Yi Wang, Markus Krantzik, Lin Xiang, and Anja Klein, “Joint Communication and Computing Optimization for Digital Twin Synchronization with Aerial Relay” in *IEEE International Conference on Communications (ICC), Denver, Colorado, June 2024.*

Joint Communication and Computing Optimization for Digital Twin Synchronization with Aerial Relay

Yi Wang, Markus Krantzik, Lin Xiang, and Anja Klein
Communications Engineering Lab, Technische Universität Darmstadt, Germany
Emails: {y.wang, l.xiang, a.klein}@nt.tu-darmstadt.de

***Abstract*—Digital twin (DT) applications usually need to be synchronized in real time with the state of the physical system (PS). This process includes both synchronizing the data collected by sensors in the PS to a server and updating the DT at the server in adaptation to the dynamics of the PS. However, communicating with power- and rate-limited sensors and processing high-volume sensed data with limited computing resources present significant challenges for DT synchronization. In this paper, we tackle both bottlenecks by proposing a joint communication and computing design for DT synchronization. In particular, we exploit buffering at an aerial cluster head of the sensors and enable buffer-aided (BA) relaying to increase the communication throughput of the sensors during data synchronization. Moreover, we adopt a novel *stream computing* scheme, which allows for DT updating in parallel with data synchronization, to accelerate DT synchronization. To maximize the performance of the proposed approach, we jointly optimize the trajectory of the aerial relay and the communication and computing resource allocation for minimizing the DT synchronization time. The formulated problem is a mixed-integer nonconvex problem, which is generally intractable. To tackle this challenge, we propose a low-complexity two-layer iterative suboptimal algorithm. Our simulation results show that the DT synchronization time can be significantly reduced by up to 43.8%, through stream computing and the joint optimization of the relay's trajectory and the communication/computing resource allocation.**

I. Introduction

Digital twins (DTs) define a novel paradigm for building the future digital society. Leveraging the capabilities of the sixth-generation (6G) communication systems, DTs can generate precise, real-time, and interactive digital replicas of physical systems (PSs) in the real world. Real-time DTs can even enable extended reality services for various sectors including manufacturing, aviation, and smart cities [1]. However, the DTs need to be seamlessly synchronized with the state of the PSs in order to capture the dynamics of the PSs. This process, known as *DT synchronization*, presents a key research challenge for 6G as it requires not only low-latency communication between the PS and the DT for the data synchronization phase, but also agile computing using this data for the subsequent DT updating phase [2].

So far, several computing and communication solutions have been proposed in the existing literature [3]–[5] to enhance both phases of DT synchronization. In [3], the average cost, including time and energy consumption, required for data synchronization between the PSs and the DTs was minimized by optimizing network selection and power allocation. Given the computation-intensive nature of updating the DT, the exclusive focus on the data synchronization time may lead to strictly suboptimal performance. In [4] and [5], the communication and computing times required for DT synchronization were jointly considered. The authors of [4] considered DT synchronization subject to a constraint on the maximum synchronization time. Besides, the authors of [5] optimized the association of PSs with edge servers selected for hosting DTs to minimize the maximum synchronization time among multiple DTs.

The aforementioned works [3]–[5] assumed single-hop data transmission from the PS to the DT. When the low-power sensors are widely distributed, single-hop transmission may no longer meet the communication requirements of DT data synchronization. To relieve the communication bottleneck, relay-aided data synchronization for DT was recently investigated in [6], which optimized relay selection in heterogeneous networks to minimize the transmission time for data synchronization. However, the computing time for DT updating was not considered in [6]. Moreover, data synchronization and DT updating phases are assumed to be *non-overlapping* over time in [3]–[6], i.e., DT updating commences only after completion of data synchronization. This unnecessarily prolongs the DT synchronization.

In this paper, we consider a novel joint communication and computing design for DT synchronization enabled by an aerial relay, such as unmanned aerial vehicle (UAV) and airship, in 6G networks. To mitigate the communication bottleneck for the rate- and power-limited sensors, we consider buffering at the aerial relay and perform buffer-aided (BA) relaying [7], [8]. By buffering, the relay can temporarily store the data, and only forward it when channel condition becomes favorable. Additionally, by changing its position, the aerial relay can proactively adjust the channel conditions to further increase the communication capacity [8].

To overcome the computation bottleneck, we adopt a novel *stream computing* scheme [9]. Given the stream characteristics of the data generated from IoT devices, data can start being processed in segments using stream computing, soon after one segment is received, without waiting for the completion of data synchronization. This allows parallel DT updating and data synchronization, i.e., the two phases are *overlapping in time*. Leveraging both, the BA aerial relay and stream computation, we further optimize the relay's trajectory and the communication and computing resource allocation to accelerate the DT synchronization. To our knowledge, neither BA relaying nor stream computing has ever been explored in the literature for DT synchronization. Our contributions are:

- We consider stream computing and BA relaying with an aerial relay to accelerate DT synchronization. To maximize the performance, we jointly optimize the relay's trajectory and the communication and computing resource allocation to minimize the DT synchronization time.
- The formulated optimization problem is a nonconvex mixed-integer nonlinear problem, which is generally intractable. To facilitate a real-time solution, we propose a two-layer iterative suboptimal algorithm with low computational complexity. We employ successive convex approximation (SCA) to solve the inner-layer nonconvex

This work has been funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY center and has been supported by the BMBF project Open6GHub under grant 16KISKO14 and by DAAD with funds from the German Federal Ministry of Education and Research (BMBF).

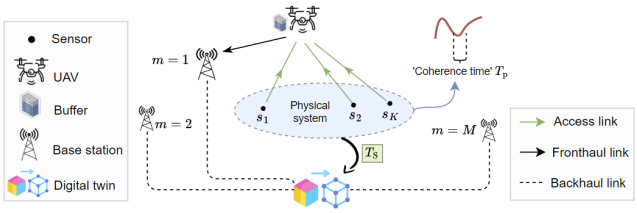

Fig. 1. System diagram of DT synchronization enabled by a BA aerial relay.

problem. The inner-layer solution is further utilized in the outer layer for updating the DT synchronization time.
- Simulation results show that the DT synchronization time can be significantly reduced by enabling stream computing and joint optimization of the relay's trajectory and the resource allocation for communication/computing.

