# Chimera: exploiting UAS flight path information to optimize heterogeneous data transmission

Russell Shirey, Sanjay Rao, Shreyas Sundaram
School of Electrical and Computer Engineering, Purdue University
{rshirey, sanjay, sundara2}@purdue.edu

***Abstract*—Unmanned Aerial Systems (UAS) collect and transmit data such as live video and radar images, which have different latency and reliability requirements, over wireless links that exhibit much performance variability. In this paper, we make three contributions. First, we show through a characterization of two real-world UAS flight datasets that there is significant opportunity to optimize data transmission in UAS settings by exploiting knowledge of UAS flight paths. Second, we developed Chimera, a system that taps into this opportunity while transmitting heterogeneous data streams over UAS networks. Chimera learns a model online that relates UAS network throughput to the flight path, and combines the model with a control framework that optimizes transmissions based on long-range throughput prediction. Third, with a combination of emulation and simulation experiments using real-world flight traces, we show Chimera's effectiveness. Specifically, Chimera reduces penalties related to dropped radar images by $72.4\% - 100\%$ compared to an algorithm agnostic to flight path information, and achieves an average bitrate of $90.5\%$ compared to an optimal scheme that knows the exact future throughput, with only a minimal increase in radar images dropped.**

***Index Terms*—Unmanned Aerial Systems (UAS), Drones, Networking, Throughput, Sensors.**

## I. Introduction

Unmanned Aerial Systems (UAS) are increasingly used to perform sensing and data-gathering in a variety of scenarios, due to their ability to go to areas where humans cannot, and gain vantage points with sensors that are not possible from the ground [1]–[6]. Sensor types and capabilities, and options for mounting on UAS, have increased to provide many options to gain insight from an aerial viewpoint [4], [7]–[9]. Furthermore, the data transmitted from these sensors has diverse network requirements; for instance, live video has stringent timeliness requirements but can tolerate some quality degradation, while radar images must be transmitted reliably and typically require transmission within several tens of seconds.

UAS networks can present **challenges** to transmit sensor data in real-time due to flight dynamics and bandwidth limitations [4], [5], [10]–[13]. However, they also offer **opportunities** since UAS network performance depends on the flight path. To motivate this, we present observations from two real-world UAS flight test datasets (§II). The datasets are unique in that they are from distances exceeding current Federal Aviation Administration (FAA) limits requiring Visual Line of Sight (VLOS) [14], motivated by the growing interest in extending the range of UAS networking [1], [15], [16].

978-1-6654-4131-5/21/$31.00 ©2021 IEEE

In this paper, we are primarily motivated by the question: is it feasible to exploit knowledge of UAS flight paths to more effectively transmit sensor data with demanding performance requirements? This problem presents multiple key design challenges in dealing with: (i) UAS flight throughput dynamics, (ii) variable sensor data rates and requirements, and (iii) the impact current sensor data transmission actions have on future transmissions. To address these challenges, we develop **Chimera**, a system for optimizing transmission of heterogeneous sensor data over variable UAS network environments. Chimera is based on an optimal control framework, performing online optimization continuously to yield a feedback control policy that makes transmission decisions for two different sensor data streams: (i) video, and (ii) Synthetic Aperture Radar (SAR) images. While we focus on these streams for concreteness, Chimera can be generalized to more diverse data streams.

A novel aspect of Chimera is its use of flight path information to predict network throughput, and using these predictions in its control algorithm. Towards this end, we develop and validate UAS network throughput prediction models, given flight path knowledge from real-world flights. A key consideration is developing a pragmatic model whose parameters can be learnt online using information from the initial stages of the flight. Our analysis indicates that simple regression models based on the distance and UAS orientation relative to the Ground Control Station (GCS) are effective in prediction, even into the future. We integrate our models into Chimera, which learns both the dependence of throughput on flight path, and also an error model pertaining to throughput prediction errors. Chimera's approach is viable since flight paths are typically determined in advance. Further, its optimal control framework uses a continual planning model, which allows it to adapt to flight path changes in addition to learning and improving throughput and error models over time (§III-B).

We implemented Chimera and integrated it with a video encoder to stream live video, and an application for generating and transmitting SAR data. We evaluated Chimera using an emulation test-bed, and also built a simulation environment to test a multitude of additional scenarios and evaluate Chimera's design-points. Through a combination of real-world UAS flight throughput traces collected in different locations with different UAS types, and synthetic traces generated using our models, we show that Chimera performs effectively when transmitting diverse sensor data. Chimera offers significant improvement over an approach that does not exploit flight path information.

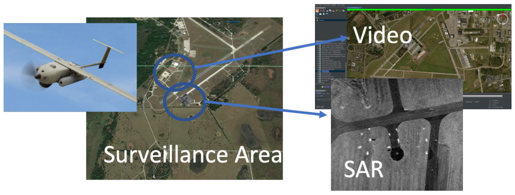

Figure 1: Example UAS Surveillance Scenario

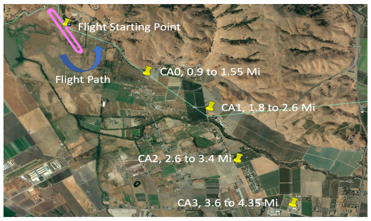

Figure 2: California flights

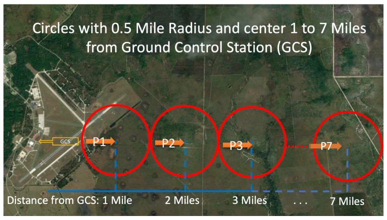
Circles with 0.5 Mile Radius and center 1 to 7 Miles from Ground Control Station (GCS)
GCS
P1
P2
P3
P7
Distance from GCS: 1 Mile
2 Miles
3 Miles
. . .
7 Miles

Figure 3: Florida flights

Chimera accomplishes this by reducing penalties related to dropped SAR image transmissions by $72.4\% - 100\%$ across all of the aforementioned real-world flight test traces, while achieving comparable video qualities on our emulation testbed. Further, Chimera achieves a bitrate of $90.5\%$ compared to an optimal scheme that knows exact future throughput information, with only a modest increase in SAR images dropped. Finally, we believe our models for generating synthetic traces can be useful to the community in their own right[1].

## II. Motivating Measurements and Challenges

In this section, we describe our problem setting, and present measurements that motivate our approach.

### A. UAS Sensor Data Transmission Problem

Many UAS settings, such as security surveillance [7], [17], search and rescue missions [18], [19] and environmental monitoring [20], [21], involve both video and other sensors such as radar imaging [1], [7], [19], [21], called SAR, as shown in Fig. 1. Video provides color imaging of small visible areas, while SAR imagery provides a wide-area all-weather capability that penetrates fog, smoke and atmospheric obstructions [21]–[24] (e.g., the ability to penetrate through smoke is critical in fire monitoring scenarios [8], [9]). Live video is often monitored by a person, while SAR images are typically used by algorithms. Video transmission is near real-time and can tolerate loss. In contrast, SAR images are more loss sensitive, but can tolerate more delays since they cover much wider regions [22], [24]. However, extensive delays (e.g., $> 1$ minute) can make the images stale and the area may need to be surveyed again. The resolution and priority of each sensor varies based on the mission.

A key challenge addressed in this paper is how to simultaneously transmit both video and SAR data in challenging UAS networking environments, while taking the requirements of each data stream into account. We focus on industrial, or *professional grade*, drones that often have the capacity to support such multi-sensor missions [1], [25]–[28]. Our approach can also be generalized to diverse sensor data types, with different reliability and timeliness requirements.

### B. Motivating flight test measurements

Most existing measurement studies of UAS networking were performed at limited flight range from the GCS (less than 0.25 miles) or using LTE, which limits scenarios because of required infrastructure [11], [29]–[34]. The only prior work we are aware of with longer range distances is [35], which is our own work. We also flew flight tests at long-range distances in California (CA) to collect additional flight test data beyond VLOS, an area seeing growing interest [1], [15], [16].

