapted such that $\mathbb{F}_{\text{FrL}}(t, T_{\text{FrL}}^*) > K_{\text{G}}$.

After deriving the probability of avoiding collision at a specific time instant t, and also the contribution of an individual frame to avoiding collision, we derive a more holistic metric calculated over the entire trajectory of the UAV. This metric represents the degree of safety over a given period of time, and may be used to evaluate and compare various video adaptation strategies along a given trajectory or to evaluate a particular adaptation strategy along various trajectories. We term this metric as the *degree of safety* and denote it as σ. σ may be calculated by integrating $\mathbb{P}_{\text{AC}}(t)$ over the evaluation period and then by normalizing it as:

$$\sigma = \frac{\int_{t=0}^{t_{\text{END}}} \mathbb{P}_{\text{AC}}(t)\ dt}{\int_{t=0}^{t_{\text{END}}} 1\ \text{dt}}, \quad (35)$$

where the evaluation period is from $t = 0$ to $t = t_{\text{END}}$. As the human visual system can parallely process and distinguish between images that are displayed 0.1 seconds apart [40], the continues integration in (35) may be converted into a discrete summation, wherein $\mathbb{P}_{\text{AC}}(t)$ is calculated at intervals of 0.1 seconds.

C. NUMERICAL RESULTS RELATED TO SAFETY METRICS

In this section we show, numerically, the influence of various parameters on the probability of collision. We also show their influence on the safety metric in (35).

The graph in Fig. 9 shows the dependency amongst variables. The safety metrics, $\mathbb{P}_{\text{AC}}(t)$ and σ, only depends on three components: the distribution of frame latency $T_{\text{FrL}}(t)$, the probability of recognizing the obstacle in the frame $\mathbb{P}_{\text{OR}}(t)$, and the UAV's velocity $\mathbf{v}_{\text{D}}(t)$. These three components capture the effects of wireless FPV transmission, human visual perception, and UAV motion respectively.

We study the variation in the instantaneous probability of avoiding collision and the degree of safety for the scenario shown in Fig. 8. Relevant parameters for the scenario are listed in Table 3. The UAV is assumed to be at a horizontal distance of 50 m from the radio station, and at an altitude of 10 m at $t = 0$ s, traveling away from the radio station and towards the obstacle in a straight line. Also, the UAV is at a distance of 100 m from the obstacle, i.e., $d_{\text{O}}(t = 0) = 100$ m.

$$T_{\text{C,n}}^2\left(\frac{\mathbf{a}_{\text{D}}^2(t)}{2\mathbf{a}_{\text{BD}}} + 0.5\mathbf{a}_{\text{D}}\right) + T_{\text{C,n}}\left(\frac{\mathbf{v}_{\text{D}}(t)\mathbf{a}_{\text{D}}(t)}{\mathbf{a}_{\text{BD}}} + \mathbf{v}_{\text{D}} + \mathbf{a}_{\text{D}}(t)\big(T_{\text{VisResp}} + T_{\text{P}}\big(t + T_{\text{C,n}} + T_{\text{VisResp}}\big)\big)\right) + \left(\frac{\mathbf{v}_{\text{D}}^2(t)}{2\mathbf{a}_{\text{BD}}} + \int_t^{t+T_{\text{VisResp}}+T_{\text{P}}\left(t+T_{\text{C,n}}+T_{\text{VisResp}}\right)} \mathbf{v}_{\text{D}}\big(t + T_{\text{C,n}}\big)\mathrm{d}t - \mathrm{d}_{\text{O}}(\mathrm{t})\right) = 0 \quad (32)$$

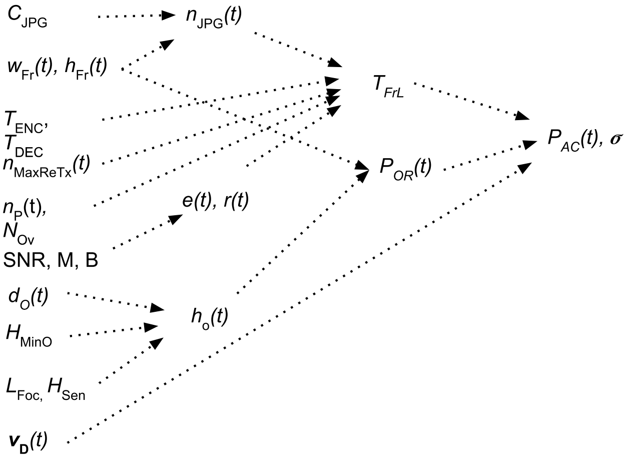


FIGURE 9. A graph showing the dependency among the variables. A relationship between two variables is represented as a dotted line between them, with the arrow at the dependent variable and the tail at the independent.

TABLE 2. Simulation parameters.