Note that our problem formulation and optimization solution are also applicable to mobile and fixed terrestrial relays with and without buffer. In the remainder of this paper, Section II introduces the system model of DT synchronization. Sections III and IV present the problem formulation and proposed solution, respectively. Section V evaluates the performance of the proposed scheme via simulations. Finally, Section VI concludes the paper.

II. System Model

This section presents the system model for DT synchronization assisted by an aerial relay, including BA aerial relaying for data synchronization and DT updating with stream computing.

A. Aerial-Relay-Aided DT Synchronization

As depicted in Figure 1, a cloud server on the Internet hosts a DT, i.e., a real-time digital replica, for a time-varying PS, such as urban areas, agricultural landscapes, and forests. Thereby, K sensors are deployed at position $\boldsymbol{u}_k^{\mathrm{s}}$, $k \in \mathcal{K} \triangleq \{1, \cdots, K\}$ to collectively gather data from the PS and transmit them to the server via a wireless network consisting of M BSs. Let $\boldsymbol{u}_m^{\mathrm{BS}}$ be the position of BS $m \in \mathcal{M} = \{1, \cdots, M\}$. The BSs are connected to the Internet via high-speed wirelines such as optical fiber. The server updates the DT upon receiving the data updates. Due to limited transmit power and remote deployment, direct connections between sensors and BSs can be obstructed or significantly attenuated. Therefore, the sensors have to first aggregate their data at a cluster head such as a UAV. The cluster head then forwards the data to the BSs, as a relay.

Due to the inertia of any PS, we assume that the dynamics in the considered PS follow a low-pass filtered process with limited bandwidth B_{p}. According to Shannon's sampling theorem, the DT should be accurately synchronized every $T_{\mathrm{p}} = 1/B_{\mathrm{p}}$ seconds in order for the DT to reliably capture the variations of the PS. In each DT synchronization round, each sensor k needs to update D_k bits of data to the server for updating the DT. In this paper, we focus on modeling and optimizing the system for a typical synchronization round. Our modeling and optimization approach can be applied to other DT synchronization rounds in a straightforward manner.

To facilitate trajectory design for the aerial relay, a discrete-time system with time index t is applied. Each time slot has duration $\Delta > 0$. We assume that Δ is sufficiently small, i.e., shorter than the channel coherence time of all communication links. This allows to approximate the relay's position and channel conditions as constants within each time slot.

B. Buffer-Aided Aerial Relaying for Data Synchronization

The relay is equipped with a finite buffer of size $Q_{\max}$. The buffer enables the relay to temporarily store data packets from the sensors, which can be forwarded to the BSs when the channel conditions become favorable. Meanwhile, aerial relay can strategically change its positions to adjust channel conditions while performing BA relaying to further enlarge the performance gains. BA aerial relaying has been shown to achieve significantly higher throughput than non-BA aerial relaying in [8]. For convenience, the sensor-to-relay, relay-to-BS, and BS-to-DT links are referred to as the access, fronthaul, and backhaul links, respectively.

1) Channel Models and Achievable Data Rates: We consider line-of-sight (LoS)-dominant channels with Rician channel fadings for both access and fronthaul links. Let $\boldsymbol{u}[t]$ denote the relay's position in time slot t. Moreover, $d_k^{\mathrm{a}}[t] \triangleq \|\boldsymbol{u}[t] - \boldsymbol{u}_k^{\mathrm{s}}\|$ and $d_m^{\mathrm{f}}[t] \triangleq \left\|\boldsymbol{u}[t] - \boldsymbol{u}_m^{\mathrm{BS}}\right\|$ denote the distances from the relay to sensor k and BS m, respectively. The channel gains in time slot t in access and fronthaul links, denoted by $H_k^{\mathrm{a}}[t]$ and $H_m^{\mathrm{f}}[t]$, respectively, are modeled as

$$H_k^{\mathrm{a}}[t] = \sqrt{A_0} \cdot \left(d_k^{\mathrm{a}}[t]\right)^{-\alpha/2} \cdot h_k^{\mathrm{a}}[t], \tag{1}$$
$$H_m^{\mathrm{f}}[t] = \sqrt{A_0} \cdot \left(d_m^{\mathrm{f}}[t]\right)^{-\alpha/2} \cdot h_m^{\mathrm{f}}[t].$$

In (1), $A_0(d_k^{\mathrm{a}})^{-\alpha}$ and $A_0(d_m^{\mathrm{f}})^{-\alpha}$ represent the path losses over the access and fronthaul links in time slot t, respectively, where $\alpha \geq 2$ is the path loss exponent. The channel fadings $h_k^{\mathrm{a}}[t] \in \mathbb{C}$ and $h_m^{\mathrm{f}}[t] \in \mathbb{C}$ are modeled as [10]

$$h_k^{\mathrm{a}}[t] = \sqrt{\tfrac{\kappa^{\mathrm{a}}}{\kappa^{\mathrm{a}}+1}}\sigma_k^{\mathrm{a}} e^{j\theta_k^{\mathrm{a}}[t]} + \sqrt{\tfrac{1}{\kappa^{\mathrm{a}}+1}}\mathcal{CN}(0, (\sigma_k^{\mathrm{a}})^2), \tag{2}$$
$$h_m^{\mathrm{f}}[t] = \sqrt{\tfrac{\kappa^{\mathrm{f}}}{\kappa^{\mathrm{f}}+1}}\sigma_m^{\mathrm{f}} e^{j\theta_m^{\mathrm{f}}[t]} + \sqrt{\tfrac{1}{\kappa^{\mathrm{f}}+1}}\mathcal{CN}(0, (\sigma_m^{\mathrm{f}})^2).$$

The first and the second terms of (2) describe the LoS and the non-LoS (NLoS) components of propagation paths, respectively. $\kappa^{\mathrm{a}\backslash\mathrm{f}} \geq 0$ denotes the energy ratio between the LoS and the NLoS components. Moreover, $\sigma_k^{\mathrm{a}} \triangleq \mathbb{E}(|h_k^{\mathrm{a}}[t]|)$ and $\sigma_m^{\mathrm{f}} \triangleq \mathbb{E}(|h_m^{\mathrm{f}}[t]|)$ are the standard deviations. The phases of access and fronthaul links, denoted as $\theta_k^{\mathrm{a}}[t]$ and $\theta_m^{\mathrm{f}}[t]$, respectively, are uniformly distributed in $[0, 2\pi]$ and are independent across different paths. Based on (2), both $|h_k^{\mathrm{a}}[t]|^2$ and $|h_m^{\mathrm{f}}[t]|^2$ follow the Rician distribution.

