Network Resource Allocation for Industry 4.0 with Delay and Safety Constraints

Asif Ahmed Sardar, Aravinda S. Rao, *Senior Member, IEEE,* Tansu Alpcan, *Senior Member, IEEE,* Goutam Das and Marimuthu Palaniswami, *Life Fellow, IEEE*

***Abstract*—In this paper, we model a futuristic factory floor equipped with Automated Guided Vehicles (AGVs), cameras, and a Virtual Reality (VR) surveillance system; and connected to a 5G network for communication purposes. Motion planning of AGVs and VR applications is offloaded to an edge server for computational flexibility and reduced hardware on the factory floor. Decisions on the edge server are made using the video feed provided by the cameras in a controlled manner. Our objectives are to ensure factory floor safety and provide smooth VR experience in the surveillance room. Providing proper and timely allocation of network resources is of utmost importance to maintain the end-to-end delay necessary to achieve these objectives. We provide a statistical analysis to estimate the bandwidth required by a factory to satisfy the delay requirements 99.999 percent of the time. We formulate a nonconvex integer nonlinear problem aiming to minimize the safety and delay violations. To solve it, we propose a real-time network resource allocation algorithm that has linear time complexity in terms of the number of components connected to the wireless network. Our algorithm significantly outperforms existing solvers (genetic algorithm, surrogate optimizer) and meets the objectives using less bandwidth compared to existing methods.**

***Index Terms*—Industry 4.0, Virtual Reality, Automated Guided Vehicle, Resource Block, Network Resource Allocation**

I. Introduction

We are on the cusp of a new industrialization known as Industry 4.0, driven by different applications (engineering demands) combined with digital advancements, which are the main drivers of this new revolution [1], [2]. The need for these demands arises primarily due to: shortening the development period, mass customization, highly flexible production methods, faster decision-making with a decentralized hierarchy, and promoting ecologically sustainable and resource-efficient processes. Connectivity is a crucial component for integrating IoT into Industry 4.0, where thousands of devices (such as sensors, actuators, automated guided vehicles (AGVs), mobile robots, etc.) need to connect for the smooth operation of the smart manufacturing processes.

The main objectives of a smart manufacturing system are to provide flexible production to meet the increasing demands for customization, increase productivity, and improve production quality through autonomous operations and monitoring [3]. To achieve these goals, the production lines of smart manufacturing systems must become more versatile, flexible, safe, and reliable. The integration of a new 5G radio (NR) into smart manufacturing can bring about all these important aspects of smart manufacturing by providing a reliable communication system with high data rate and low latency [3–5]. 5G and beyond networks promise support for Industry 4.0 applications. In addition, 5G provides a reliable connection to edge/cloud servers and promotes centralized control of applications on the edge / cloud. For example, motion control functions can be offloaded to the edge server to save hardware costs, increase flexibility in terms of mobility (owing to wireless connections) and improve storage capacity and computational power for scalability on demand in the factory floor [3], [6], [7]. The motivation behind our work is threefold: (i) increase scalability in terms of the number of AGVs on the factory floor, (ii) maintain safety on the factory floor, and (iii) estimate the network resource requirement along with a real-time allocation policy to satisfy the end-to-end delay requirement for the VR maintenance system with 99.999% availability. In the following paragraphs, we give an overview of our approaches to meet these objectives.

• **Improve scalability by considering the synergy among various components of the Industry 4.0 system:** In an automated Industry 4.0 system, in addition to the data injection process on the floor (video feed from the monitoring camera system on the factory floor) and the network scheduler, another critical component is the motion control of the AGVs. In our work, motion control-related functions are implemented on an edge server. An efficient communication system must be provided between the factory floor and the edge server for the smooth operation of such an industrial system. In this work, we demonstrate that the process of data injection from the factory floor to the communication system, the motion controller function on the edge server, and the communication system between the factory floor and the edge server are the three key components that dictate the proper functionality of an Industry 4.0 system.