Name	Variable	Value
Frame Resolution	$h_{\text{Fr}}(t)$	240, or 360, or 480, or 720, or 1080, or 1440, or 2160 pixels.
Focal length	L_{Foc}	0.021 m
Sensor height	H_{Sen}	0.0045 m
Minimum obstacle dimension to avoid	H_{MinO}	0.1 m
Initial distance to obstacle	$d_{\text{O}}(0)$	100 m
UAV height	$h(t)$	10 m, for all t
Distance of UAV from the radio station	$d_{\text{R}}(0)$	$\sqrt{10^2+25^2} \approx 27\ m.$
UAV velocity	$\mathbf{v}_{\text{D}}(t)$	$v_{\text{Dx}}(t)$ = 5 to 20 m/s in steps of 2.5 m/s, $v_{\text{Dy}}(t)$ = 0 m/s, $v_{\text{Dz}}(t)$ = 0 m/s
UAV braking acceleration	$\mathbf{a}_{\text{B}}$	4 m/s^2
Number of retransmission attempts	$n_{\text{MRT}}(t)$	0 (Transmit the latest available frame without waiting for ACKs)
Frame encoding delay	T_{ENC}	0.05 s
Frame decoding delay	T_{DEC}	0.05 s
Re-transmission timeout interval	T_{O}	0.25 s
Network delay	T_{N}	0.1 s
Packet size	$n_{\text{P}}(t)$	2132 bytes
Modulation order	$M(t)$	2, corresponding to 2-QAM
Signal bandwidth	B	10 MHz
JPEG compression factor	C_{JPG}	0.1
Frame colour depth	N_{Color}	24 bits

To account for the randomness in the communication channel due to Rician fading, simulations were performed for 50 runs. The mean values of $\mathbb{P}_{\text{AC}}(t)$, across these runs, were used in Figs. 10–12.

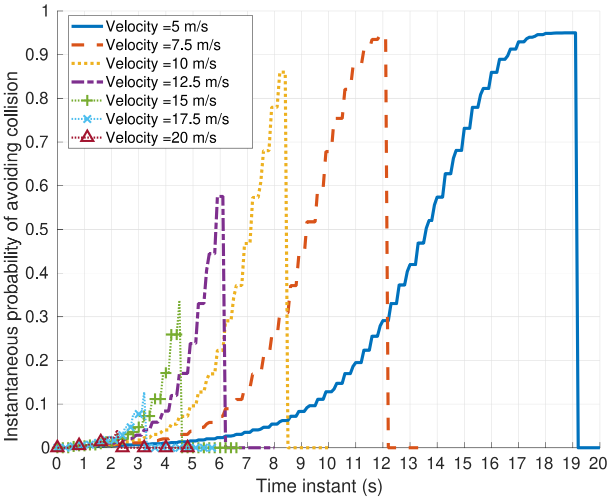


FIGURE 10. Variation in the instantaneous probability of avoiding collision at different velocities at a resolution of 720p, and an initial distance to obstacle of 100 m.

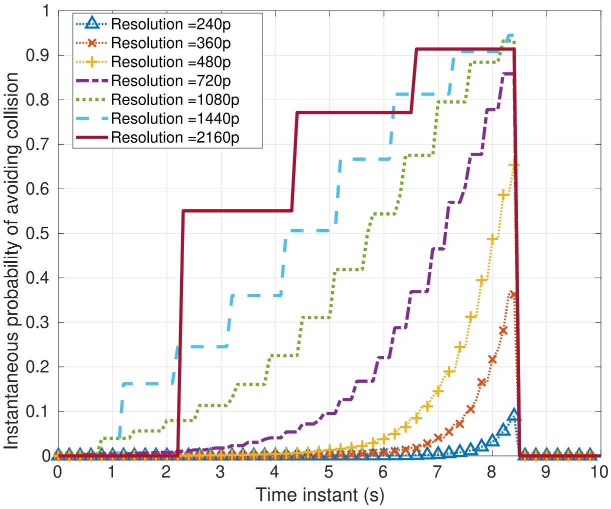


FIGURE 11. Variation in the instantaneous probability of avoiding collision at different resolutions for a velocity of 10 m/s, and an initial distance to obstacle of 100 m.

Fig. 10 shows the variation of the instantaneous probability of avoiding collision with the UAV's velocity when the video resolution is 720p. At $t = 0$, the size of the obstacle in the FPV image is too small to be recognized by the pilot. As the UAV approaches the obstacle, the size of the obstacle in the FPV image becomes larger, increasing the probability of the obstacle being recognized ($\mathbb{P}_{\text{OR}}(t)$). This increase in $\mathbb{P}_{\text{AC}}(t)$ can be seen from Fig. 10. Once the UAV is in the *collision inevitable* region, $\mathbb{P}_{\text{AC}}(t) = 0$ as can be seen by the abrupt drop in the curves at the time instants corresponding to the velocities. Hence, the FPV pilot has a limited period of time in which to recognize the obstacle, as represented by the time instants for which $\mathbb{P}_{\text{AC}}(t) > 0$. It can also be seen that, at higher velocities, the FPV pilot has less time to recognize the obstacle before the obstacle is closer than the braking distance. For instance, when $v_{\text{D}} \geq 17.5\ m/s$, the