The system has a total bandwidth of B, which is divided into small, non-overlapping frequency sub-channels using frequency-division multiple access (FDMA). Sub-channels with bandwidth $b_k^{\mathrm{a}}[t]$ and $b_{k,m}^{\mathrm{f}}[t]$, orthogonal to each other, are assigned to sensor k for access and fronthaul communication, respectively. Let $r_k^{\mathrm{a}}[t]$ and $r_{k,m}^{\mathrm{f}}[t]$ be the achievable data rates over access and fronthaul links, respectively. According to Shannon's channel capacity formula [10], we have

$$r_k^{\mathrm{a}}[t] = b_k^{\mathrm{a}}[t] \cdot \log_2\left(1 + \frac{|H_k^{\mathrm{a}}[t]|^2 \cdot p_k^{\mathrm{a}}[t]}{N_0^{\mathrm{a}} \cdot b_k^{\mathrm{a}}[t]}\right), \tag{3}$$
$$r_{k,m}^{\mathrm{f}}[t] = b_{k,m}^{\mathrm{f}}[t] \cdot \log_2\left(1 + \frac{|H_m^{\mathrm{f}}[t]|^2 \cdot p_{k,m}^{\mathrm{f}}[t]}{N_{0,m}^{\mathrm{f}} \cdot b_{k,m}^{\mathrm{f}}[t]}\right),$$

where $p_k^{\mathrm{a}}[t]$ and $p_{k,m}^{\mathrm{f}}[t]$ denote the transmit powers of sensor k and of the relay for forwarding sensor k's data to BS m in time slot t, respectively. N_0^{a} and $N_{0,m}^{\mathrm{f}}$ are the noise power spectral densities at the relay and BS m, respectively.

The data received at the BSs can be stored before being

transmitted to the server over the high-speed backhaul links. We consider a fixed-latency backhaul communication model with instantaneous data rate $r^{\mathrm{b}}_{k,m}[t] = r^{\mathrm{f}}_{k,m}[t-\tau]$, where $\tau \geq 1$ denotes the latency incurred for data transmission over the backhaul link, including one additional time slot required for decoding and buffering.

2) Buffer Status Evolution: The total buffer $Q_{\max}$ at the relay is dynamically allocated to the K sensors. Let $q_k[t]$ be the amount of data buffered at the relay, also referred to as the queue length, for sensor k in time slot t. We have

$$q_k[t] = \max\left\{q_k[t-1] - \sum\nolimits_{m=1}^{M} r^{\mathrm{f}}_{k,m}[t]\Delta, 0\right\} + r^{\mathrm{a}}_k[t]\Delta, \quad (4)$$

where we require $\sum_{k=1}^{K} q_k[t] \leq Q_{\max}$ to avoid buffer overflow. The $\max\{\cdot, 0\}$ operator ensures that the volume of data forwarded by the relay at time t does not exceed the amount received by it until time $t-1$, thereby maintaining causality in the data transmission. After the data has been forwarded in time slot t, its occupied buffer space is immediately freed up.

C. Stream Computing for DT Modeling

1) Computing Model: Unlike batch computing that requires receiving the entire data before processing, stream computing tackles data in small segments, such as image frames. Stream computing facilitates real-time data processing, for DT updating, in parallel with data transmission. For a tractable design, we consider data processing conducted on infinitesimal data segments. Let $r^{\mathrm{c}}_{k,m}[t]$ be the processing speed in bits per second in time slot t for the data received from sensor k through BS m and $s_k[t]$ denote the amount of processed data in bits from sensor k in time slot t. We have

$$s_k[t] = \sum\nolimits_{m=1}^{M} r^{\mathrm{c}}_{k,m}[t]\Delta, \quad (5)$$

where the processing speed $r^{\mathrm{c}}_{k,m}[t]$ is limited by both the amount of received data and the total computing resources. We will show in Sec. V that our result provides a tight performance upper bound for stream computing with fixed-size segments.

2) Computing Resource Consumption: Let c_k denote the computing complexity in CPU cycles per bit required for processing data from sensor k during DT updating [11]. The total computation resources required in time slot t for processing data from all sensors is given by

$$c[t] = \sum\nolimits_{k=1}^{K} s_k[t] \cdot c_k. \quad (6)$$

We require $c[t]/\Delta \leq C_{\mathrm{DT}}$, where C_{DT} is the available CPU frequency at the server.

III. Problem Formulation

To maximize the performance of the proposed joint communication and computation design, in this section, we optimize the trajectory and the resource (bandwidth, transmit power, buffer and processing speed) allocation for minimization of the DT synchronization time, denoted by $T_{\mathrm{S}} \in \mathbb{N}$. T_{S} includes the time for both data transmission from the sensors to the server and DT updating using this data at the server. We assume that the initial position $\boldsymbol{u}[0]$ of the relay and the channel conditions for the fronthaul and access links are known. The optimization problem is formulated as

$$\mathcal{P}_1: \min_{\boldsymbol{u}[t], \boldsymbol{P}[t], \boldsymbol{B}[t], \boldsymbol{R}^{\mathrm{c}}[t]} T_{\mathrm{S}} \quad (7)$$

$$\text{s.t. } \mathrm{C}_1: \|\boldsymbol{u}[t] - \boldsymbol{u}[t-1]\|/\Delta \leq V_{\max}, \quad t \in \mathcal{T}$$

$$\mathrm{C}_2: p^{\mathrm{a}}_k[t] \leq P^{\mathrm{a}}_{k,\max}, \quad \forall t, k \in \mathcal{K}$$

$$\mathrm{C}_3: \sum\nolimits_{k=1}^{K}\sum\nolimits_{m=1}^{M} p^{\mathrm{f}}_{k,m}[t] \leq P^{\mathrm{f}}_{\max}, \quad \forall t$$

$$\mathrm{C}_4: \sum\nolimits_{k=1}^{K}\left(b^{\mathrm{a}}_k[t] + \sum\nolimits_{m=1}^{M} b^{\mathrm{f}}_{k,m}[t]\right) \leq B, \quad \forall t$$