In traditional communication systems, the sources of data injection into the wireless network are unregulated, independent, and not under the control of the network scheduler. Therefore, the network scheduler designs network resource allocation policies to distribute limited network resources to meet the different quality of service (QoS) requirements of data-generating sources. However, in an Industry 4.0 setting

This research was supported in part by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP190102828).

A. A. Sardar, A. S. Rao, T. Alpcan, and M. Palaniswami are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria - 3010, Australia (e-mail: asardar@student.unimelb.edu.au, aravinda.rao@unimelb.edu.au, tansu.alpcan@unimelb.edu.au, palani@unimelb.edu.au).

G. Das is now with the G.S. Sanyal School of Telecommunications, Indian Institute of Technology Kharagpur, West Bengal - 721302, India. (e-mail: gdas@gssst.iitkgp.ac.in).

where motion control-related functions are implemented on an edge server, sending a continuous video feed from the factory floor is less useful if the corresponding motion control decisions are not received by the AGVs in time. Therefore, it is very important to control the injection of data from the factory floor to the wireless network to maximize the efficiency of the usage of network resources.

In the literature, a few works have analyzed network-controlled systems with communication constraints [8], [9]. In these works, the controller does not regulate the data injection process. However, in an Industry 4.0 system where the controller functions are placed on an edge server, the control decisions based on the most recent camera video feed must reach the appliances on the factory floor before the next video feed is received. Therefore, it is obvious that the data injection process must be controlled along with resource allocation. Another major issue with existing approaches on network-controlled systems is that the controller function is assumed to be a continuous function. However, in a practical setup like the one considered in this work, the motion controller is a complex algorithm [10] that cannot be expressed as a continuous function of the input variables.

Therefore, we focus on controlling the injection of data into the wireless network on the factory floor according to the conditions of the channel inside the factory and the criticality of motion of the AGV. This allows us to allocate network resources to a greater number of AGVs and improve the scalability of the industrial process. In other words, for an industrial process with a fixed number of AGVs, we can achieve the QoS requirements with fewer network resources. In the results section, we show that controlling the data injection process on the factory floor in accordance with motion control and traffic scheduling brings about nearly 25% to 30% less usage of network resources while still achieving 99.999% availability in terms of end-to-end delay.

- **Network resource allocation policy with multiple objectives to reduce end-to-end delay violations and maximize safety:** A factory floor might be subject to emergency hazardous situations demanding human intervention. This requires a remote monitoring system for the smooth operation of a smart manufacturing process. Virtual Reality (VR) and Augmented Reality (AR) are part of a growing Industry 4.0 ecosystem. These technologies have been considered to be the main 5G application use cases due to their very high bandwidth and strict latency requirements [11]. VR/AR developers agree that for Motion-to-Photon (MTP) latency to become imperceptible, the application round-trip latency should be less than 15-20 ms [11], [12]. Therefore, the end-to-end delay must be regulated by a proper resource allocation strategy. On the other hand, proper delivery of motion controls is also important for correctly guiding AGVs on the factory floor and maintaining a safe environment.

Keeping the aforementioned issues in mind, we design a network resource allocation policy that satisfies both the end-to-end delay requirement for the VR maintenance system and the AGV safety requirement that dictates the productivity of the Industry 4.0 system. If we remove the safety criterion from the network resource allocation problem, the AGVs may not be able to receive enough resources to receive the motion controls, which in turn increases the risk of collisions on the floor. On the other hand, if we remove the end-to-end delay requirement, the network resource allocation process is concerned only with the safety of the factory floor, and as a result, the round-trip delay requirement for VR maintenance is not considered at all. Therefore, we propose a joint optimization problem in our work and develop a network resource allocation policy based on the optimization framework.

A. *Related works*

In the literature, a lot of work has been done on different aspects of Industry 4.0 systems, such as navigation, control, scheduling and path planning for AGVs, VR/AR frameworks for the smart factory, wireless communication systems inside the factory, etc. We briefly discuss some of the relevant work.