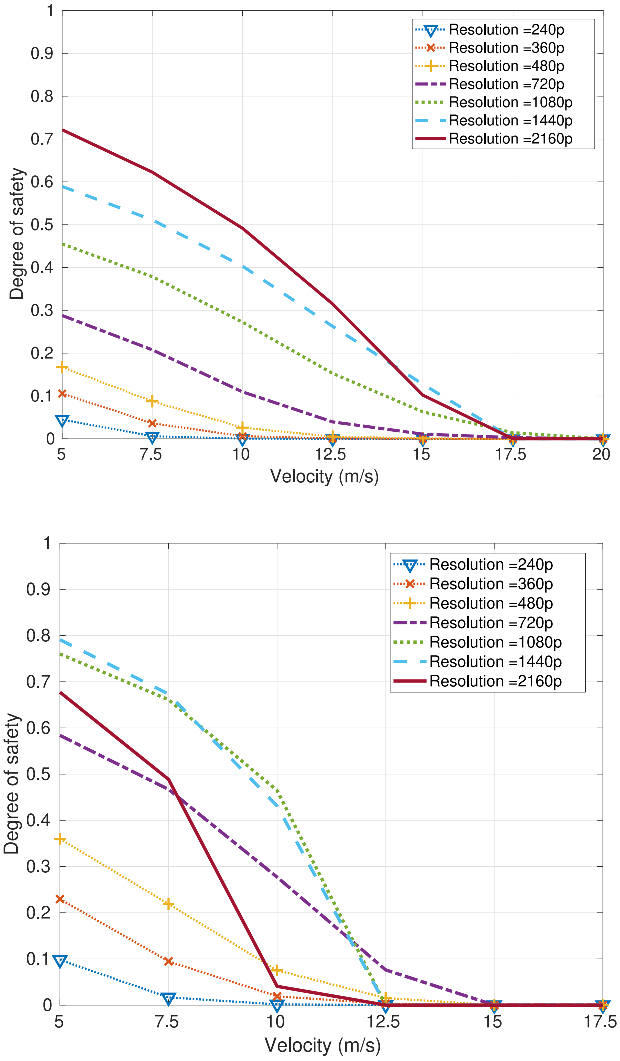


FIGURE 12. Variation in degree of safety with velocity at different video resolutions, when the obstacle is at a distance of 100 m (a) or 50 m (b).

latency of the 720p resolution frame is such that FPV pilot never gets a chance to recognize the obstacle, i.e., the UAV is in the *collision inevitable* region before a suitable frame is received at the FPV screen, from which the obstacle could be recognized.

Fig. 11 shows the variation in $\mathbb{P}_{AC}(t)$ with video resolution, for a velocity of 10 m/s. In general, the obstacle is recognizable at the FPV screen from farther away when the resolution is higher. At resolutions that are lower, the obstacle is visible only when the UAV is closer to the obstacle. For this scenario, at resolutions lower than 240p the obstacle is never visible in any frame and the video resolution should be at-least 360p if the pilot is to have any chance of seeing the obstacle in a frame. The obstacle is visible in the 1440p (Quad HD) and 1080p (Full HD) frames at an earlier time instant and at distances farther away, than in the 2100p (4K) frame. This can be explained by taking into account the lower frame size, lower frame transmission time, and hence, lower frame latency at these resolutions. It is also observed that the curves for higher resolutions are more jagged. This is again due to the fact that higher resolutions result in larger latencies. So, $\mathbb{P}_{AC}(t)$ does not increase between successive time instants as the latest captured frame has not been received yet at the FPV screen.

The variation in the degree of safety with velocity at different video resolutions is plotted in Fig. 12(a) when the obstacle is at a distance of 100 m from the UAV and in Fig. 12(b) for a distance of 50 m. It can be seen that the highest available resolution is always optimum when the obstacle is farther away, at 100 m. Interestingly, at a velocity of 15 m/s and an initial distance to obstacle of 100 m, a resolution of 1440p performs better than a resolution of 2160p. The lower resolution allows the FPV pilot to recognize the obstacle earlier due to lower latency, and the higher UAV velocity reduces any gains achieved due to the higher resolution, i.e., the UAV reaches the collision inevitable region quickly such that the higher latency of 2160p is undesirable. When the obstacle is closer at a distance of 50 m, lower resolutions than 2160p are always optimum on account of lower transmission times that allow the FPV pilot to view the obstacle earlier, as is also shown in Fig. 11. Lower latencies also result in the FPV feed being *fresher*, i.e., the FPV screen is more frequently updated.