$$\mathrm{C}_5: r^{\mathrm{a}}_k[t] \geq b^{\mathrm{a}}_k[t] \cdot E^{\mathrm{a}}_{\min}, \quad \forall t, \ \forall k$$

$$\mathrm{C}_6: r^{\mathrm{f}}_{k,m}[t] \geq b^{\mathrm{f}}_{k,m}[t] \cdot E^{\mathrm{f}}_{\min}, \quad \forall t, \ \forall k, \ m \in \mathcal{M}$$

$$\mathrm{C}_7: \sum\nolimits_{t=1}^{i}\left(r^{\mathrm{a}}_k[t-1] - \sum\nolimits_{m=1}^{M} r^{\mathrm{f}}_{k,m}[t]\right)\Delta \geq 0, \ \forall i, \ \forall k$$

$$\mathrm{C}_8: \sum_{t=1}^{i}\sum_{k=1}^{K}\left(r^{\mathrm{a}}_k[t-1] - \sum_{m=1}^{M} r^{\mathrm{f}}_{k,m}[t]\right)\Delta \leq Q_{\max}, \ i \in \mathcal{T}$$

$$\mathrm{C}_9: \sum\nolimits_{t=1}^{i}\left(r^{\mathrm{b}}_{k,m}[t-1] - r^{\mathrm{c}}_{k,m}[t]\right)\Delta \geq 0, \forall k, \forall m, \forall i$$

$$\mathrm{C}_{10}: c[t]/\Delta \leq C_{\mathrm{DT}}, \ \forall t$$

$$\mathrm{C}_{11}: \sum\nolimits_{t=1}^{T_{\mathrm{S}}}\sum\nolimits_{m=1}^{M} r^{\mathrm{c}}_{k,m}[t]\Delta \geq D_k, \ \forall k,$$

where $\boldsymbol{u}[t]$ is the trajectory of the aerial relay, $\boldsymbol{P}[t] \triangleq \{p^{\mathrm{a}}_k[t], p^{\mathrm{f}}_{k,m}[t] | \ \forall k, m\}$ is the nonnegative power allocation, $\boldsymbol{B}[t] \triangleq \{b^{\mathrm{a}}_k[t], b^{\mathrm{f}}_{k,m}[t] | \ \forall k, m\}$ is the nonnegative bandwidth allocation, and $\boldsymbol{R}^{\mathrm{c}}[t] \triangleq \{r^{\mathrm{c}}_{k,m}[t] | \ \forall k, m\}$ is the processing speed for time $t \in \mathcal{T} \triangleq \{1, \cdots, T_{\mathrm{S}}\}$. Constraint C_1 limits the relay's maximum velocity by $V_{\max}$. C_2 and C_3 ensure that the transmit power of sensor k and the relay do not exceed $P^{\mathrm{a}}_{k,\max}$ and $P^{\mathrm{f}}_{\max}$, respectively. C_4 limits the total allocated bandwidth by the system bandwidth, B. In C_3 and C_4, the relay is allowed to communicate with multiple BSs simultaneously. However, our simulation results suggest that the relay always communicate with the closest BS at one time. C_5 and C_6 guarantee a minimum spectral efficiency $E^{\mathrm{a}}_{\min}$ and $E^{\mathrm{f}}_{\min}$ in time slot t for each active access and fronthaul link, respectively; otherwise, C_5 and C_6 can be ignored for inactive access or fronthaul links. C_7 ensures data causality for BA relaying, i.e., the total data transmitted in the fronthaul links within time period $[1, i]$ cannot exceed that received from the access link within time period $[1, i-1]$, for any $i \in \mathcal{T}$. C_8 limits the amount of data stored in the buffer to prevent buffer overflow. C_9 and C_{10} guarantee that for DT updating based on stream computing, the data volume being processed at the server can neither exceed that being received nor utilize computing resources beyond the server's CPU frequency. Finally, C_{11} requires the data processing for each sensor k to be completed by time T_{S}.

Problem $\mathcal{P}_1$ is a mixed-integer nonconvex problem due to the integer variable T_{S} in the objective function and constraint C_{11}, as well as the nonconvex constraints $C_5 - C_9$. Moreover, T_{S} is coupled with other optimization variables, which further complicates the solution of $\mathcal{P}_1$. This type of problem is generally considered intractable [14]. To facilitate a real-time solution of $\mathcal{P}_1$, we propose in Section IV a novel low-complexity two-layer iterative suboptimal algorithm.

IV. Problem Solution

In this section, problem $\mathcal{P}_1$ is first decomposed into two-layer subproblems to decouple the optimization of T_{S} and other variables [8]. Subsequently, to facilitate the solution, the nonconvex constraints in problem $\mathcal{P}_1$ are transformed into

equivalent difference of convex (DC) forms and then tackled using SCA.

A. Proposed Iterative Two-Layer Solution

We propose a novel two-layer decomposition for $\mathcal{P}_1$. The DT synchronization time $T_{\rm S}$ is updated in the outer layer, using information provided by the inner layer. In the inner layer, the objective is to maximize the *progress* of DT synchronization for a $T_{\rm S}$ specified by the outer layer. Here, the progress of DT synchronization is indicated using an auxiliary variable $x \geq 0$. In particular, $D_k \cdot x$ represents the volume of data transmitted from the sensor k and subsequently processed by the server for updating DT. For each $T_{\rm S}$ determined by the outer layer, we maximize the progress of DT synchronization via the following inner-layer problem,

$$\mathcal{P}_2: \max_{\boldsymbol{u}[t],\boldsymbol{P}[t],\boldsymbol{B}[t],\boldsymbol{R}^{\rm c}[t]} x \tag{8}$$
$$\text{s.t. } \mathrm{C}_1 - \mathrm{C}_{10}, \tilde{\mathrm{C}}_{11}: \sum\nolimits_{t=1}^{T_{\rm S}} \sum\nolimits_{m=1}^{M} r_{k,m}^{\rm c}[t]\Delta \geq D_k \cdot x, \ \forall k.$$

$\mathcal{P}_2$ in turn maximizes the *average throughput* of DT synchronization $\sum_{k=1}^{K} D_k \cdot x / T_{\rm S}$, which measures the average speed of DT synchronization. Similar to [13, Theorem 1], we can show that the defined two-layer subproblems are equivalent to $\mathcal{P}_1$, but the details are omitted due to limited page space.