Extensive research has been done on the applications of AGV technology in smart manufacturing [13]. They can be roughly divided into the following groups: (a) algorithms for localization, scheduling, docking, and path planning algorithms; (b) navigation control and guidance algorithms; (c) wireless communication between AGVs; (d) power consumption and management; and (e) AGV design and applications.

In recent years, VR/AR technology has been envisioned as a critical tool for remote maintenance of future smart factories [14–16]. VR helps visualize a factory floor, which can be used for maintenance training, to predict design defects, etc. [16]. Several works have been carried out on different aspects of VR/AR in the context of Industry 4.0, such as (a) developing methods for data collection from the factory floor and visualization [17–19], and (b) designing edge computing-based architecture for VR/AR applications [20–22].

As mentioned previously, there has been a growing trend of moving VR/AR applications to the edge server, aided by the powerful wireless framework supported by 5G [20–22]. Similarly, the controller functions necessary for a smart manufacturing system are also expected to be moved to edge servers [3], [23], [24]. Although edge/fog computing brings about many advantages on the computing side, there is a limitation in terms of high delay that can disrupt delay-sensitive VR/AR devices and control functions. The authors of [23] investigate the consequences of moving toward cloud-based controllers using existing frameworks. To reduce the problem of delays, a smart manufacturing process with edge-based control systems must have enough network resources and a good policy for allocating those resources. Very few works in the literature [25–27] have used heuristic and greedy approaches to address the issue of bandwidth allocation in an Industry 4.0 setting. To the best of our knowledge, no work in the literature explores the resource allocation problem for smart manufacturing systems with an edge-based controller.

B. *Contributions*

The summary of our contributions in this work is as follows:

- *Modelling a smart factory:* We model a fully automated smart factory system containing stationary robots, AGVs, a monitoring camera system, and a VR surveillance