V. FPV VIDEO ADAPTATION

In this section, we mathematically formulate the FPV video adaptation problem and present a solution. The FPV video adaptation at the can be modelled as a discrete time decision process, which takes place at the *FPV controller* in the UAV. The decision that the controller makes is to choose the resolution, the number of re-transmission attempts, and the modulation scheme for each frame read from the UAV's camera, such that the probability of avoiding collision is maximized. The time steps of this decision process are as decided by the inter-frame intervals, as expressed in (6).

In this section we present a simple video adaptation strategy. If a latency constraint is not provided, the algorithm chooses the action that maximizes the probability of recognizing the obstacle in the current image frame, before the UAV reaches the collision inevitable region, assuming that the UAV continues on its current trajectory. Each possible action is evaluated in a greedy manner to choose the action that maximizes the metric derived in (33). If a latency constraint is provided, the algorithm chooses the action that maximizes the probability of recognizing the obstacle in the current image frame, while ensuring that the frame reaches the FPV screen within the specified latency at the specified guarantee. In this case, the algorithm optimizes the metric derived in (34). This search is also performed in a greedy manner. Optimizing the metric derived in (33) is only an indirect way of optimizing the degree of safety metric derived in (35), since it maximizes the contribution of the current frame to avoiding collision, rather than attempting to maximize future reward by considering all possible sets of

TABLE 3. Video adaptation parameters.

FPV Parameter	Values
Video resolution	240p, 360p, 480p, 720p, 1080p, 1440p
Re-transmission attempts	0, 1, 2
Modulation order	2, 4, 8
Frame latency constraints	0.3 s, 0.7 s
Frame latency guarantee	95%

future actions. On the other hand, this approach of maximizing the current reward ensures that knowledge of the UAV trajectory, obstacle locations, and channel conditions in the future is not required. The algorithm can hence adapt to changes in channel conditions, UAV trajectory, and obstacle locations. The decision process is mathematically expressed as follows. For frame-n read from the UAV's camera at time instant t:

$$\max_{h_{\mathrm{Fr}}(t), n_{\mathrm{MRT}}(t), M(t)} \mathbb{P}_{\mathrm{AC},n}(h_{\mathrm{Fr}}(t), n_{\mathrm{MRT}}(t), M(t)), \quad (36)$$
$$\text{such that, } \mathbb{F}_{\mathrm{FrL}}\left(t, T^{*}_{\mathrm{FrL}}\right) > K_{\mathrm{G}},$$

where the latency constraint may or may not be specified. We denote the action space by $\mathcal{A}$, where,

$$\mathcal{A} = \{h_{\mathrm{Fr}}(t), n_{\mathrm{MRT}}(t)M(t)\}. \quad (37)$$

The size of the action space is denoted by $|\mathcal{A}|$, i.e., this size represents the number of possible combinations of video resolutions, re-transmission attempts, and modulation schemes. The computational complexity of this FPV adaptation algorithm is $\mathcal{O}(|\mathcal{A}|)$.

A. NUMERICAL RESULTS RELATED TO VIDEO ADAPTATION

The results using the video adaptation algorithm described above are presented for a specific scenario, similar to that described in Section IV-C wherein a UAV is approaching an obstacle at a constant velocity. The parameters of the scenario remain the same as those listed in Table 3. The algorithm was configured to provide maximum frame latencies of 0.3 s and 0.7 s each with a guarantee of 95%. Additionally, the algorithm was also run without any constraint on latency. The adaptation algorithm was allowed to vary the number of re-transmissions from 0 to 2, the video resolution from 240p to 1440p, and the modulation order as either 2-QAM, 4-QAM, or 8-QAM. The size of the action, for the assumed values of parameters, is $6 \times 3 \times 3 = 54$. These parameters are listed in Table 3. To measure the performance gain of our proposed video adaption algorithm with reference to non-adaptive approaches, the instantaneous probability of avoiding collision was also calculated at constant video resolutions of 1440p, 1080p, 720p, and 480p assuming no re-transmissions. We provide the MATLAB source code, which we developed for our simulations, in a public GitHub repository [41]. This code implements our theoretical framework, calculates the safety metric, and implements the FPV adaptation algorithm.

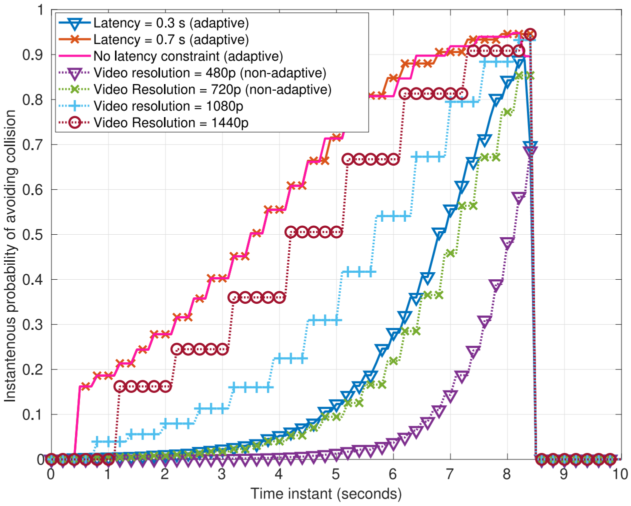


FIGURE 13. Instantaneous probability of avoiding collision, using the video adaptation algorithm, when the obstacle is at an initial distance of 100 m from the UAV. Results with non-adaptive algorithms are also shown for comparison.