The proposed two-layer solution is summarized in Algorithm 1. Let T_0 be the starting point of $T_{\rm S}$. Moreover, x^* is the maximum progress of DT synchronization for a specified $T_{\rm S}$, obtained from the inner-layer problem $\mathcal{P}_2$. If $x^* \geq 1$, more than $\sum_{k=1}^{K} D_k$ amount of data can be transmitted to and processed by the server within $T_{\rm S}$. In this case, $T_{\rm S}$ and x^* define upper bounds for the DT synchronization time and its associated progress, $T_{\rm ub}$ and $x_{\rm ub}$, respectively. However, if $x^* < 1$, the DT synchronization cannot be finished within the specified time $T_{\rm S}$, such that $T_{\rm S}$ and x^* provide lower bounds for DT synchronization time $T_{\rm lb}$ and progress $x_{\rm lb}$. x^* and its bounds $x_{\rm lb}$ and $x_{\rm ub}$ can be further utilized to improve the selection of $T_{\rm S}$ for the next iteration, denoted as $T_{\rm next}$, in Algorithm 1. To this end, we estimate the time required for synchronizing the remaining $\sum_{k=1}^{K} D_k \cdot (1 - x_{\rm lb})$ amount of data using the average synchronization throughput derived from $\sum_{k=1}^{K} D_k \cdot x_{\rm lb} / T_{\rm lb}$ or $\sum_{k=1}^{K} D_k (x_{\rm ub} - x_{\rm lb}) / (T_{\rm ub} - T_{\rm lb})$. In case $x_{\rm ub}$ and $T_{\rm ub}$ are available, the second estimate is used as a more accurate estimate of throughput for synchronizing the remaining data. This estimated time for synchronizing the remaining data is then added to $T_{\rm lb}$ for determining $T_{\rm next}$. The search procedure continues until $T_{\rm S}$ in the current iteration equals $T_{\rm next}$.

It remains to solve the inner-layer problem $\mathcal{P}_2$ with nonconvex constraints $C_5 - C_9$. To this end, below we transform $\mathcal{P}_2$ into an equivalent DC problem and tackle it using SCA.

B. Transformation of Problem $\mathcal{P}_2$

Let us define $g\left(b[t], r[t]\right) \triangleq b[t]\left(\exp\left(\frac{r[t]\ln 2}{b[t]}\right) - 1\right)$, as a perspective function of the exponential function. Hence, $g(\cdot,\cdot)$ is jointly convex with respect to its arguments [12]. We note that constraints $C_5 - C_9$ are convex with respect to $\{r_k^{\rm a}[t], r_{k,m}^{\rm f}[t]\}$ [8]. To exploit this structure, we substitute $p_k^{\rm a}[t]$ and $p_{k,m}^{\rm f}[t]$ with $r_k^{\rm a}[t]$ and $r_{k,m}^{\rm f}[t]$ in C_2 and C_3 using

$$p_k^{\rm a}[t] = C_k^{\rm a}[t] \cdot \left(d_k^{\rm a}[t]\right)^{\alpha} \cdot g\left(b_k^{\rm a}[t], r_k^{\rm a}[t]\right), \tag{9}$$
$$p_{k,m}^{\rm f}[t] = C_m^{\rm f}[t] \cdot \left(d_m^{\rm f}[t]\right)^{\alpha} \cdot g\left(b_{k,m}^{\rm f}[t], r_{k,m}^{\rm f}[t]\right),$$

Algorithm 1: Proposed Iterative Two-layer Algorithm

Input: $T_0, B, Q_{\max}, V_{\max}, P_{\max}^{\rm f}, C_{\rm DT}, E_{\min}^{\rm a\backslash f}, \{\boldsymbol{u}_m^{\rm BS}\}_{m=1}^{M}, \left\{P_{k,\max}^{\rm a}, D_k, c_k, \boldsymbol{u}_k^{\rm s}\right\}_{k=1}^{K}$

Output: $T_{\rm S}^*$, $\boldsymbol{u}^*[t]$, $\boldsymbol{R}^*[t]$, $\boldsymbol{B}^*[t]$

Let $T_{\rm next} = T_0$, $T_{\rm lb} = 0$, $x_{\rm lb} = 0$, $T_{\rm ub} = \emptyset$, $x_{\rm ub} = \emptyset$
while $T_{\rm S} \neq T_{\rm next}$ **do**
  $T_{\rm S} = T_{\rm next}$;
  Solve $\mathcal{P}_2$ and obtain the optimal solution x^*;
  if $x^* \geq 1$ **then**
    $T_{\rm ub} = T_{\rm S}$; $x_{\rm ub} = x^*$;
    $T_{\rm next} = T_{\rm lb} + \left\lceil \frac{\sum_{k=1}^{K} D_k \cdot (1 - x_{\rm lb})}{\sum_{k=1}^{K} D_k (x_{\rm ub} - x_{\rm lb})/(T_{\rm ub} - T_{\rm lb})} \right\rceil$;
  else
    $T_{\rm lb} = T_{\rm S}$; $x_{\rm lb} = x^*$;
    if $x_{\rm ub} = \emptyset$ **then**
      $T_{\rm next} = T_{\rm lb} + \left\lceil \frac{\sum_{k=1}^{K} D_k \cdot (1 - x_{\rm lb})}{\sum_{k=1}^{K} D_k \cdot x_{\rm lb}/T_{\rm lb}} \right\rceil$;
    else
      $T_{\rm next} = T_{\rm lb} + \left\lceil \frac{\sum_{k=1}^{K} D_k \cdot (1 - x_{\rm lb})}{\sum_{k=1}^{K} D_k (x_{\rm ub} - x_{\rm lb})/(T_{\rm ub} - T_{\rm lb})} \right\rceil$;
    end
  end
end

where $C_k^{\rm a}[t] \triangleq \frac{N_0^{\rm a}}{A_0 |h_k^{\rm a}[t]|^2}$ and $C_m^{\rm f}[t] \triangleq \frac{N_{0,m}^{\rm f}}{A_0 |h_m^{\rm f}[t]|^2}$.