- *CA Dataset.* We flew and collected this dataset in California (CA) in partnership with a local flight business. Fig. 2 shows the flight setup. We had permission to fly in a range of roughly 0.8 miles (due to private land). To extend the collection range, we moved our GCS (co-located with our car), flying a full flight and collecting several loops of data (relevant for aerial surveillance) at each GCS location. The entire dataset was collected at four different GCS collection locations at distances ranging from 0.9 to 4.35 miles. The flight used a multirotor UAS [36] flying in an oval racetrack orbit at speeds of 15-25 knots, slower than the fixed wing alternative.
- *FL Dataset.* We collected this dataset [35] using a fixed wing UAS [36] at distances exceeding FAA UAS limits [14] with necessary approvals. The flight path (Fig. 3) spanned a distance of up to 7.5 miles between the UAS and GCS. The UAS flew circular patterns at speeds of 50-60 knots at each mile interval, with 0.5 mile radius each, and collected data using two different power levels (500mW and 2W), with performance at the 2W level generally being better at the same range. We refer to these datasets as **FL1** and **FL2**, respectively.

All data communication in the datasets used point-to-point tactical radios [37], [38]. The UAS had an omnidirectional antenna (due to the constant UAS movement), while the GCS had either directional or omnidirectional antenna configurations. Both are common in practice and each has trade-offs [5], [39], [40]. Omnidirectional antennas provide more flexibility and do not need a tracker, while directional antennas provide higher gain (more range and higher throughput) at the cost of a larger antenna and tracker to be pointed at the UAS. Both datasets contain second-by-second throughput (collected using iPerf [41]) and location data. The FL datasets contain TCP data, while the CA datasets are based on UDP measurements.

**Observations from flight test data:** For each dataset, we analyze the network performance as a function of the UAS distance. By *distance*, we refer to the distance from the GCS to UAS (including altitude). We notice the performance depends on whether the UAS flies *towards* or *away* from the GCS (for reasons discussed below). Hence, we present measurements collected when the UAS is in each orientation separately.

Fig. 4 (left) shows loss rates for the CA data grouped by

[1] We release our trace generation code and generated traces at: https://github.com/Purdue-ISL/ChimeraTraceGeneration

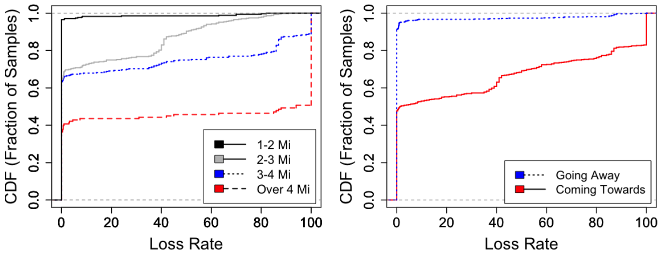

Figure 4: UDP loss with distance (left), orientation (right), **CA**

distance ranges with a UDP transmission rate of 5.0 Mbps. We see the loss rate increases with distance, as expected. More interestingly, the right figure shows the UAS orientation affects network performance, revealing higher UDP loss rates in the **coming towards** orientation. Upon further inspection, the UAS tilts slightly in the direction it is facing and thus has some obstruction while **coming towards**. We also analyzed the data in the FL dataset and found similar trends held, with throughput showing dependence on both distance and UAS orientation (key FL dataset orientation factors being the asymmetrical shape of the fixed wing UAS and the UAS antenna gain patterns being slightly different in each direction [35]). The FL throughput is higher than CA at similar distances because the FL dataset used a directional antenna on the ground with higher gain than the CA omnidirectional.

For the CA dataset, notice that loss rates in the range of over 4 miles are too high to support most usable sensor data applications, while performance at a range of under a mile was typically abundantly good, such that simple transmission algorithms in this range are sufficient for data transmission. Hence, we focus our analysis in the rest of the paper on data collected in **CA1-2** (see Fig. 2). This data includes UDP measurements conducted at multiple sending rates (several loops were flown in each orbit, and a different sending rate used in each loop), and we consider the effective throughput seen by the receiver when the sender transmitted at a rate that saturated the link. Since the FL dataset was collected in one flight from the same GCS, we stitch the circles together in the rest of the paper to form an extended flight path from 0.5 to 7.5 miles, used for our analysis. We do not stitch the CA dataset together since that was collected with various flights at different GCS locations. Table I shows our test datasets.

Table I: Flight Test Traces

| Name | UAS Type | Distance (Mi) | Power | Antenna |
|---|---|---|---|---|
| FL1 | Fixed Wing | 0.5-7.5 | 500mW | Dir |
| FL2 | Fixed Wing | 0.5-7.5 | 2W | Dir |
| CA1 | Multirotor | 1.8-2.6 | 2W | Omni |
| CA2 | Multirotor | 2.6-3.4 | 2W | Omni |

**Opportunities and challenges:** Overall, our datasets show the dependence of UAS network performance on both distance and UAS orientation, providing an opportunity to plan data transmission based on the flight path. The key challenges are building throughput prediction models that exploit knowledge of the flight path, and handling prediction errors. We will tackle these challenges in the rest of the paper.

## III. Chimera Design

Influenced by the challenges, insights, and opportunities discussed in the previous section, we next outline the problem we are seeking to solve, and details of our design of Chimera.

### A. Problem Formulation

We now formally state our UAS sensor data transmission problem. Consider a flight of duration $T$ seconds. The UAS transmits video and SAR data, with an associated reward for each. Our objective is to maximize the reward over the full flight. We focus on settings where SAR images are continuously generated, resulting in periodic traffic, typical of many real-world use-cases [17]. The SAR data stream involves a sequence of images, each $B_s$ bytes, generated every $G$ seconds (typically $< 10$ seconds) [17], [22], [24], and which must be transmitted within a time limit of $L$ seconds (on the order of tens of seconds to a few minutes based on the scenarios described in §II). Any image that is transmitted within the deadline $L$ is valuable, and an image that misses the deadline is stale and loses its value. The video transmission objective is to achieve the highest bitrate possible (up to a maximum bitrate needed by that video). We design Chimera to capture the relative importance of video and SAR transmission in a reward function, which we describe in this section. This function rewards higher video bitrates and penalizes dropped SAR images. We often do not have sufficient throughput to transmit the entirety of our data throughout the flight. As a result, Chimera must make decisions on what data to transmit, or delay, at any given time. If we assign too much throughput to video, then SAR cannot keep up with timeliness requirements and becomes stale, resulting in dropped image penalties. If we assign too little throughput to video, then we are potentially wasting opportunity where we could have transmitted higher quality video and still met our SAR requirements.

### B. Chimera overview

We briefly describe Chimera's four major components and then dive into further details in the subsequent sections.

**Optimal control framework:** Chimera produces a feedback control policy that makes transmission decisions at each time-step as a function of the current state and predictions of future throughput. In particular, Chimera optimizes expected rewards over the duration of the flight while accounting for prediction errors, which we characterize and model online.

**Prediction model:** Chimera's prediction model is built using real-world flight data and utilizes the known UAS flight path to predict future network performance. It includes parameters unique to UAS flight, such as distance and orientation, building a robust model for future throughput prediction. Chimera can work with any generic transport protocol, using network bandwidth estimates provided by the transport layer.

**Adding robustness to prediction errors:** Chimera adjusts to network prediction errors by carefully building a weighted

Table II: Problem notation table

| Term | Meaning |
|---|---|
| $T, \tau, N$ | flight, epoch duration (secs), flight duration (epochs) |
| $R$, $R_d$, $R_v$ | total reward, dropped SAR penalty, and video reward functions |
| $G, L, B_s$ | SAR gen time (secs), transmission deadline (secs), size (bytes) |
| $V_t, V_n$ | live video average bitrate (Mbps) during time $t$, during epoch $n$ |
| $C_t, C_n$ | average available throughput (Mbps) during time $t$, during epoch $n$ |
| $B_{max}$, $b_n$ | max images in buffer, buffer images (fraction) at start of epoch $n$ |
| $G_n, D_n$ | SAR images generated during epoch $n$, dropped at the end of $n$ |
| $P_n$ | SAR images (fractional) transmitted during epoch $n$ |
| $e(n)$ | subset of all timesteps $t$ representing timesteps $t$ in the $n$th epoch |
| $\bar{\sigma}_n, \bar{\omega}_n, \bar{\epsilon}_n$ | distance, orientation, and throughput prediction error at epoch n |

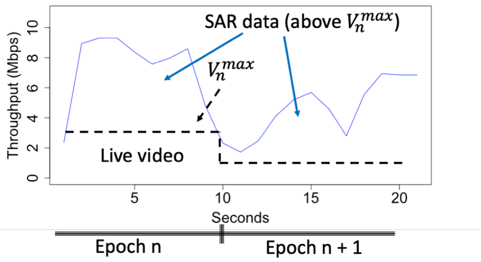

Figure 5: Chimera transmission illustration

probabilistic error model. This error model is integrated into the planning and decision-making process in order to provide robustness to errors and improve performance.