The results are shown in Fig. 13, which plots the instantaneous probability of avoiding collision. At a latency constraint of 0.3 s, the adaptation algorithm is forced to choose lower resolutions to meet the constraint, which do not allow the pilot to view the obstacle from far away. At a higher latency constraint 0.7 s and also when no latency constraint is provided, higher resolutions are possible, allowing the pilot to view the obstacle from farther away. It can also be seen that, at higher resolutions of 1440p and 1080p, the plots are more jagged than when using 720p or 480p, i.e., the probability of collision stays constant for a larger duration than when using 720p or 480p and the FPV screen is updated less frequently. This implies that the resulting FPV feed, when using 720p or 480p, is *fresher*, than when using 1080p or 1440p, though the probability of recognizing the obstacle is lower in each frame. The same observation is true when comparing the results with a higher latency constraint of 0.7 s (fewer updates of the FPV screen) and a lower constraint of 0.3 s (more frequent updates). The instantaneous probability of avoiding collision, with a latency constraint of 0.7 s. is very close to that without any constraint, at all time instants. This also implies that collision may be avoided satisfactorily while ensuring, with a 95% guarantee, that the frame latency never exceeds 0.7 s. The mean degree of safety, across 10 runs of the scenario, was 0.4863 when no latency constraint was provided, 0.4884 with a latency constraint of 0.7 s, 0.1133 with a latency constraint of 0.3 s, 0.3848 with a constant resolution of 1440p, 0.2551 with a constant resolution of 1080p, 0.0992 with a constant resolution of 720p, and 0.0221 with a constant resolution of 480p. In this scenario, in terms of maximizing the probability of avoiding collision, the adaptive FPV algorithm has a gain of at-least 26.8% over all other non-adaptive approaches that we consider. The instantaneous probability of avoiding collision when the obstacle is at an initial distance of

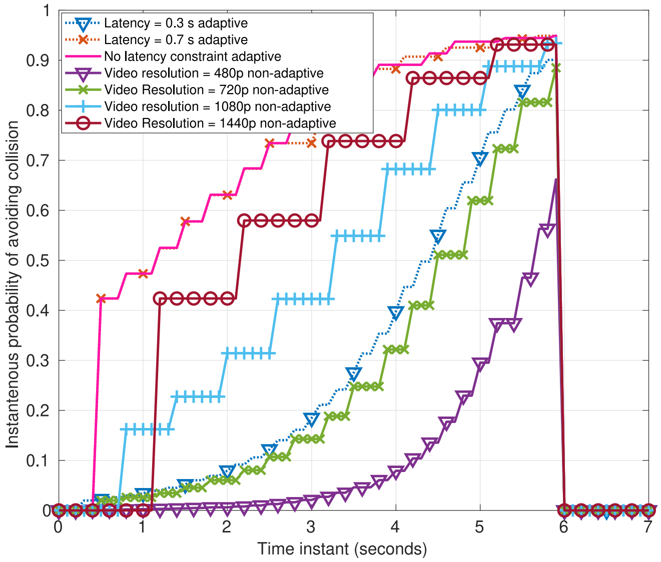


FIGURE 14. Instantaneous probability of avoiding collision, using the video adaptation algorithm when the obstacle is at an initial distance of 75 m from the UAV. Results with non-adaptive algorithms are also shown for comparison.

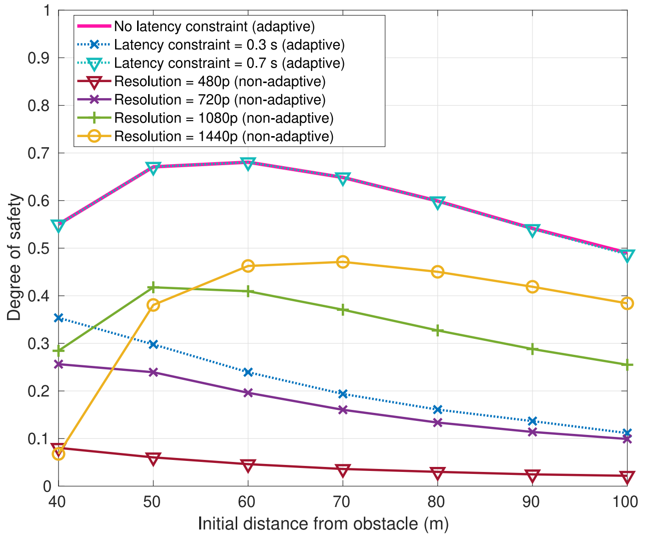


FIGURE 15. Variation in the degree of safety with initial distance to the obstacle, for adaptive and non-adaptive algorithms.