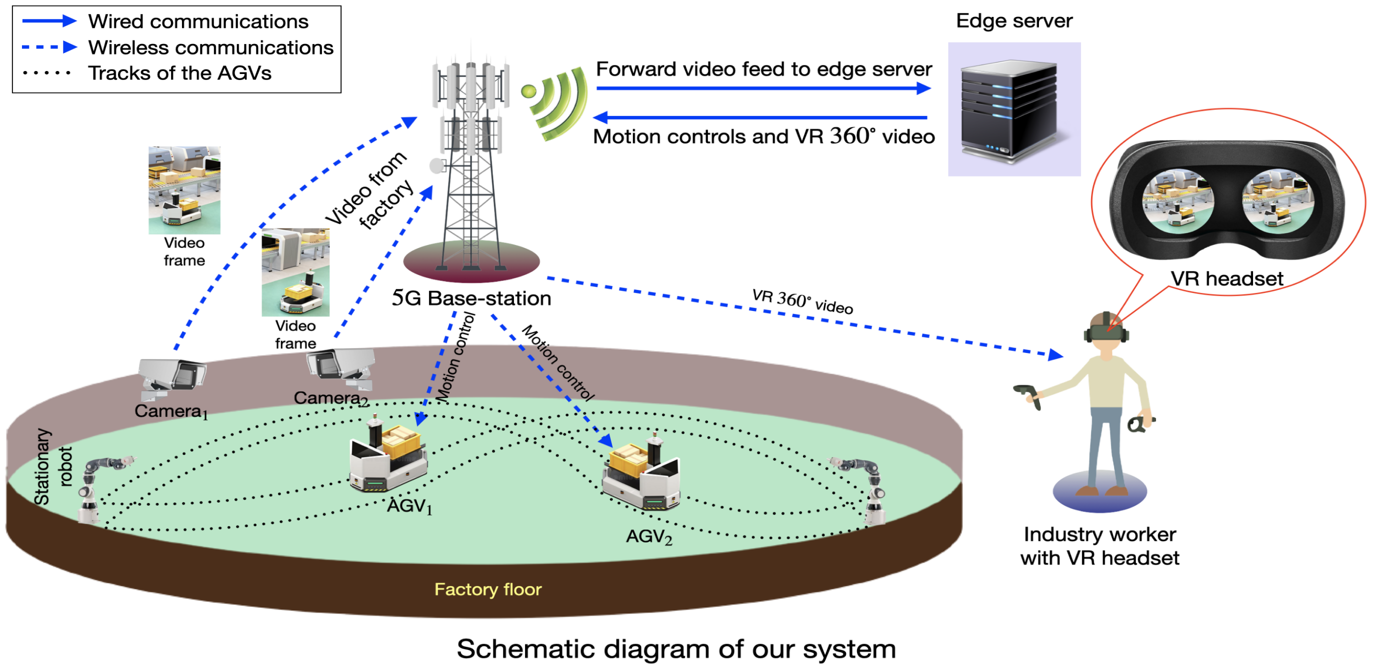


Fig. 1: Sequence of operation at a fully automated factory floor: (1) 5G base station allocates network resources to the camera system $(\text{Camera}_1, \cdots, \text{Camera}_C)$ at the factory floor, (2) The edge server receives the video feed from base station via a wired connection and process it, (3) Edge server forwards the VR 360° video and motion controls to the base station, (4) Base station allocates network resources to the surveillance system and AGVs $(\text{AGV}_1, \text{AGV}_2, \cdots, \text{AGV}_M)$, (5) VR 360° video and motion control decisions are received by the surveillance system and AGVs respectively.

system. Industrial controller functions are offloaded to an edge server. We consider various streams of communication among the components of the smart factory that affect the functionality of the industrial system.

- *Controlled data injection to the wireless network:* We control the flow of camera video feed from the factory floor to the edge server, which is a departure from traditional wireless communication systems where data-generating sources are not regulated. In the Industry 4.0 system, the camera video feed and the corresponding control decisions go hand in hand, requiring us to control the data injection process for a superior use of network resources. Our controlled data generation approach reduces network load, which is a key innovation in factory floor settings.
- *Delay, safety, and efficiency:* The delay in end-to-end communication between the factory floor and the edge server affects the safety of the factory floor, as well as the VR experience. We connect the delay and safety requirements with the allocation of network resources to the various industrial components connected to the wireless network. Then, we develop a non-convex integer nonlinear programming with a network resource constraint, aiming to reduce the events of delay and safety violation. This is a novel problem formulation compared to existing approaches in the literature.
- *Estimation of necessary network resources:* In the planning phase of a smart factory floor, it is important to have a good estimate of the necessary network resources for the design of the network slice. We provide a statistical method to estimate the lower bound of the amount of bandwidth necessary to achieve 99.999% availability in terms of latency, safety, and efficiency for a smart factory system. To the best of our knowledge, our work is the first to provide such an estimate.
- *A linear algorithm for real-time network resource allocation:* We propose an algorithm for real-time network resource allocation that has linear complexity in terms of the number of AGVs and the number of monitoring cameras. Our algorithm outperforms global optimizers (such as genetic and surrogate optimization algorithms) and is suitable for real-time resource allocation.

The remainder of the paper is organized as follows. In Section II, we describe the model of the smart factory floor system. In Section III, we define various streams of communication related to factory functionality and discuss the resource allocation architecture of the 5G NR. In Section IV, we define the system parameters and functions necessary to analyze our problem. We describe the design and operational phases of the Industry 4.0 system in Section V. In Section VI, we develop a real-time network resource allocation policy and a corresponding linear-time algorithm to solve it. Extensive simulation results are given in Section VII. Finally, we conclude and give possible future directions in Section VII.

II. System Model

We consider a factory floor where mobile robots / automated guided vehicles (AGV) are used in an automation process as shown in Fig. 1. Each AGV is assumed to move forward and backward between two stationary robots, following a fixed path. A 5G base station is located near the factory complex [28]. Cameras, AGVs, and surveillance gadgets are interconnected over the 5G NR network.