**Online learning:** It is difficult to gather UAS wireless performance data [5], [11], and flight environments can vary [6]. Because of this, Chimera incorporates an online learning process to train network and error models in flight. This online learning process can be used on its own, or be combined with models learnt from previous flights to increase accuracy.

**Adjustment to flight path changes:** Chimera's approach is viable given that flight paths are typically known *a priori* as part of the mission planning process, and to facilitate flight approval and coordination with proper authorities. We show in §V that Chimera need not know the entire flight path in order to optimize data transmission. Further, if flight paths change (e.g., due to an emergency or abrupt mission change), Chimera's feedback control policy enables it to quickly adapt. Specifically, Chimera's online optimization would be based on the current system state and models, with the new flight path, and provide a new set of optimal transmission decisions.

**Roadmap:** In the following sections, (i) we detail Chimera's optimal control model with knowledge of future throughput (§III-C), (ii) we develop models for predicting throughput using real-world datasets (§III-D), and (iii) we discuss how Chimera builds its throughput and error models online, and Chimera's algorithmic approach that utilizes them (§III-E).

### *C. Optimal control framework*

Let there be $T$ second time-steps, indexed by $t \in \{0, 1, 2, \ldots, T-1\}$. The throughput at each time-step is $C_t$. Our goal is to allocate the throughput between video and SAR traffic at each time-step so as to optimize the overall reward.

A unique opportunity in UAS settings is that the entire flight path is typically planned in advance, providing the potential to use a long term look-ahead window for planning. For computational efficiency reasons (as we will discuss further in §V), it may sometimes be desirable to conduct Chimera's planning at a coarser granularity than at each time-step. Towards this end, we partition the duration of the flight into $N$ epochs, each of a fixed length of time $\tau$. We index the epochs by the variable $n \in \{0, 1, 2, \ldots, N-1\}$. We next present Chimera's control model (Table II summarizes notation).

**Video:** Let $V_t$ be the achieved video bitrate at a given time-step $t$ (such that $V_t <= C_t$). Consider a sequence of achieved video bitrates at each time-step: $V_0, V_1, \ldots, V_{T-1}$. The achieved bitrate over the entire flight is $\sum_{t=0}^{T-1} V_t$. To enable epoch level decision-making, we define the quantity $e(n)$, where for each $n \in \{0, 1, \ldots, N-1\}$, $e(n)$ is the set of all time-steps $t$ in the $n$th epoch. Thus, our average video bitrate for epoch $n$, denoted $\bar{V}_n$ is:

$$\bar{V}_n = \frac{\sum_{t \in e(n)} V_t}{\tau}. \tag{1}$$

We select video values in order to maximize the reward (provided later in Eq. (7)) by influencing the balance of dedicated throughput to video and SAR. To do this, we specify the maximum video bitrate within an epoch, denoted $\bar{V}_n^{max}$, selected from a finite set of possible bitrates $\mathcal{V}$. The encoder targets (and never exceeds) a bitrate of $\bar{V}_n^{max}$ in each epoch, but adapts to dips in network throughput by encoding at lower bitrates as needed. For each $t \in e(n)$, we let $V_t \in [0, \bar{V}_n^{max}]$ denote the realized throughput of video during time-step $t$:

$$V_t = \min(C_t, \bar{V}_n^{max}). \tag{2}$$

Note that $V_t$ will be a random variable, determined by actual throughput $C_t$ during that time-step. The remaining throughput $C_t - V_t$ in any given time-step is dedicated to SAR transmission, through a scheduler. Fig. 5 shows an example of Chimera over a period of 20 seconds, split into 10 second epochs. During each epoch $n$, $\bar{V}_n^{max}$ is a dotted line and excess throughput during that epoch is devoted to SAR transmission. For each epoch, we seek a $\bar{V}_n^{max}$ that is achievable but also leaves sufficient throughput for SAR transmission. Without this limitation, SAR and video flows would compete with each other, and potentially starve the system. By predicting future throughput based on flight path, and varying data prioritization at different points during flight, we can improve our reward.

**SAR image transmission:** After allocation of throughput to video, the remaining throughput for each epoch is dedicated to SAR images, given by $\sum_{t \in e(n)} (C_t - V_t)$ (note that remaining throughput for SAR can be zero if $\bar{V}_n^{max}$ is equal to or exceeds available throughput at each time-step during epoch $n$). This strategy allows for long-term planning to maximize reward, while also adapting live video to short term fluctuations. The use of multiple levels of planning and adaptation takes advantage of long-term horizon throughput prediction based on flight path and also allows faster computations compared to second-by-second planning and processing.

**SAR buffer:** During each epoch $n$, $\bar{G}_n$ new full (integer) SAR images are generated and stored in the buffer for transmission. Let $\bar{b}_n$ denote the number of SAR images (fractional) stored in the buffer at the start of epoch $n$, and let $\bar{P}_n$ denote the number of SAR images (fractional as partial images can be in transmit) transmitted to the GCS during that epoch:

$$\bar{P}_n = \min\left\{ \frac{\sum_{t \in e(n)} (C_t - V_t)}{B_s}, \bar{b}_n \right\}. \tag{3}$$

Let $B_{max}$ denote the maximum number of full images that can be stored in the buffer (calculated based on the deadline, $L$, to transmit SAR images before they become stale). $L$ is a multiple of the generation time, $G$, such that:

$$B_{max} = L/G. \tag{4}$$

Let $\bar{D}_n$ denote the number of SAR images dropped from the buffer during epoch $n$ (penalized as an integer, even if part of the image was transmitted because only full SAR images transmitted are useful to us). The dynamics of the buffer are:

$$\begin{aligned} &\bar{b}_0 = 0, \\ &\bar{b}_{n+1} = \min\{\bar{b}_n - \bar{P}_n + \bar{G}_n, B_{max}\}, n \in \{0, 1, \ldots, N-1\}. \end{aligned} \tag{5}$$

Furthermore, the number of SAR images dropped from the buffer at the end of epoch $n$ is given by:

$$\bar{D}_n = \max\{0, \lceil \bar{b}_n - \bar{P}_n + \bar{G}_n - B_{max} \rceil\}. \tag{6}$$

**Decisions and rewards:** Consider a sequence of specified maximum video bitrates

$$\bar{V}^{max}_{0:N-1} \triangleq \{\bar{V}^{max}_0, \bar{V}^{max}_1, \ldots, \bar{V}^{max}_{N-1}\} \in \mathcal{V}^N,$$

where each $\bar{V}^{max}_n$ denotes the maximum video bitrate during epoch $n$. Each such sequence induces a sequence of video bitrates and dropped SAR images. Note that each of these quantities is a random variable, dependent on the realized throughput during each epoch. During flight, the system operator obtains a video transmission reward, $R_v(V_t)$, every second. The system operator also incurs a penalty, $R_d$, for the total SAR images dropped during flight, which can be represented as the summation of images dropped at each time-step, or equivalently the summation of images dropped across epochs.

**Flight reward with epochs:** The *expected reward* earned over the duration of the flight for a given sequence $\bar{V}^{max}_{0:N-1}$ is:

$$R(\bar{V}^{max}_{0:N-1}) = \mathbb{E}\left[ \sum_{t=0}^{T-1} R_v(V_t) - R_d\Big(\sum_{n=0}^{N-1} \bar{D}_n\Big) \right]. \tag{7}$$

**Optimization problem with epochs:** Our optimization problem based on epochs is:

$$\begin{aligned} &\max_{\bar{V}^{max}_{0:N-1} \in \bar{\mathcal{V}}^N} R(\bar{V}^{max}_{0:N-1}) \\ &\text{subject to (2), (3), (4), (5), and (6).} \end{aligned} \tag{8}$$

**Chimera extensions:** Chimera is extendable to different reward functions using the above framework. For example, it is easy to add a reward for successful full SAR image transmissions, instead of, or in addition to penalties for dropped SAR images. It is also possible to extend the reward function to minimize fluctuations in video bitrate by adding a smoothness function, $R_\Delta$, that penalizes changing video bitrates. To do this, we would calculate the difference in bitrates at each time-step: $\Delta_t = V_{t+1} - V_t$. Each change in bitrate would result in a corresponding penalty based on the smoothness function, $R_\Delta(\Delta_t)$, and a summation of these penalties may be taken.