Using (9) and introducing auxiliary variables $\mu_m[t] \geq 0$, $\mu_k[t] \geq 0$, and $y_m[t] \geq 0$, constraints C_2 and C_3 can be rewritten as

$$\mathrm{C}_{2a}: g\left(b_k^{\rm a}[t], r_k^{\rm a}[t]\right) \leq \mu_k[t], \quad \forall k, t \tag{10}$$
$$\mathrm{C}_{2b}: C_k^{\rm a}[t] \cdot \left(d_k^{\rm a}[t]\right)^{\alpha} / P_{k,\max} \leq 1/\mu_k[t], \quad \forall k, t$$
$$\mathrm{C}_{3a}: \sum\nolimits_{k=1}^{K} g\left(b_{k,m}^{\rm f}[t], r_{k,m}^{\rm f}[t]\right) \leq \mu_m[t], \quad \forall m, t$$
$$\mathrm{C}_{3b}: C_m^{\rm f}[t] \cdot \left(d_m^{\rm f}[t]\right)^{\alpha} / y_m[t] \leq 1/\mu_m[t], \quad \forall m, t$$
$$\mathrm{C}_{3c}: \sum\nolimits_{m=1}^{M} y_m[t] \leq P_{\max}^{\rm f}, \quad \forall t.$$

Note that C_{2a}, C_{3a} and C_{3c} are convex, while C_{2b} and C_{3b} are nonconvex DC constraints for $\alpha \geq 2$. Meanwhile, we can equivalently reformulate $\mathcal{P}_2$ as

$$\mathcal{P}_3: \max_{\boldsymbol{u}[t],\boldsymbol{R}[t],\boldsymbol{B}[t],\boldsymbol{Y}[t]} x, \quad m \in \mathcal{M}, k \in \mathcal{K}, t \in \mathcal{T} \tag{11}$$
$$\text{s.t. } \mathrm{C}_1, \mathrm{C}_{2a-b}, \mathrm{C}_{3a-c}, \mathrm{C}_4 - \tilde{\mathrm{C}}_{11},$$

where $\boldsymbol{R}[t] \triangleq \{r_k^{\rm a}[t], r_{k,m}^{\rm f}[t], r_{k,m}^{\rm c}[t] \mid \forall k, m\}$ is the data rate allocation for different sensors in access and fronthaul links and the processing speed at the server. $\boldsymbol{Y}[t] \triangleq \{y_m[t] \mid \forall m\}$ is the power allocation of the relay to different BSs in the fronthaul links. The nonconvex DC constraints C_{2b} and C_{3b} can be further tackled below using SCA method [14].

C. Applying SCA to the DC Constraints

Let j be the iteration index for solving the inner-layer problem $\mathcal{P}_2$. Utilizing SCA, C_{2b} and C_{3b} can be approximated in each iteration j as

$$\mathrm{C}_{2b}^{(j)}: C_k^{\rm a}[t] \cdot \left(d_k^{\rm a}[t]\right)^{\alpha} / P_{k,\max} - f(\mu_k[t]; \mu_k^{(j)}[t]) \leq 0, \tag{12}$$
$$\mathrm{C}_{3b}^{(j)}: C_m^{\rm f}[t] \cdot \left(d_m^{\rm f}[t]\right)^{\alpha} / y_m[t] - f(\mu_m[t]; \mu_m^{(j)}[t]) \leq 0,$$

where $f(\mu; \mu^{(j)})$ is a concave lower bound for the function $1/\mu$ around the point $\mu^{(j)}$, such that $f(\mu; \mu^{(j)}) \leq 1/\mu, \forall \mu^{(j)}$.

TABLE I
PARAMETER SETTINGS [8], [13]

Parameter	Value	Parameter	Value
$P^{\mathrm{f}}_{\max}$	20 dBm	Δ	0.1 s
$P^{\mathrm{a}}_{\{k=1,2\},\max}$	15 dBm	B	1 MHz
$N^{\mathrm{a}}_0/N^{\mathrm{f}}_{0,m=1,2}$	-90 dBm	C_{DT}	100 MHz
$E^{\mathrm{a\backslash f}}_{\min}$	-10 dB	A_0	-30 dB
$\kappa^{\mathrm{a\backslash f}}$	100	ξ	10^{-5}
$\sigma^{\mathrm{a}}_k/\sigma^{\mathrm{f}}_m$	1	α	2

As $1/\mu$ is a convex function, $f(\mu;\mu^{(j)})$ can be derived as its first-order Taylor expansion, given by

$$f(\mu;\mu^{(j)}) = 2/\mu^{(j)} - \mu/(\mu^{(j)})^2. \tag{13}$$

To find a suboptimal solution of problem $\mathcal{P}_2$, we iteratively solve the following problem in each iteration

$$\mathcal{P}_3^{(j)}: \max_{\boldsymbol{u}[t],\boldsymbol{R}[t],\boldsymbol{B}[t],\boldsymbol{Y}[t]} x, \quad m\in\mathcal{M}, k\in\mathcal{K}, t\in\mathcal{T} \tag{14}$$
$$\text{s.t. } \mathrm{C}_1, \mathrm{C}_{2a}, \mathrm{C}^{(j)}_{2b}, \mathrm{C}_{3a}, \mathrm{C}^{(j)}_{3b}, \mathrm{C}_{3c}, \mathrm{C}_4 - \tilde{\mathrm{C}}_{11}.$$

As $\mathcal{P}_3^{(j)}$ is convex, it can be solved using off-the-shelf solvers such as CVX [15]. The iterative procedure is initialized by input of $\mu_k^{(0)}[t]$ and $\mu_m^{(0)}[t]$ for $t\in\mathcal{T}$, which can be flexibly selected. The iteration continues until $(x^{(j)}-x^{(j-1)})/x^{(j)} \le \xi$, where ξ is a threshold for termination. By employing SCA to tackle the DC constraints in $\mathcal{P}_3$, Algorithm 1 can return a high-quality suboptimal solution of $\mathcal{P}_1$ in polynomial time.