It is of utmost importance to properly control the motion of the AGVs to avoid collisions between the AGVs and maintain proper functionality on the factory floor. Motion planning-related functions are deployed on an edge server, which is connected to the 5G base station via a wired connection. Safety-related control functions are retained on board the AGVs, allowing them to prevent collisions in the absence of motion controls from the edge server (due to network connectivity issues). The factory floor is monitored by a video camera system mounted on the walls of the factory floor [3]. Once motion control decisions based on the previous video feed are sent to the edge server AGVs, the 5G base station allocates network resources to the camera system for transmitting the most recent video feed. Older video feed that was not previously transmitted is discarded, as these data do not contribute to deciding the future motion controls. So, in our system, the process of data injection to the 5G network is not independent like other wireless communication systems but is carried out in a controlled fashion. The industry is also equipped with a virtual reality surveillance system. The video feed from the factory floor is processed at the edge server for machine vision processing and estimation of an object's position, velocity, and orientation in the physical environment, and for the creation of VR 360° video of the factory floor. Motion control decisions and VR 360° video are sent back to the 5G base station from the edge server. Motion controls are transmitted to AGVs and VR 360° video is transmitted to the surveillance room of the factory floor.

The performance of operations at a smart factory floor depends heavily on the *availability* of the communication system inside the factory. The term *availability* means the "communication service availability" inside the factory. A system is considered to be available only if it satisfies all required QoS parameters, such as latency, data rate, efficiency, etc. [4]. For smooth operation on the factory floor, the desired target value of availability should be greater than 99.999% [3]. In this work, we provide a network resource allocation policy and a method to estimate a lower bound on the number of network resource blocks necessary to maintain 99.999% availability in terms of latency (i.e., delay violation probability ≤ 0.001) and safety at the factory floor.

III. Network Resource Allocation problem

A. Communication Streams

The timeline of the factory floor is slotted into intervals of T_S. The video feed from the camera can be sent at the beginning of a slot. In general, there are five communication streams in our model, three wireless communication streams, and the rest are wired communication through optical fibers. Only the wireless communication streams are relevant to our analysis (marked with *).

(1) ***Camera to 5G base station (wireless):** Video feed is transmitted from the camera to the 5G base station at the beginning of a slot (as requested by the edge server).
(2) **5G base station to the edge (wired):** The video feed from the camera system is sent to the edge server through a wired connection (optical fiber).
(3) **Edge to 5G base station (wired):** The motion controls of the AGVs are decided and the VR 360° video is created at the edge server and sent to the 5G base station.
(4) ***5G base station to the camera (wireless):** 5G base station requests the camera system to provide the next video feed after receiving motion control data and VR 360° video from the edge server.
(5) ***5G base station to AGVs (wireless):** 5G base station forwards the motion control parameters to AGVs.
(6) ***5G base station to surveillance room (wireless):** The VR 360° video feed is transmitted to the surveillance room.

B. Resource Block Allocation

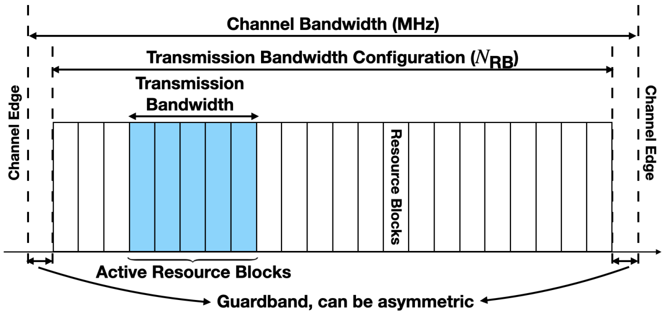


Fig. 2: 5G New Radio Resource Block Architecture [29]

5G NR is a new radio access technology developed by 3GPP for the 5G (fifth generation) mobile network [30] (shown in Fig. 2). In 5G NR, network resource blocks are allocated to users according to their QoS requirements. According to 3GPP TS 38.211 version 16.3.0 release 16, a Resource Block (RB) is a block of 12 consecutive subcarriers in the frequency domain. 5G NR provides different subcarrier spacings that can be obtained by multiplying the base long-term evolution (LTE) subcarrier spacing (180 kHz) by 2^{μ}, where $\mu = 0, 1, 2, 3, 4$. The Transmission Time Interval (TTI) duration, which is 1 ms for $\mu = 0$, is scaled down by 2^{μ}. The maximum number of available RBs depends on the subcarrier spacing and the maximum available channel bandwidth [31]. The time interval T_S (defined in Section III-A) is assumed to be an integer multiple of the TTI.