### D. Throughput prediction model with Chimera

Our discussions in §III-C assume we know the throughput for each time-step, $C_t$, in advance. In this section, we develop and validate models to capture how network throughput depends on UAS flight path, leveraging the datasets in §II. Our models consider three key factors: (i) UAS distance to the GCS, (ii) UAS orientation, and (iii) recent throughput samples. We first look at prediction for the immediate next time-step, and then consider longer look-aheads.

*1) Prediction over the next time-step:* Our data shows that UAS distance to the GCS affects throughput. We built a regression model to calculate predicted throughput at time $t$, $\hat{C}_t$, based on distance, $\sigma_t$, with an estimated error term, $\hat{\epsilon}_t$:

$$\hat{C}_t = \alpha\sigma_t + \gamma + \hat{\epsilon}_t. \tag{9}$$

In Eq. (9), coefficients $\alpha \; and \; \gamma$ are calculated based on gathered data. We considered a distance to throughput relationship of linear, log, and a combination of both, and found linear to be slightly better than log, and comparable to combination. Further investigation showed that SNR had a logarithmic relationship to distance, as expected [42]–[44], but throughput is also affected by factors other than SNR.

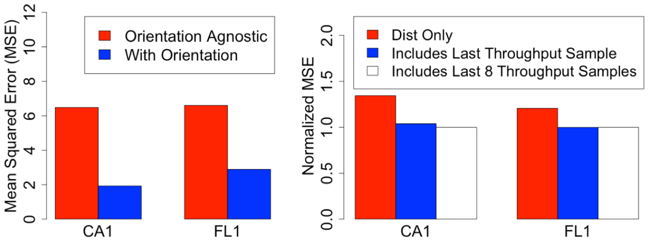

Figure 6: Throughput model differences with orientation (left) and inclusion of previous throughput samples (right).

**Considering orientation:** We built two regression models: one using the entire dataset (*orientation agnostic*) and the other with separate models for each orientation. Fig. 6 (left) shows prediction errors are reduced by building seperate models for **CA1** and **FL1**. The other datasets were qualitatively similar.

**Previous throughput samples:** We explore if we can improve our model by including additional previous throughput samples. We modify Eq. (9) to include previous throughput samples and calculate regression models using 0, 1, and 8 previous throughput samples and compare predicted throughput to actual over the same time period. Fig. 6 (right) shows incorporating previous throughput improves prediction accuracy.

However, the improvement gained by including throughput samples beyond the previous time-step is small.

**Network model for Chimera:** Based on our results, Chimera uses a model for throughput prediction at time $t$, $\hat{C}_t$, based on $C_{t-1}$ (the actual throughput at time $t-1$), an estimated error term $\hat{\epsilon}_t$, and $\omega_t$ and $\sigma_t$, which respectively denote the orientation and distance at time $t$:

$$\hat{C}_t = \begin{cases} \delta C_{t-1} + \alpha\sigma_t + \gamma + \hat{\epsilon}_t & \text{if } \omega_t = 0 \\ \delta' C_{t-1} + \alpha'\sigma_t + \gamma' + \hat{\epsilon}_t & \text{if } \omega_t = 1. \end{cases} \tag{10}$$

In Eq. (10), coefficients $\delta$, $\alpha$, $\gamma$, $\delta'$, $\alpha'$, *and* $\gamma'$ are calculated based on gathered data, while $\omega_t$ indicates orientation (i.e., $\omega_t = 0$ for **going away** at timestep $t$ and $\omega_t = 1$ otherwise).

*2) Prediction over longer look-ahead:* Chimera must predict throughput over the entire flight path. Consider that throughput samples up to time $t-1$ are available. We seek to predict throughput for time-steps $t, t+1, \ldots, t+k$. Extending our analysis, we consider a model where throughput at time-step $t+k$, $C_{t+k}$, depends on $C_{t-1}$, $\omega_{t+k}$ and $\sigma_{t+k}$ (orientation and distance at time $t+k$), and $\hat{\epsilon}_{t+k}$ (estimated error):

$$\hat{C}_{t+k} = \begin{cases} \delta_k C_{t-1} + \alpha_k\sigma_{t+k} + \gamma_k + \hat{\epsilon}_{t+k} & \text{if } \omega_t = 0 \\ \delta'_k C_{t-1} + \alpha'_k\sigma_{t+k} + \gamma'_k + \hat{\epsilon}_{t+k} & \text{if } \omega_t = 1. \end{cases} \tag{11}$$

Fig. 7 compares throughput prediction errors for **FL1**, using Eq. (11), with different coefficients turned off (i.e., considering distance + past throughput, and each on its own). We repeat this regression for each lag $k$, iterating through all $t$ values for each $k$ value in our flight, and show the MSE across lags. The left side considers the **going away** orientation and the right considers **coming towards**. The results show that (i) distance is important, and (ii) only considering previous throughput is insufficient. Further, previous throughput helps for low lag, but not larger lag. We also see a larger error in **coming towards** compared to **going away**, owing to higher variability, as discussed in §II. We see a slight oscillation in MSE when considering past throughput only, but it is reduced with distance considered, since the distance coefficient magnitude is more significant. Based on these results, we remove previous throughput samples from our long-term prediction model.

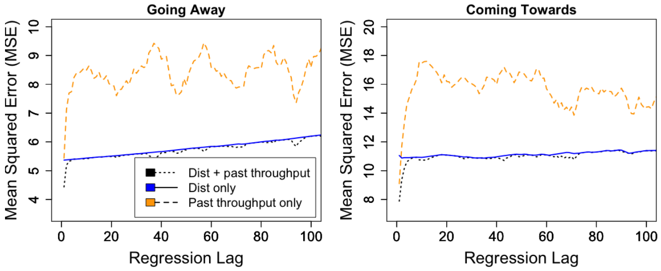

Figure 7: MSE across different time lags when considering distance + past throughput, or each individually, **FL1**.

*3) Prediction over epochs:* Chimera may make decisions at a coarser granularity of epochs, as discussed in §III-C. This is done by using epochs instead of time-steps in Eq. (10), where the epoch distance and throughput ($\bar{\sigma}_n$ and $\bar{C}_n$) are the average distance and throughput over all time steps within that epoch, and we remove dependence on previous throughput, as previously discussed. Throughput dependence on both UAS distance and orientation continued to hold at the epoch scale. Further, average throughput prediction error within an epoch decreased as epoch size increased, with median prediction errors of 15.3%, 11.8%, 9.0%, and 5.2% for epoch sizes of 1, 5, 10, and 30, respectively. However, there may still be significant throughput variability within an epoch, and larger epochs may reduce Chimera's responsiveness (§V-C).

### *E. Incorporating predictions into Chimera*

We next discuss how we integrate the network models in §III-D into Chimera's planning algorithm, run each epoch.

**Online optimization:** We implement Chimera's online optimization as a Dynamic Program (DP) that incorporates the UAS flight network characteristics to maximize reward (evaluation results are shown in §V). We first discuss the basic DP, and then detail our error model integration. The DP state is the number of untransmitted SAR images (SAR buffer). For each epoch and state, the DP computes the optimal sequence of $\bar{V}_n^{max}$, starting at that epoch until the end of the look-ahead window (by default, the trace end), maximizing the reward.

Let the throughput prediction for the next epoch be $\hat{\bar{C}}_n$. For each value of $\bar{V}_n^{max}$, the average video bitrate transmitted during that epoch, $\bar{V}_n$, is limited to $\min\{\hat{\bar{C}}_n, \bar{V}_n^{max}\}$. We modify Eq. (3) to determine the (possibly fractional) SAR images transmitted during an epoch, $\bar{P}_n$, as follows:

$$\bar{P}_n = \min\{\tau\frac{\hat{\bar{C}}_n - \min\{\hat{\bar{C}}_n, \bar{V}_n^{max}\}}{B_s}, \bar{b}_n\}. \tag{12}$$

Note that if $\hat{\bar{C}}_n \leq \bar{V}_n^{max}$ then $\bar{P}_n = 0$. In this case, our throughput estimate is such that the target video bitrate cannot be, or exactly is, achieved. Available throughput will be devoted to video, estimated as $\hat{\bar{C}}_n$, and no SAR data will be transmitted. The rest of the computations follow §III-C. Our test results showing Chimera's benefits are presented in §V.