V. Simulation Results

In this section, we evaluate the performance of the proposed scheme for DT synchronization via simulations. We consider a PS with $K=2$ sensors located at $\boldsymbol{u}_1^{\mathrm{s}} = [10, 90, 0]$ m and $\boldsymbol{u}_2^{\mathrm{s}} = [90, 90, 0]$ m. The sensors need to update $D_1 = D_2 = 0.2$ Mbits of data to the server in a DT synchronization round. The complexity of computing the sensed data at the server is $c_1 = c_2 = C$. To assist the data synchronization, an aerial relay moving at a height of 30 m forwards data received from the sensors to $M=2$ BSs located at $\boldsymbol{u}_1^{\mathrm{BS}} = [10, 10, 0]$ m and $\boldsymbol{u}_2^{\mathrm{BS}} = [90, 10, 0]$ m. For the backhaul links, we assume $\tau = 1$ and $r^{\mathrm{b}}_{k,m}[t] = r^{\mathrm{f}}_{k,m}[t-1]$, i.e., the BSs transmit data to the DT over high-speed backhaul links in the subsequent time slot after receiving it from the relay. Unless stated otherwise, we set the simulation parameters according to Table I. For comparison, we consider four baseline schemes:

- *Baseline Scheme 1:* The optimal solution of the proposed scheme is employed for processing non-infinitesimal data segments, each with size of 10 kbits, while updating DT.
- *Baseline Scheme 2:* Unlike the proposed scheme, the communication resource (bandwidth, power, buffer) is equally allocated to each sensor in both the access and fronthaul links.
- *Baseline Scheme 3:* Unlike the proposed scheme, DT updating phase commences after the completion of the data synchronization phase. Here, the time for data synchronization is minimized.
- *Baseline Scheme 4:* Unlike Baseline Scheme 3, communication resource (bandwidth, power, buffer) is equally allocated to each sensor in both access and fronthaul links.

For the baseline schemes 2-4, the non-fixed variables are optimized using Algorithm 1.

Figure 2 (a) shows the DT synchronization time T_{S} of the considered schemes versus the maximum velocity $V_{\max}$ of the relay, for $Q_{\max} = 0.2$ Mbit and $C = 2000$ cycles/bit. When $V_{\max} = 0$, we optimize the relay's hovering position. This solution is also used as the initial position of the relay for optimizing the trajectory when $V_{\max} > 0$. We observe that T_{S} monotonically decreases with $V_{\max}$. When $V_{\max}$ is increased from 0 m/s to 50 m/s, the proposed scheme reduces T_{S} by 25.6%. This is because the high mobility allows the relay more flexibility in seeking favorable channel conditions while performing BA aided relaying. For more insights, Figure 2 (b) shows the trajectories of the proposed scheme with $V_{\max} = 10$ m/s and $V_{\max} = 50$ m/s. We observe that, with a high $V_{\max}$, the relay can move close to the sensors or the BSs. Additionally, along its trajectory, the relay strategically selects the close BS to forward the data to. Further result of relay's power allocation to the BSs shows that the relay predominantly allocates its power to the closest BS. Due to limited page space, relay's power allocation result is not included. Compared with baseline schemes 2, 3 and 4, the proposed scheme reduces T_{S} by 9%, 38.9% and 43.8%, respectively, when $V_{\max} = 60$ m/s, thanks to the optimized resource for joint communication and computing. Interestingly, the performance gap between the baseline scheme 2 and the proposed scheme increases with $V_{\max}$, highlighting the importance of resource allocation optimization for mobile relays. The same observation applies to the gap between baseline schemes 3 and 4. This is because when the relay's position is fixed, the channel conditions remain relatively stable. Therefore, the difference between fixed resource allocation and optimized resource allocation is negligible. However, as $V_{\max}$ increases, the benefits of optimized resource allocation become more pronounced. Note that the baseline scheme 1, with data segment size of 10 kbit, performs very closely to the proposed scheme. This suggests that the lower bound on the DT synchronization time provided by the proposed scheme is indeed tight.

Figure 3 (a) shows the DT synchronization time T_{S} of the considered schemes versus the computing complexity C, for $Q_{\max} = 0.2$ Mbit, $V_{\max} = 30$ m/s, and $\boldsymbol{u}[0] = [50, 0, 30]$ m. For the baseline schemes 3 and 4, where DT updating phase commences after the completion of data synchronization, T_{S} grows linearly with C. In contrast, by employing stream computing, T_{S} of the baseline schemes 1 and 2 and the proposed scheme exhibit much slower growths than baseline schemes 3 and 4, as C increases. Notably, when $C = 4500$ cycles/bit, the proposed scheme and the baseline scheme 2 achieves a 39% reduction in T_{S}, compared to the baseline scheme 4. This is because stream computing enables a more efficient utilization of the limited computing resource. For further insights, Figure 3 (b) shows the trajectories of the proposed scheme for different computing complexities. When computing complexity C is small, the communication time dominates the DT synchronization time. To accelerate data synchronization, the relay moves close to the sensors for data fetching, temporarily stores it in the buffer, and forwards the data to the BSs upon approaching them. This leads to an increased average communication throughput. In contrast, when C is large, the bottleneck for DT synchronization shifts from communication to computation. In this case, for reducing the DT synchronization time, it is critical to prioritize the early data processing at the server and enhance the utilization of

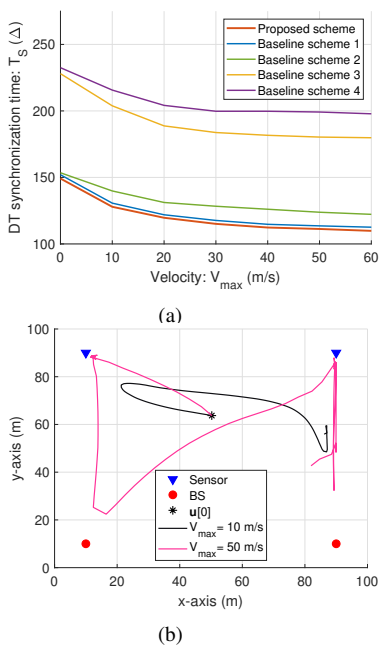

Fig. 2. (a) DT synchronization time and (b) trajectory of relay versus maximum velocity with $Q_{\max} = 0.2$ Mbits and $C = 2000$ cycles/bit.