In this work, we focus on massive Machine Type Communications (mMTC) with stringent delay requirements. For the Ultra Reliability and Low Latency Communications (URLLC) pertaining to different sensors on the factory floor, it is assumed that a separate network slice is available. In our system, we are concerned about the allocation of RBs to cameras, AGVs, and surveillance system by the 5G base station. Upon receiving the motion control parameters and VR 360° video from the edge server, the 5G base station allocates the available RBs among the AGVs and the VR surveillance system to send the control decisions and VR video feed, respectively. Once the surveillance system and all AGVs acknowledge the retrieval of the messages, the 5G base station then assigns the RBs among the cameras. The RB allocations must be done judiciously to keep transmission delays within the tolerable level. We assume flat fading inside the factory floor [26]. Therefore, we are only concerned about the number of RBs allocated to different components of the automation process.

IV. Problem Formulation

The factory floor under consideration employs M AGVs for the automation process. The start and end coordinates of the i^{th} AGV are denoted by $s_i = (s_i^x, s_i^y)$ and $e_i = (e_i^x, e_i^y)$. The AGVs move between their respective source and destination on predefined paths. The maximum achievable velocity for each AGV is v_{max}. The AGVs must maintain a minimum distance to prevent collision events, which is referred to as the collision avoidance radius and is denoted by r_a. The factory floor is monitored by the C number of cameras mounted on the walls. We send the video feed from the camera system to the edge server in a controlled manner. The index of the time slot when the edge server requests the cameras to provide the n^{th} video feed to the 5G base station is denoted by k_n. The video feed consists of a fixed number of video frames just before the k_n^{th} time slot. We analyze the system for a finite number of video transmission requests (denoted by N). N depends on the duration of the industry operation. The key system parameters are summarized in Table I. Now we define certain key parameters related to motion and resource allocation. Then, we discuss the mutual dependence between the resource allocation problem and the motion controller on the edge server. Finally, we discuss the constraints and objectives of our problem.

A. Motion and Resource Allocation Related Parameters

In this subsection, we shall discuss the motion related parameters for the AGVs and the resource allocation parameters for the network resource allocation problem. The controller on the edge server extracts the motion parameters from the video feed provided by the camera system. Position, velocity, and acceleration of the i^{th} AGV at the beginning of the n^{th} video transmission (as calculated at the edge server) are denoted by $\mathbf{p}_i(k_n)$, $\mathbf{v}_i(k_n)$, and $\mathbf{a}_i(k_n)$ respectively. The distance traveled by the i^{th} AGV starting at the k_n^{th} time slot until the k_{n+1}^{th} time slot is denoted by $d_i(k_n)$. If an AGV detects any obstacle on its current trajectory, it will stop before the obstacle comes within its collision avoidance radius.

The number of resources available on the factory floor is denoted by N_{RB}. The spacing of the subcarriers of the RBs is Δf. Let $x_k(n)$ denote the number of RBs allocated to the k^{th} camera prior to n^{th} video transmission. The transmission delays over wireless channels depend on both the number of allocated RBs and the channel conditions. The gain of the channel from the k^{th} camera to the 5G base station at the beginning of the n^{th} video transmission is denoted by $h_k(n)$. Once the 5G base station receives messages (based on the n^{th} video transmission) from the edge server, the 5G base station allocates the RBs among the cameras and the VR surveillance system. The number of RBs assigned to the i^{th} AGV and the surveillance system is indicated by $y_i(n)$ and $y_{\text{VR}}(n)$ respectively.