75 m, is shown in Fig. 14. In this case, the adaptive FPV algorithm has a gain of at-least 35.19% over other non-adaptive algorithms in terms of avoiding collision. Fig. 15 shows compares the degree of safety achieved by adaptive and non-adaptive algorithms, for multiple initial distances to the obstacle, from 40 m to 100 m. The performance of the adaptive algorithm was 58.63% higher than the closest non-adaptive one, averaged over all the considered distances.

VI. CONCLUSION

In this work, a theoretical model was presented to predict the probability of a UAV colliding with the nearest obstacle based on the current video feed and the communication channel conditions. This model was then used to adapt the FPV feed in real time to avoid collision. Along with maximizing the probability of avoiding collision, this video adaptation algorithm can also be configured to meet specific latency constraints of the application. The performance of this algorithm was evaluated to meet various specified latency constraints, and its ability to maintain a proper trade-off between video resolution, re-transmission attempts, and modulation scheme was demonstrated. It was found that, for the considered scenarios, the performance of the adaptive algorithm was, on an average, 58.63% higher than the closest non-adaptive one in terms of maximizing the probability of avoiding collision. Such video adaptation strategies can be used to increase the safety of UAV FPV applications.

ACKNOWLEDGMENT

A portion of this work has been carried out during Simran Singh's internship at AT&T Labs, Bedminster, NJ, USA, from Fall 2019 through Summer 2020.

REFERENCES

[1] (Mar. 2020). *Federal Aviation Administration: Unmanned Aircraft Systems (UAS) Traffic Management (UTM) Concept of Operations*. [Online]. Available: https://www.faa.gov/uas/research_development/traffic_management/media/UTM_ConOps_v2.pdf

[2] A. Bhuyan *et al.*, "Secure 5G network for a nationwide drone corridor," in *Proc. IEEE Aerosp. Conf.*, 2021, pp. 1–10.

[3] S. Singh, S. L. Sunkara, İ. Güvenç, A. Bhuyan, H. Dai, and M. L. Sichitiu, "Spectrum reuse among aerial and ground users in mmWave cellular networks in urban settings," in *Proc. IEEE 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2020, pp. 1–6.

[4] S. Singh, H. Narasimhan, İ. Güvenç, A. Bhuyan, H. Dai, and M. L. Sichitiu, "Coverage analysis for ground and aerial users in mmWave cellular networks in urban settings," in *Proc. SoutheastCon*, Raleigh, NC, USA, 2020, pp. 1–8.

[5] "Enhanced LTE support for aerial vehicles," 3GPP, Sophia Antipolis, France, 3GGP Rep. TR 36.777, Feb. 2021. [Online]. Available: ftp://www.3gpp.org/specs/archive/36_series/36.777

[6] DJI. (Oct. 2020). *DJI's Digital FPV System*. [Online]. Available: https://www.dji.com/fpv

[7] S. Fouladi, J. Emmons, E. Orbay, C. Wu, R. S. Wahby, and K. Winstein, "Salsify: Low-latency network video through tighter integration between a video codec and a transport protocol," in *Proc. 15th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Baltimore, MD, USA, Aug. 2018, pp. 267–282.

[8] (Feb. 2021). *WebRTC*. [Online]. Available: https://webrtc.org/

[9] L. N. N. T. Ha, D. H. P. Bui, and S. K. Hong, "Nonlinear control for autonomous trajectory tracking while considering collision avoidance of UAVs based on geometric relations," *Energies*, vol. 12, no. 8, p. 1551, 2019.

[10] X. Xiao, W. Wang, T. Chen, Y. Cao, T. Jiang, and Q. Zhang, "Sensor-augmented neural adaptive bitrate video streaming on UAVs," *IEEE Trans. Multimedia*, vol. 22, no. 6, pp. 1567–1576, Jun. 2020.

[11] M. Naveed, S. Qazi, S. M. Atif, B. A. Khawaja, and M. Mustaqim, "SCRAS server-based crosslayer rate-adaptive video streaming over 4G-LTE for UAV-based surveillance applications," *Electronics*, vol. 8, no. 8, p. 910, 2019.

[12] Y. Zhang, C. Xu, I. A. Hemadeh, M. El-Hajjar, and L. Hanzo, "Near-instantaneously adaptive multi-set space-time shift keying for UAV-aided video surveillance," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12843–12856, Nov. 2020.

[13] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.

[14] K. McGuire, G. De Croon, C. De Wagter, K. Tuyls, and H. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1070–1076, Apr. 2017.

[15] Z. Gosiewski, J. Ciesluk, and L. Ambroziak, "Vision-based obstacle avoidance for unmanned aerial vehicles," in *Proc. Int. Congr. Image Signal Process.*, vol. 4. Shanghai, China, Nov. 2011, pp. 2020–2025.