**Incorporating prediction error:** To be robust to prediction inaccuracies, we build an error model online. Chimera maintains a distribution of errors in predictions made earlier in the flight. Next, Chimera selects discrete points in the distribution (in our implementation, we select quartiles). Given a predicted throughput, $\hat{\bar{C}}_n$, Chimera considers each quartile error, $\epsilon_q$, adjusting throughput prediction for each error as follows: $\hat{\bar{C}}_q = \hat{\bar{C}}_n + \hat{\bar{C}}_n \times \epsilon_q$. Our DP evaluates the choice of $\bar{V}_n^{max}$ with each error-adjusted throughput prediction, weighing the resulting reward according to error probability. Chimera considers the summation of these weighted rewards to select the optimal choice of $\bar{V}_n^{max}$ at each epoch. Our results show incorporating an error model improves Chimera's performance (§V-C).

**Online learning:** Chimera uses online learning for throughput and error models. At the start of each epoch, Chimera calculates the previous throughput prediction error and reruns the throughput regression model in §III-D with all observed flight

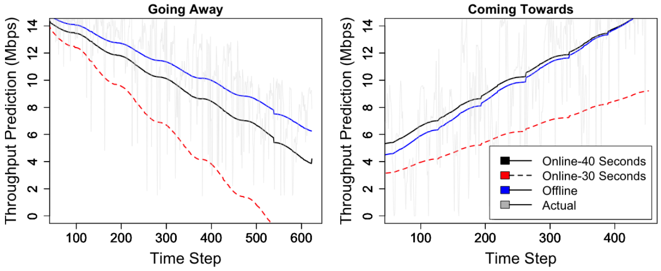

Figure 8: Comparison of online learning throughput prediction, after 30 and 40 seconds, compared to offline learning.

throughput data, allowing Chimera to adjust to changes and update predictions during flight. Figure 8 illustrates Chimera's online learning with the **FL1** trace, and shows the throughput predicted by the regression models learnt online after 30 and 40 seconds, compared to the prediction from an offline regression model that has all of the data ahead of time. The figure shows that Chimera can converge to the offline model within tens of seconds, a trend we also observed in other traces. This furthers our confidence in Chimera being able to quickly adapt to different environments.

An important consideration is how much previous data should be included in online learning. We experimented with several strategies for learning throughput and error models. For throughput, we considered (i) building a single regression model for the entire flight using all of the available data up to that point in the flight and (ii) building separate regression models based on orientation. For error models, we considered both strategies of using all flight data verses separate models based on orientation. Further, we also considered building separate error models based on distance ranges (e.g., distance bins covering 2 miles each) and orientation, motivated by the throughput prediction error distribution in §IV. We show in §V that building orientation-specific models typically improves performance for both error and throughput models, with the most benefits in traces that see each orientation more than once. Further, we found the addition of error models learnt online based on distance bins was similar to and did not improve Chimera's performance beyond using orientation-specific models-potentially due to limiting the data available for learning. Finally, while we focus on online learning for the most general settings, we note that UAS settings may have error and/or throughput model data from prior flights that Chimera can also leverage (shown in §V-C).

## IV. Evaluation Methodology

We evaluate Chimera with an actual implementation on an emulation test-bed and simulations, as we discuss below.

**Implementation and test-bed Setup:** We implemented Chimera and integrated it with a VP8 video encoder to stream live video, and an application for generating and transmitting SAR data. We leveraged the Salsify [45] codebase, making modifications to inform the encoder of a target bitrate by Chimera, described in §III. We used Mahimahi [46] to emulate flight network throughput, replaying the traces from the flight test datasets (§II). Our control setup integrated UAS distance and orientation, allowing Chimera to continuously run the online optimization, and make decisions based on the flight path and current state, while also providing an opportunity to adapt to changes in the progressive throughput and error models. We generate representative SAR image files and transmit these to the client GCS at a set rate that is updated every second based on the estimated remaining available throughput, per our protocol design. Our tests are run on a 64-bit Ubuntu 2-core machine, with 8 GB RAM - representative of a typical UAS system. We also tested at a large scale with a simulated setup that integrated the distance and orientation of the UAS into the logic, in order to provide more extensive sensitivity studies.

**Real-world traces:** We test using the throughput traces from multiple flights in FL and CA (§II, Table I). We also perform tests with dataset variants, which we discuss in §V. UAS flights typically involve several to tens of minutes of flight time. Since our CA flight traces are shorter duration, we extended them by an additional 3 loops, by synthetically generating throughput loops (as described below) and appending it to our dataset (this is reasonable because survey missions often complete multiple loops of an area).

**Synthetic traces:** Since real-world flight traces are challenging to collect, we generated synthetic traces for additional testing using the network model from §III-D (Eq. (10)), and coefficients gleaned from real-world flight data. We model the error term $\epsilon_t$ separately for different distance range and orientation bins, using a procedure described below.

Fig. 9 shows a histogram of prediction errors, and best fit for both a Normal and Cauchy distribution, for the FL1 trace and an example bin. Visually, the Normal distribution is not a good fit because it does not encompass the peak or tails, while the Cauchy distribution is a better fit. We used the Kolmogorov-Smirnov (KS) test [47] to evaluate the null hypothesis that the prediction errors are drawn from a specific distribution. We considered Normal, Cauchy, Lognormal, Gamma, and Weibull distributions (we considered shifts of errors to make errors non-negative for the latter distributions). For all bins of the FL and CA datasets, we were unable to reject the null hypothesis that the prediction errors fit a Cauchy distribution at a significance level of 0.05. In contrast, we rejected the null hypothesis that the errors fit a Lognormal and Gamma distribution in all cases, and in the vast majority of cases for Normal and Weibull. We considered both percentage and absolute errors, with consistent results. These results motivated us to model errors based on the Cauchy distribution, confirming what we visually saw in Fig. 9: that the error distribution has a higher peak and longer tails than can be captured with a Normal distribution. To ensure meaningful results (e.g., avoid negative throughput), the tail was truncated on both sides.

**Schemes:** We compared Chimera with many schemes:

- *Flight Agnostic*: Our baseline scheme uses a model-predictive controller with a look-ahead window of 5 epochs, and throughput prediction based on the average throughput in the past 5 epochs. In each look-ahead window, the same DP as

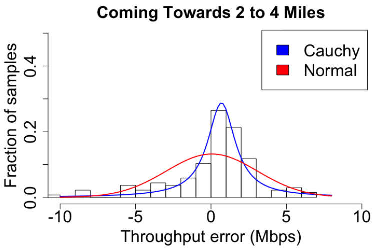

Figure 9: Modeling prediction errors

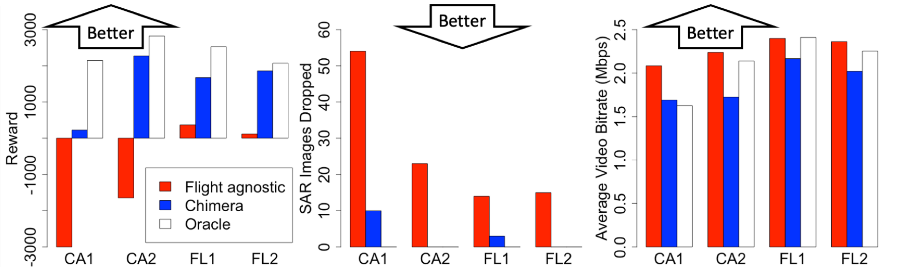

Figure 10: Chimera performance with emulation testing, showing comparisons of reward (left), SAR images dropped (middle), and the average video bitrate (right).

Chimera is run to best allocate bandwidth between video and SAR data streams. This scheme does not exploit knowledge of the UAS flight path, and is inspired by an algorithm widely used in the context of Internet video streaming [48].

- *Oracle*: This scheme assumes perfect knowledge of throughput for the duration of the trace at the start of the flight, and executes a DP based on the model in §III-C.
- *Chimera and variants*: By default, we evaluate *Chimera* with all its features, including online learning and a probabilistic error model. We also explore several variants to test key decisions of Chimera, which we detail later.