TABLE I: List of system parameters

Parameter	Definition
T_S	Slot duration
C	Number of cameras at the factory floor
M	Number of AGVs at the factory floor
N	Number of video transmissions
r_a	Collision avoidance radius
v_{max}	Maximum velocity of each AGV
(s_i^x, s_i^y)	Start coordinates of the i^{th} AGV
(e_i^x, e_i^y)	End coordinates of the i^{th} AGV
N_{RB}	Number of available resource blocks
Δf	Sub-carrier spacing
θ	Threshold of end-to-end delay
P_{cam}	Transmission power of each camera
P_{B}	Transmission power of 5G base station
δ_{PROC}	Processing delay at the edge server

TABLE II: List of parameters relevant to n^{th} video transmission from the factory floor

Parameter	Definition
k_n	Slot index of the n^{th} video transmission
$\mathbf{p}_i(k_n)$	Position of the i^{th} AGV
$\mathbf{v}_i(k_n)$	Velocity of the i^{th} AGV
$\mathbf{a}_i(k_n)$	Acceleration of the i^{th} AGV
$d_i(k_n)$	Distance traversed by the i^{th} AGV
$x_k(n)$	Number of RBs allocated to the k^{th} camera
$y_i(n)$	Number of RBs allocated to the i^{th} AGV
$y_{\text{VR}}(n)$	Number of RBs allocated to surveillance system
$h_k(n)$	Channel gain from k^{th} camera to 5G base station
$g_i(n)$	Channel gain from 5G base station to the i^{th} AGV
$g_{\text{VR}}(n)$	Channel gain from 5G base station to surveillance system

The channel gains from the 5G base station to the i^{th} AGV and the surveillance system are denoted by $g_i(n)$ and $g_{\text{VR}}(n)$ respectively. The delay in transmission of the n^{th} video feed from the k^{th} camera is represented by $\delta_{\text{CAM}}^k\left(x_k(n); h_k(n)\right)$. The processing delay on the edge server is denoted by δ_{PROC}. Finally, $\delta_{\text{AGV}}^i\left(y_i(n); g_i(n)\right)$ and $\delta_{\text{VR}}\left(y_{\text{VR}}(n); g_{\text{VR}}(n)\right)$ denote the transmission delay to send the motion control parameters to i^{th} AGV and VR 360° video to the surveillance room from 5G base station. All the parameters pertaining to the n^{th} video transmission are summarized in Table II.

B. Delay Constraint

We formulate the delay constraint that ensures the timely delivery of motion control parameters and VR 360° video from the edge server. Latency is a critical quality parameter for VR applications. An excessive delay can disrupt the immersive VR experience [11]. According to the existing literature [12], [32], the end-to-end latency must not exceed 15 ms. On the other hand, the maximum velocity of the AGVs is in the range of 1-4 meter/second. As a result, the movements of AGVs in a few milliseconds do not increase the chance of collisions. So, the latency requirement is more critical for the VR 360° video feed transmission compared to the delivery of motion controls to AGVs. For a concise mathematical description of the delay constraint, we need to define some peripheral functions.

- The uplink delay corresponding to the n^{th} video transmission $= \delta_{\text{UL}}(n) = \max_{k=1}^{C} \delta_{\text{CAM}}^k\left(x_k(n); h_k(n)\right)$.

TABLE III: List of delay functions relevant to n^{th} video transmission from the factory floor

Parameter	Definition
$\delta^{k}_{\text{CAM}}\left(x_k(n); h_k(n)\right)$	Tx. delay from the k^{th} camera to 5G base station
$\delta^{i}_{\text{AGV}}\left(y_i(n); g_i(n)\right)$	Tx. delay from 5G base station to the i^{th} AGV
$\delta_{\text{VR}}\left(y_{\text{VR}}(n); g_{\text{VR}}(n)\right)$	Tx. delay from 5G base station to surveillance
$\delta_{\text{UL}}(n)$	Uplink delay from factory to 5G base station
$\delta_{\text{DL}}(n)$	Downlink delay from 5G base station to factory

- The downlink delay corresponding to the n^{th} video transmission $= \delta_{\text{DL}}(n) = \max\{\delta_{\text{VR}}\left(y_{\text{VR}}(n); g_{\text{VR}}(n)\right)$, $\max_{i=1}^{M} \delta^{i}_{\text{AGV}}\left(y_i(n); g_i(n)\right)\}$.