[16] A. Al-Kaff, Q. Meng, D. Martín, A. de la Escalera, and J. M. Armingol, "Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Gothenburg, Sweden, Jun. 2016, pp. 92–97.

[17] A. Carrio, Y. Lin, S. Saripalli, and P. Campoy, "Obstacle detection system for small UAVs using ADS-B and thermal imaging," *J. Intell. Robot. Syst.*, vol. 88, no. 2, pp. 583–595, 2017.

[18] (Feb. 2021). *The Connex ProSight HD Vision Kit*. [Online]. Available: https://www.getfpv.com/the-connex-prosight-hd-vision-kit.html

[19] (Feb. 2021). *Insight SE 5G 1080p FPV Transmitter*. [Online]. Available: https://www.foxtechfpv.com/insight-se-5g-1080p-200mw-full-hd-digital-video-system.html

[20] W. G. Aguilar, V. P. Casaliglla, and J. L. Pólit, "Obstacle avoidance based-visual navigation for micro aerial vehicles," *Electronics*, vol. 6, no. 1, p. 10, 2017.

[21] P. Gao, D. Zhang, Q. Fang, and S. Jin, "Obstacle avoidance for micro quadrotor based on optical flow," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Chongqing, China, 2017, pp. 4033–4037.

[22] Y. Mao, M. Chen, X. Wei, and B. Chen, "Obstacle recognition and avoidance for UAVs under resource-constrained environments," *IEEE Access*, vol. 8, pp. 169408–169422, 2020.

[23] J. Sun, J. Tang, and S. Lao, "Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm," *IEEE Access*, vol. 5, pp. 18382–18390, 2017.

[24] L. A. Tony, D. Ghose, and A. Chakravarthy, "Avoidance maps: A new concept in UAV collision avoidance," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Miami, FL, USA, Jun. 2017, pp. 1483–1492.

[25] Y. Lin and S. Saripalli, "Sampling-based path planning for UAV collision avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3179–3192, Nov. 2017.

[26] Y. Watanabe, A. Calise, and E. Johnson, "Vision-based obstacle avoidance for UAVs," in *Proc. AIAA Guid. Navig. Control Conf. Exhibit.*, 2007, p. 6829.

[27] J. Seo, Y. Kim, S. Kim, and A. Tsourdos, "Collision avoidance strategies for unmanned aerial vehicles in formation flight," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 6, pp. 2718–2734, Dec. 2017.

[28] Y. Kuriki and T. Namerikawa, "Formation control with collision avoidance for a multi-UAV system using decentralized MPC and consensus-based control," *J. Control Meas. Syst. Integr.*, vol. 8, no. 4, pp. 285–294, 2015.

[29] M. Ille and T. Namerikawa, "Collision avoidance between multi-UAV-systems considering formation control using MPC," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Munich, Germany, Jul. 2017, pp. 651–656.

[30] S. Konatowski and P. Pawłowski, "Ant colony optimization algorithm for UAV path planning," in *Proc. Int. Conf. Adv. Trends Radioelecrtron. Telecommun. Comput. Eng. (TCSET)*, Lviv-Slavske, Ukraine, Feb. 2018, pp. 177–182.

[31] V. Patil, W. Van Gansbeke, D. Dai, and L. Van Gool, "Don't forget the past: Recurrent depth estimation from monocular video," 2020. [Online]. Available: arXiv:2001.02613.

[32] L. Sharpe, D. M. Ott, and C. Fleischhauer, "Library of congress manuscript digitization demonstration project: Final report, chapter 8," Library Congr., Washington, DC, USA, Rep., 1998.

[33] R. F. Haines, *The Effects of Video Compression on Acceptability of Images for Monitoring Life Sciences Experiments* (NASA Technical Paper 3239). Washington, DC, USA: Nat. Aeronaut. Space Admin., 1992.

[34] J. Niederst and J. N. Robbins, *Web Design in a Nutshell: A Desktop Quick Reference, Section 20.2*, vol. 2. Beijing, China: O'Reilly Media, Inc., 2001.

[35] W. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schneckenburger, "A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2361–2391, 3rd Quart., 2019.

[36] Z. Qiu, X. Chu, C. Calvo-Ramirez, C. Briso, and X. Yin, "Low altitude UAV air-to-ground channel measurement and modeling in semiurban environments," *Wireless Commun. Mobile Comput.*, vol. 2017, Nov. 2017, Art. no. 1587412. [Online]. Available: https://www.hindawi.com/journals/wcmc/2017/1587412/

[37] J. Lu, K. B. Letaief, J. C.-I. Chuang, and M. L. Liou, "M-PSK and M-QAM BER computation using signal-space concepts," *IEEE Trans. Commun.*, vol. 47, no. 2, pp. 181–184, Feb. 1999.