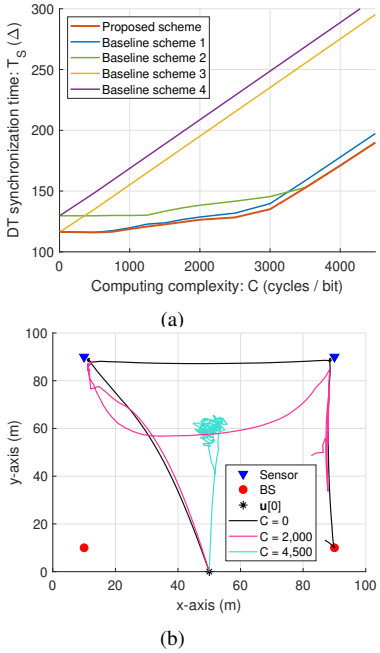

Fig. 3. (a) DT synchronization time and (b) trajectory of relay versus computing complexity with $Q_{\max} = 0.2$ Mbits and $V_{\max} = 30$ m/s.

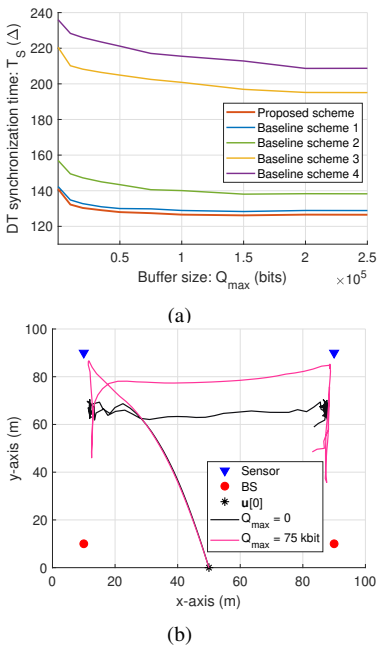

Fig. 4. (a) DT synchronization time and (b) trajectory of relay versus buffer size with $V_{\max} = 30$ m/s and $C = 2000$ cycles/bit.

computing resources. We observe from Figure 3 (b) that the relay stays within a small area, strategically located midway between the sensors and the BSs. This placement ensures relatively good and similar channel conditions for both access and fronthaul links, enabling the relay to rapidly transmit the data received from the sensors to the BSs. Consequently, this facilitates earlier data processing at the server for the considered scenario.

Figure 4 (a) shows the DT synchronization time $T_{\rm S}$ of the considered schemes versus the buffer size $Q_{\max}$ at the relay, for $V_{\max} = 30$ m/s, $C = 2000$ cycles/bit, and $\boldsymbol{u}[0] = [50, 0, 30]$ m. We observe that $T_{\rm S}$ of all considered schemes monotonically decrease with the buffer size $Q_{\max}$. This is because, buffering mitigates the need for immediate data forwarding upon reception, thereby allowing the relay to dynamically adjust its location and forwarding the received data under more favorable channel conditions. For more insights, Figure 4 (b) shows the trajectories of the proposed scheme with $Q_{\max} = 0$ and $Q_{\max} = 75$ kbits, we can see that a large buffer allows the relay to move close to the sensors and BSs for data handling.

VI. Conclusions

In this paper, we investigated DT synchronization time minimization by jointly considering the time required for data synchronization and DT updating. We exploited the deployment of a BA aerial relay to address the communication bottleneck experienced by sensors. Furthermore, we considered the novel stream computing, allowing DT updating in parallel with data synchronization, to ultimately accelerate DT synchronization. To maximize the performance, we jointly optimized the relay's trajectory and the resource allocation for communication and computing. This comprehensive approach aims to minimize DT synchronization time. To solve the formulated mixed-integer nonconvex problem, we proposed a low-complexity two-layer iterative suboptimal algorithm. Simulation results show that the DT synchronization time can be reduced by up to 43.8%. Additionally, our results reveal that, for relays with high mobility, resource allocation optimization can significantly enlarge the performance gains.

References

[1] L. U. Khan, Z. Han, W. Saad, et al., "Digital Twin of Wireless Systems: Overview, Taxonomy, Challenges, and Opportunities," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 4, pp. 2230-2254, 2022.

[2] Y. Han, D. Niyato, et al., "A Dynamic Hierarchical Framework for IoT-Assisted Digital Twin Synchronization in the Metaverse," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 268-284, 1 Jan.1, 2023.

[3] J. Zheng, et al., "Digital Twin Empowered Heterogeneous Network Selection in Vehicular Networks With Knowledge Transfer," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12154-12168, 2022.

[4] O. Hashash, C. Chaccour, W. Saad, et al., "Towards a Decentralized Metaverse: Synchronized Orchestration of Digital Twins and Sub-Metaverses," *IEEE ICC*, pp. 1905-1910, 2023.

[5] M. Vaezi, K. Noroozi, T. D. Todd, et al., "Digital Twin Placement for Minimum Application Request Delay With Data Age Targets," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11547-11557, 2023.

[6] J. Zheng, et al., "Data Synchronization in Vehicular Digital Twin Network: A Game Theoretic Approach,"*IEEE Trans. Wirel. Commun.*, 2023.

[7] J.-H. Lee, K.-H. Park, et al., "Throughput Maximization of Mixed FSO/RF UAV-Aided Mobile Relaying with a Buffer," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 683-694, Jan. 2021.

[8] Y. Wang, L. Xiang, and A. Klein, "Completion Time Minimization for UAV-Based Communications with a Finite Buffer," *IEEE ICC*, 2023.

[9] L. Neumeyer, B. Robbins, et al.,"S4: Distributed Stream Computing Platform," *IEEE ICDM*, pp. 170-177, 2010.

[10] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[11] T. Mahn, D. Becker, H. Al-Shatri and A. Klein,"A Distributed Algorithm for Multi-Stage Computation Offloading," *IEEE CloudNet*, 2018.

[12] S. P. Boyd and L. Vandenberg, *Convex Optimization*. Cambridge University Press, 2004.

[13] L. Xiang, D. W. K. Ng, et al., "Cross-Layer Optimization of Fast Video Delivery in Cache- and Buffer-Enabled Relaying Networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11366-11382, 2017.

[14] T. Lipp and S. Boyd. "Variations and Extension of the Convex–Concave Procedure," *Optim. Eng.*, vol. 17, no. 2, pp. 263-287, 2016.

[15] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," Mar. 2014. [Online]. Available: http://cvxr.com/cvx