**Evaluation settings:** We consider a reward function (§III-C) based on live video and SAR image transmission. We set video reward at each time-step, $R_v(V_t)$, to equal the video bitrate (Mbps) received at that time-step. Let the maximum possible reward for video transmissions be $M$ (achieved when video is always transmitted at the highest rate), $I$ be the total number of SAR images generated, and $D$ be the total number of dropped images during flight. Then, we set the penalty for dropped SAR images to be $W \times \frac{M}{I} \times D$. Here, $W$ is a parameter capturing that if all SAR images were dropped, the penalty would be $W$ times the reward obtained if video were always transmitted at the highest rate. We use a default $W$ of 8 since a dropped SAR image implies the data is completely lost, while video could still be transmitted at degraded bitrates. We also evaluate Chimera with different $W$ values. We set possible values of $\bar{V}_n^{max}$ to be $\{1, 2.5, 5\}$, corresponding to bitrates typical for standard and high definition video [49]. Our SAR sensor generates a full image every 5 seconds, based on a real-world system [17]. We use SAR images of 16, 32, and 40 Mb for **CA1-2**, **FL1**, and **FL2**, respectively, modeling higher resolution images in datasets with better throughput. We set the SAR transmission deadline to be $L = 60$ *seconds*.

## V. Results

Our test results show that (i) utilizing UAS flight path improves performance (beneficial to use both distance and orientation), (ii) performance is further increased by using a probabilistic error model, and (iii) online learning performs well, even with no prior knowledge or model insights.

### A. *Effectiveness of Chimera: Emulation results*

Fig. 10 shows a comparison of Flight Agnostic, Chimera, and Oracle with our traces on the emulation testbed. The left-most figure corresponds to total reward, while the other figures show performance of each sensor stream: SAR images dropped (middle), and the average video bitrate measured at the receiver (right). Chimera performs much better than the Flight Agnostic scheme because it significantly reduces SAR drops, while maintaining a comparable video bitrate (the negative Y-Axis is truncated at -3000, and Flight Agnostic achieves a reward of -8049.12 in **CA1**). Flight Agnostic performs worse because it is more aggressive with video bitrate and does not properly prepare for periods of poor throughput. In contrast, Chimera and the Oracle account for future periods of lower throughput by throttling back the maximum permitted video bitrate. Consequently, the SAR buffer (number of untransmitted images) fills up faster for Flight Agnostic with 54 images dropped in (**CA1**), while only 10 images are dropped with Chimera.

While Chimera performs comparably to the Oracle in most traces, there is a noticeable gap to the Oracle for the **CA1** trace. Upon further inspection, this is because **CA1** involved a sharp and prolonged drop in throughput when transitioning to the orientation with poor performance for the first time, leading to dropped images. Chimera starts with no prior knowledge, but its performance improves over time as it quickly learns better throughput and error models. We note that UAS flights are typically longer, allowing online learning approaches to work even better. Finally, Chimera can improve performance using models learnt from previous flights (§V-C), which may be available in many scenarios.

### B. *Sensitivity to traces*

Since we are not aware of any other real-world UAS flight datasets at long ranges, we test Chimera with several additional traces, described below. Our results are based on simulations. All schemes perform slightly better using the simulator (since it does not account for factors such as processing and encoding delays), but we verified the relative performance is similar to the emulator. Across the traces used for validation, Chimera achieved 63.1% of the reward of the Oracle in emulation, compared to 63.6% of the Oracle reward in simulation.

**Trace variants:** Fig. 11 shows the results of trace variants with simulated testing. We consider each trace in reverse (noted Rev) and also explore starting at different points in the flight (i.e., starting at the 1Q, 2Q, and 3Q point in the

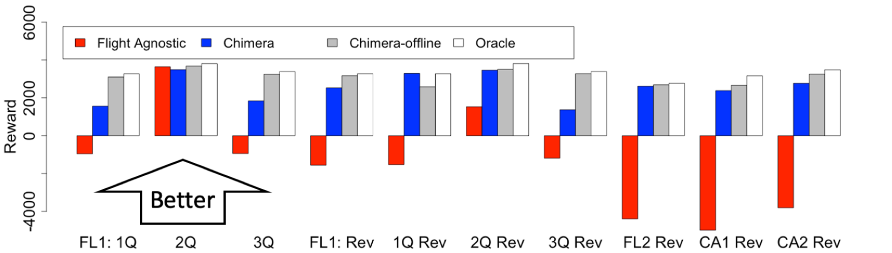

Figure 11: Chimera performance for variants, showing simulated testing of starting in different positions with **FL1**, and also the reverse direction for all traces.

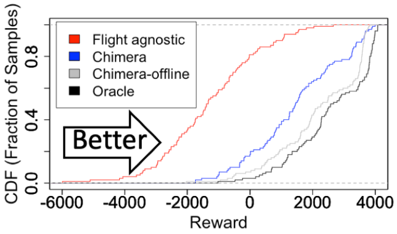

Figure 12: Chimera reward improvement with synthetic traces.

flight) by adjusting our starting position and throughput to the corresponding time-step in the trace and then completing a full loop. This works out to the same number of time-steps as the original trace, with location variance relative to the beginning and end of the flight (e.g., starting and ending closer or further from the GCS, or in a different orientation). Chimera performs well in all cases, with relative scheme performance following the same trends as in our initial testing. For the 2Q **FL1** variant, Flight Agnostic performs slightly better than Chimera. Here, the flight begins in an area of poor throughput which gets better, leading Chimera to be conservative in bitrate allocated to video. Flight Agnostic is also slightly more conservative, but its model ignores the initial data after 5 epochs, whereas Chimera continues to utilize this data in its throughput model.

**Synthetic traces:** We generated 100 synthetic traces using the methodology in §IV, based on a flight path spanning 0.5 to 7.5 miles (like the FL datasets). Fig. 12 shows a CDF of the reward across traces. Chimera out-performs Flight Agnostic in all cases, and performs much closer to the Oracle scheme.

### *C. Evaluating design variants*

We next evaluate variants of Chimera with simulated tests to both explore the importance of some of its decisions and also understand the trade-offs involved with some alternatives.

**Importance of considering prediction error:** Chimera trains and uses a probabilistic error model to account for prediction errors (III-E). Fig. 13 (left) compares Chimera and a variant, *Chimera-Non robust*, which does not account for prediction errors. The performance is improved when using an error model, with the benefits being particularly significant in the **FL1** and **CA2** traces. For **FL2** alone, adding the error model results in a slight reduction in performance since the algorithm is a bit more conservative in terms of the maximum video bitrates permitted. For this trace, throughput is both consistent and also plentiful enough that the prediction errors can be recovered from and are not enough to cause SAR drops.

**Online learning variants:** We compare Chimera to an orientation agnostic variant of online learning that does not consider UAS orientation in the network and error models. Fig. 13 (right) shows a comparison of the rewards. We see the FL dataset tests are comparable. CA performance is much higher when considering orientation because (i) the throughput in these traces is more sensitive to orientation and (ii) the traces have more loops, allowing the network models to fully train with each orientation and then utilize the models in the subsequent loops. We also considered variants where the switch to a new orientation model was delayed several epochs. This allowed time for the new orientation model to ingest data prior to use, but we found the tests were inconclusive in providing a clear direction for higher performance.

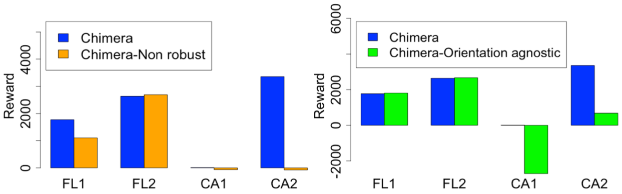

Figure 13: Showing the benefits of modeling the prediction errors with Chimera (left), and orientation sensitivity (right)

**Facilitating learning with prior data:** We explore the feasibility of training a model in one flight and using this model for a flight in a similar environment. To test, we trained a model learned from the actual trace, and used this model to test with multiple synthetic traces. We saw benefits to using a prior model, improving the average reward from $1388.44$ to $2377.76$, an increase of $71.25\%$, with **FL1**. We also tested how online learning improves over multiple loops in the flight path, common for many scenarios. To test, we appended a synthetic trace loop to our original trace and compared the performance over two loops with the single loop (with the reward multiplied by 2 for relevant comparison). We repeated this test with 5 different appended traces and saw a $9.3\%$ average reward increase compared to the reward for the regular trace, showing Chimera can improve performance by utilizing models trained from different flights in similar environments.