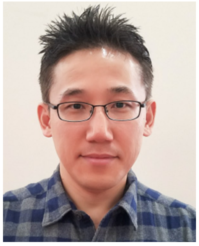
[38] A. Torralba, "How many pixels make an image?" *Vis. Neurosci.*, vol. 26, no. 1, pp. 123–131, 2009.

[39] G. J. Berry, R. J. Cawood, and R. G. Flood, "Curve fitting of germination data using the Richards function," *Plant Cell Environ.*, vol. 11, no. 3, pp. 183–188, 1988.

[40] C. Keysers, D.-K. Xiao, P. Földiák, and D. I. Perrett, "The speed of sight," *J. Cogn. Neurosci.*, vol. 13, no. 1, pp. 90–101, 2001.

[41] (Jun. 2021). *MATLAB Simulation Code for the Collision Avoidance Scenario and the UAV FPV Adaptation Algorithm*. [Online]. Available: https://github.com/ssinghjah/FPVForCollisionAvoidance

SIMRAN SINGH (Member, IEEE) received the B.Tech. degree in electronics from Veermata Jijabai Technological Institute, Mumbai, India, in 2011. He is currently pursuing the Ph.D. degree in electrical engineering from NCSU, Raleigh. His topic of research is wireless communications for reliable UAV operations.



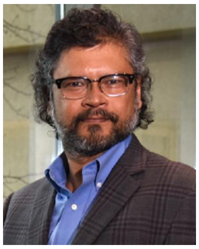
HEE WON LEE received the B.E. degree in electrical engineering from Korea University in 2002, the M.S.E. degree in software engineering from Carnegie Mellon University in 2005, and the Ph.D. degree in computer science from North Carolina State University in 2015. He was a member of Technical Staff with Korea Telecom from 2002 to 2009, and a Principal Inventive Scientist with AT&T Labs Research from 2015 to 2020. He is currently leading Cloud Platform Group as the Head of Group at Data and Information Technology Center, Samsung Electronics. His research interest lies in computer networking, storage systems, distributed systems, high-performance computing, wireless networking, mobile computing, edge computing, and machine learning.

TUYEN X. TRAN (Member, IEEE) received the B.Eng. degree (Hons. program) in electronics and telecommunications from the Hanoi University of Science and Technology, Vietnam, in 2011, the M.Sc. degree in electrical and computer engineering (ECE) from the University of Akron, Ohio, in 2013, and the Ph.D. degree in ECE from Rutgers University, NJ, USA, in 2018. He is currently a Senior Inventive Scientist with AT&T Labs Research, Bedminster, NJ, USA. His research interests include wireless communications, mobile cloud computing, and network optimization. He received the Best Paper Award at the IEEE/IFIP Wireless On-demand Network Systems and Services Conference in 2017 and the Outstanding Graduate Student Award from the Rutgers School of Engineering in 2018. He is a member of the ACM.

YU ZHOU received the B.S. degree in computer science from Peking University, China, and the Ph.D. degree in computer science from George Washington University. He is currently a Principal Inventive Scientist with AT&T Labs Research, Bedminster, NJ, USA. He has been working in the telecommunications industry for more than ten years and has 90 patents in the wireless communications. His research interests include graph theory and network optimization algorithms.

MIHAIL L. SICHITIU (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Notre Dame in 2001. He is teaching wireless networking and UAV courses. He is a Professor with the Department of Electrical and Computer Engineering, NCSU. His current research interests include wireless networks and communications for UAVs. In these systems, he is studying problems related to localization, time synchronization, emulation, routing, fairness, and modeling.

ISMAIL GÜVENÇ (Fellow, IEEE) is a Professor with the Department of Electrical and Computer Engineering, North Carolina State University. He has published more than 260 conference/journal papers, several standardization contributions, four books, and over 30 U.S. patents. His recent research interests include 5G/6G wireless networks, UAV communications, millimeter/terahertz communications, and heterogeneous networks. He is a recipient of several awards, including the 2021 University Faculty Scholar Award, the 2019 R. Ray Bennett Faculty Fellow Award, the 2016 FIU COE Faculty Research Award, the 2015 NSF CAREER Award, the 2014 Ralph E. Powe Junior Faculty Award, and the 2006 USF Outstanding Dissertation Award. He is a Senior Member of the National Academy of Inventors.



ARUPJYOTI (ARUP) BHUYAN (Senior Member, IEEE) received the Ph.D. degree in engineering and applied sciences from Yale University. He is a Wireless Researcher with Idaho National Laboratory (INL) and the Technical Director of the INL Wireless Security Institute. He has extensive industry experience in wireless communications from his work before he joined INL in October, 2015. The focus of his research is on secure implementation of future generations of wireless communications with scientific exploration and engineering innovations across the fields of wireless technology, cybersecurity, and computational science. Specific goals are to lead and focus wireless security research efforts for 5G and Beyond with national impact, to secure communications for a nationwide unmanned aerial system and for 5G spectrum sharing with distributed scheduling.