**Look-ahead sensitivity:** We explored Chimera's reward with different look-ahead windows. Our results concluded that increasing the look-ahead provides benefits up to 25 epochs (250 seconds). Benefits diminish beyond this point, indicating that while a longer look-ahead window is useful, opportunities exist to reduce Chimera's computational requirements by reducing the look-ahead window, while also enabling Chimera to better adapt to future flight path changes (e.g., emergency situations). We also tested Flight Agnostic and found even with an unlimited look-ahead, it performed worse than Chimera: for

**FL1**, the reward was lower by 761.46 and 2537.45, relative to Chimera, for the forward and reverse trace, respectively.

**Impact of epoch size:** The choice of epoch size impacts Chimera's performance. Since the maximum permitted video bitrate is constant during an epoch, a longer epoch size reduces Chimera's ability to adapt. Conversely, with a longer epoch size the DP is run less often and takes less time to run (due to less iterations). We tested epoch sizes of 5, 10, and 30 seconds, and found 5 and 10 to perform best with Chimera, with rewards of 1993.71 and 1772.85, respectively, for the **FL1** trace. Performance degraded at 30 seconds, although Chimera still out-performed the Flight Agnostic scheme significantly. We also measured DP computation times on a MacBook Pro. Average run time was 6.484, 1.357, 0.700, and 0.210 seconds for epoch sizes of 1, 5, 10, and 30, respectively, indicating an epoch size of 5 or 10 strikes a good balance between responsiveness and computation needs. Computational requirements can also be further reduced by limiting the look-ahead.

**Varying reward function:** We explore Chimera's ability to work with different reward functions by changing the parameter $W$, which captures the importance of SAR relative to video. We have so far used $W = 8$, which indicates that the penalty of dropping all SAR images is 8 × higher than the reward of seeing video at the highest bitrate throughout the flight. We experimented with weights of $W = 2$ and $W = 12$, which decrease and increase the importance of SAR relative to video, respectively. Chimera significantly out-performed Flight Agnostic, with an average reward increase of 2980.65 compared to Flight Agnostic for **FL1**, showing robustness even with smaller and more severe SAR drop penalties.

## VI. Related Work

**UAS data transmission:** Recent papers [11], [29], [30], [50] explore UAS video streaming for video on demand settings using Adaptive BitRate (ABR) algorithms. These works use past network performance (e.g., throughput of the last few video chunks) for future throughput prediction (similar to Flight Agnostic). In contrast, we focus on *live video*, and joint transmission with SAR data. Further, we develop models that predict future throughput based on flight path, complemented with an error model. Uses of wireless sensor networks for UAS surveillance with path planning is studied in [3], [51]–[53]. These studies focus on discrete connectivity and sensor data sizes rather than the dynamic throughput of real-world UAS networks and variable sensor data. In contrast, we focus on how to predict network throughput, and effectively transmit heterogeneous sensor data, given a long-range UAS flight path.

**UAS communication and networking:** Recent work [34] analyzes hobby UAS flight data and how to generate traffic unique to these settings. However, this work encompasses limited distances based on WiFi, and only works for a few specific types of UAS. The paper [54] augments low-power IoT devices with nearby UAS to provide edge processing and secure end-to-end cloud server connectivity. Recent work focuses on wireless UAS networking [31], [32], [55]–[58], exploring dynamic UAS communication networks, but at shorter distances (e.g., within VLOS) and single data types. In contrast, our work is supported by real-world UAS flights at distances exceeding VLOS, a relatively unexplored research area. We conducted a long range UAS measurement study in [35], but without prediction or error models, or an application. In contrast, Chimera focuses on optimization of sensor data transmission for long-range UAS flights. There has been work on modeling UAS communication channels for data transmission [6], [44], [59], reinforcing our theoretical observations (e.g., the effect of distance to throughput). The papers [4], [5], [10], [13] provide high-level information about challenges and open problems in UAS communication networks, but lack flight data or working solutions to the problems.

**Internet video:** Much recent work has focused on efficiently delivering Internet video through the design of ABR algorithms [48], [60]–[64], considering video conferencing challenges [45], and exploring the simultaneous transfer of live and time-shifted video [65]. These works do not account for challenges unique to UAS settings, which is our focus.

## VII. Conclusion

In this paper we have made the following contributions: first, we have shown through a characterization of two real-world UAS flight datasets that there is significant opportunity to optimize data transmission in UAS settings by exploiting knowledge of UAS flight paths. Second, we have presented Chimera, a system for simultaneously transmitting heterogenous sensor data with different timeliness and reliability requirements by taking advantage of UAS flight path information. As part of Chimera, we have developed models grounded in real-world data that relate UAS network throughput to flight path. Chimera uses an optimal control framework, performing online optimization and augmented with a robust prediction and error model in planning its heterogeneous data transmissions. Third, we evaluated Chimera using a combination of simulation and emulation experiments, and with multiple real-world flight traces and synthetic traces generated using a methodology that we have developed and validated. Our results show that Chimera offers significant improvement over an approach that does not exploit flight path information. Specifically, in evaluation on our emulation test-bed, Chimera is able to reduce penalties related to dropped SAR image transmissions by $72.4\% - 100\%$ relative to *Flight Agnostic* and achieve comparable video qualities of $90.5\%$, with only a minimal increase in SAR images dropped, compared to a perfect Oracle (optimal) scheme that knows future throughput.

**Acknowledgements:** We thank Nathan Cohen for help in our emulations. We also express gratitude to Aero Systems West for their assistance with UAS flight testing for our data collection. This work was partially supported by the National Science Foundation (NSF) Award ICE-T 1836889, Cisco, and NSF CAREER award 1653648. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF or Cisco.

# References

[1] F. A. A. (FAA), "FAA aerospace forecast fiscal years 2021-2041," Tech. Rep., 2021.

[2] Deloitte, "Managing the evolving skies. Unmanned aircraft system traffic management (UTM), the key enabler," 2019. [Online]. Available: https://www2.deloitte.com/global/en/pages/energy-and-resources/articles/managing-evolving-skies.html

[3] Y. Chen, H. Zhang, and M. Xu, "The coverage problem in UAV network: A survey," in *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE, 2014, pp. 1–5.

[4] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[5] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[6] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5g and beyond," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2327–2375, 2019.

[7] A. Hanscom and M. Bedford, "Unmanned aircraft system (UAS) service demand 2015–2035, literature review & projections of future usage," *Res. Innov. Technol. Admin., US Dept. Transp., Washington, DC, USA*, 2013.

[8] R. Dunkel, R. Saddler, and A. Doerry, "Use of unmanned SAR and EO/IR sensor suites for monitoring wildfires," in *Radar Sensor Technology XXI*, vol. 10188. International Society for Optics and Photonics, 2017, p. 101881F.

[9] Y. Ban, P. Zhang, A. Nascetti, A. R. Bevington, and M. A. Wulder, "Near real-time wildfire progression monitoring with sentinel-1 SAR time series and deep learning," *Scientific Reports*, vol. 10, no. 1, pp. 1–15, 2020.

[10] O. K. Sahingoz, "Mobile networking with UAVs: Opportunities and challenges," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2013, pp. 933–941.

[11] S. Kacianka and H. Hellwagner, "Adaptive video streaming for UAV networks," in *Proceedings of the 7th ACM International Workshop on Mobile Video*, 2015, pp. 25–30.

[12] F. L. Templin, R. Jain, G. Sheffield, P. Taboso-Ballesteros, and D. Ponchak, "Requirements for an integrated uas cns architecture," in *2017 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2017, pp. 2E4–1.

[13] H. Nawaz, H. M. Ali, and A. A. Laghari, "UAV communication networks issues: a review," *Archives of Computational Methods in Engineering*, pp. 1–21, 2020.

[14] C. of Federal Regulations (CFR), "14 CFR: Aeronautics and Space," 2019.

[15] S. Mitchell, "Unmanned aircraft flies first U.S. beyond-line-of-sight mission," 2019. [Online]. Available: https://news.uaf.edu/unmanned-aircraft-flies-first-u-s-beyond-line-of-sight-mission/

[16] U. Airspace, "Drone delivery operations underway in 27 countries," 2019. [Online]. Available: https://www.unmannedairspace.info/latest-news-and-information/drone-delivery-operations-underway-in-26-countries

[17] IMSAR, "Synthetic aperture radar," 2020. [Online]. Available: https://www.imsar.com/portfolio-posts/synthetic-aperture-radar/

[18] N. Search and R. Committee, "Unmanned aircraft system (UAS) search and rescue addendum to the national search and rescue supplement to the international aeronautical and maritime search and rescue manual," Tech. Rep., 2016.

[19] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019.

[20] S. Manfreda, M. F. McCabe, P. E. Miller, R. Lucas, V. Pajuelo Madrigal, G. Mallinis, E. Ben Dor, D. Helman, L. Estes, G. Ciraolo *et al.*, "On the use of unmanned aerial systems for environmental monitoring," *Remote sensing*, vol. 10, no. 4, p. 641, 2018.

[21] P. Hügler, F. Roos, M. Schartel, M. Geiger, and C. Waldschmidt, "Radar taking off: New capabilities for UAVs," *IEEE Microwave Magazine*, vol. 19, no. 7, pp. 43–53, 2018.

[22] G. Stimson, *Introduction to Airborne Radar*. SciTech Publishing Inc, 1998.

[23] A. W. Doerry and F. M. Dickey, "Synthetic aperture radar," *Optics and photonics news*, vol. 15, no. 11, pp. 28–33, 2004.

[24] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 1, pp. 6–43, 2013.

[25] S. K. Chaturvedi, R. Sekhar, S. Banerjee, and H. Kamal, "Comparative review study of military and civilian unmanned aerial vehicles (UAVs)," *INCAS Bulletin*, vol. 11, no. 3, pp. 183–198, 2019.

[26] B. P. Rohman, M. B. Andra, H. F. Putra, D. H. Fandiantoro, and M. Nishimoto, "Multisensory surveillance drone for survivor detection and geolocalization in complex post-disaster environment," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 9368–9371.

[27] "Industrial drones and the rise of unmanned aerial systems (UAS)," 2021. [Online]. Available: https://www.trihydro.com/news/news-details/2019/05/28/using-drones-for-safer-dam-inspections-and-evaluations

[28] "Primoco UAV integrates synthetic aperture radar," 2020. [Online]. Available: https://www.eenewseurope.com/news/primoco-uav-integrates-synthetic-aperture-radar

[29] X. Wang, A. Chowdhery, and M. Chiang, "Skyeyes: adaptive video streaming from uavs," in *Proceedings of the 3rd Workshop on Hot Topics in Wireless*. ACM, 2016, pp. 2–6.

[30] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *IEEE ICDCS*, 2017, pp. 308–318.

[31] S. Hayat, C. Bettstetter, A. Fakhreddine, R. Muzaffar, and D. Emini, "An experimental evaluation of LTE-A throughput for drones," in *DroNet 2019*, 2019, pp. 3–8.

[32] M. Asadpour, D. Giustiniano, K. A. Hummel, and S. Heimlicher, "Characterizing 802.11n aerial communication," in *MobiHoc 2013*, 2013, pp. 7–12.

[33] R. Gangula, O. Esrafilian, D. Gesbert, C. Roux, F. Kaltenberger, and R. Knopp, "Flying rebots: First results on an autonomous UAV-based lte relay using open airinterface," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2018, pp. 1–5.

[34] A. Baltaci, M. Klügel, F. Geyer, S. Duhovnikov, V. Bajpai, J. Ott, and D. Schupke, "Experimental UAV data traffic modeling and network performance analysis," in *Proceedings of the 40th IEEE International Conference on Computer Communications*, 2021.

[35] R. Shirey, S. Rao, and S. Sundaram, "Measuring fixed wing UAS networks at long range," in *Proceedings of the 6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2020, pp. 1–6.

[36] A. Filippone, *Flight performance of fixed and rotary wing aircraft*. Elsevier, 2006.

[37] Trellisware, "Tw-900/950 tsm shadow radio," 2020. [Online]. Available: https://www.trellisware.com/wp-content/uploads/2020/09/TW-900-950-TSM-Shadow-Product-Datasheet.pdf

[38] P. Systems, "Mpu specification sheet," 2016. [Online]. Available: http://www.persistentsystems.com/pdf/MPU4_SpecSheet.pdf

[39] S. Temel and I. Bekmezci, "Scalability analysis of flying ad hoc networks (fanets): A directional antenna approach," in *2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2014, pp. 185–187.

[40] Z. Zheng, A. K. Sangaiah, and T. Wang, "Adaptive communication protocols in flying ad hoc network," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 136–142, 2018.

[41] iPerf, 2019. [Online]. Available: https://iperf.fr

[42] H. T. Friis, "A note on a simple transmission formula," *proc. IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[43] J. L. Volakis, *Antenna engineering handbook*. McGraw-Hill Education, 2007.

[44] F. Ono, H. Ochiai, and R. Miura, "A wireless relay network based on unmanned aircraft system with rate optimization," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7699–7708, 2016.

[45] 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 *15th {USENIX} Symposium*

*on Networked Systems Design and Implementation ({NSDI} 18)*, 2018, pp. 267–282.

[46] R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, J. Mickens, and H. Balakrishnan, "Mahimahi: Accurate record-and-replay for http," in *USENIX*, 2015, pp. 417–429.

[47] R. B. D'Agostino and M. A. Stevens, *Goodness-of-fit-techniques*. Routledge, 2017.

[48] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *ACM SIGCOMM*, 2015, pp. 325–338.

[49] Google, "Recommended upload encoding settings," 2021. [Online]. Available: https://support.google.com/youtube/answer/1722171

[50] R. Shirey, S. Rao, and S. Sundaram, "Optimizing quality of experience for long-range UAS video streaming," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQoS)*. IEEE, 2021.

[51] P. Sun and A. Boukerche, "Performance modeling and analysis of a uav path planning and target detection in a UAV-based wireless sensor network," *Computer Networks*, vol. 146, pp. 217–231, 2018.

[52] A. Lamine, F. Mguis, H. Snoussi, and K. Ghedira, "Coverage optimization using multiple unmanned aerial vehicles with connectivity constraint," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1361–1366.

[53] L. C. B. da Silva, R. M. Bernardo, H. A. de Oliveira, and P. F. F. Rosa, "Unmanned aircraft system coordination for persistent surveillance with different priorities," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2017, pp. 1153–1158.

[54] A. Rajakaruna, A. Manzoor, P. Porambage, M. Liyanage, M. Ylianttila, and A. Gurtov, "Enabling end-to-end secure connectivity for low-power IoT devices with UAVs," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*. IEEE, 2019, pp. 1–6.

[55] M. Asadpour, D. Giustiniano, and K. A. Hummel, "From ground to aerial communication: Dissecting wlan 802.11n for the drones," in *WiNTECH 2013*, 2013, pp. 25–32.

[56] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility," in *INFOCOM 2013*. IEEE, 2013, pp. 120–124.

[57] S. Hayat, E. Yanmaz, and C. Bettstetter, "Experimental analysis of multipoint-to-point UAV communications with ieee 802.11n and 802.11ac," in *PIMRC 2015*. IEEE, 2015, pp. 1991–1996.

[58] G. Geraci, A. Garcia-Rodriguez, L. G. Giordano, D. López-Pérez, and E. Björnson, "Understanding uav cellular communications: From existing networks to massive mimo," *IEEE Access*, vol. 6, pp. 67 853–67 865, 2018.

[59] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in uav enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, 2017.

[60] Z. Akhtar, Y. S. Nam, R. Govindan, S. Rao, J. Chen, E. Katz-Bassett, B. Ribeiro, J. Zhan, and H. Zhang, "Oboe: auto-tuning video abr algorithms to network conditions," in *ACM SIGCOMM*, 2018, pp. 44–58.

[61] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with Pensieve," in *ACM SIGCOMM*, 2017, pp. 197–210.

[62] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *ACM SIGCOMM*, 2016, pp. 272–285.

[63] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *IEEE INFOCOM*, 2016, pp. 1–9.

[64] C. Qiao, J. Wang, and Y. Liu, "Beyond QoE: Diversity adaption in video streaming at the edge," in *IEEE ICDCS*, 2019, pp. 317–326.

[65] D. Ray, J. Kosaian, K. Rashmi, and S. Seshan, "Vantage: optimizing video upload for time-shifted viewing of social live streams," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 380–393.