The various delay functions are summarized in Table III. The end-to-end delay is the sum of the uplink transmission delay, processing delay (denoted δ_{PROC}), and downlink transmission delay. We denote end-to-end latency by $\Delta(n) = \delta_{\text{UL}}(n) + \delta_{\text{PROC}} + \delta_{\text{DL}}(n)$. The end-to-end delay must be within a threshold, denoted by θ. Now we can define a recurrence relation for the sequence $\{k_1, k_2, \cdots, k_N\}$ (defined as the indices of the time slots when the base station requests the camera system to provide a video feed in Section IV) as follows:

$$k_1 = 1, k_{n+1} = k_n + \left\lceil \frac{\Delta(n)}{T_S} \right\rceil. \tag{1}$$

C. Safety and Efficiency Objective

In this subsection, our aim is to quantify one of the most important objectives of a fully automated factory, which is maintaining safety throughout the production process. To ensure safety, we must ensure that the AGVs do not collide with each other or with the machinery inside the factory. Therefore, our objective should be to minimize events in which something enters within the collision avoidance radius (r_a) of any AGV. We can avoid the occurrence of any such event by simply keeping the AGVs still. However, this will negatively impact the factory operations. Therefore, we need to define a slightly different metric that indirectly captures the consequence of collision events.

The timely delivery of motion controls from the edge server to the AGVs is necessary for the collision-free motion of the AGVs. In the absence of proper motion control, an AGV will move uncontrollably and may collide with another AGV or other stationary equipment. At that point, the crashed AGV will not have any further involvement in the production process until an outside intervention is made. Assuming the availability of perfect motion controls, the AGVs will be able to move within the factory without any collisions. So, we define a new metric, namely *efficiency*, defined as the total distance traversed by the AGVs in a time window divided by the maximum possible distance that the AGVs can cover during the same time period. If a collision event occurs, the involved AGVs cease to operate until external intervention and therefore do not contribute to the total distance covered by the AGVs, reducing the value of *efficiency*. So, if we are able to avoid collision events, it ensures that all AGVs can move around the factory and keep the production process running smoothly. As a result, the distance covered by the AGVs is maximized, and the *efficiency* factor increases. So, we can minimize the number of collision events by maximizing *efficiency*. In this work, we focus on the real-time allocation of network resources. We denote the efficiency corresponding to n^{th} video transmission from the factory floor by $\eta(n)$. We use the moving average of window size W to define the efficiency as

$$\eta(n) = \frac{1}{W}\left[\sum_{i=1}^{W-1} \eta(n-i) + \frac{\sum_{i=1}^{M} d_i(k_n)}{v_{\max} M(k_{n+1} - k_n) T_S}\right], \tag{2}$$

where the first term (summation term) corresponds to the accumulated efficiency the previous $W-1$ videos transmissions before n^{th} video transmission; the numerator of the second term denotes the total distance covered by M AGVs before the start of $(n+1)^{th}$ video transmission; and the denominator conveys the total distance covered by these AGVs in the same time window at the highest possible speed (denoted by $v_{\max}$).

V. Planning and Operational Phases

During the planning phase of a 5G connected automated industrial system, it is necessary to design the network slice for the different use cases, such as Enhanced Mobile Broadband (eMBB), massive Machine Type Communication (mMTC), and Ultra Reliability and Low Latency Communication (URLLC). The requirement for RBs depends on the number of AGVs operating on the factory floor, as well as the number of cameras to be used to monitor the floor. Once the planning phase is completed and network slices are created, a real-time resource allocation scheme must be designed to allocate the RBs to the various components of the Industry 4.0 system. In the following sections, we give a method to create a network slice for the specific problem described in Section IV and propose a real-time resource allocation policy for the operational phase of the system.

A. Planning phase: Estimation of the number of resource blocks

In this subsection, we describe a statistical method to estimate the number of resource blocks necessary to maintain the end-to-end delay 99.999% of time, for the URLLC and eMBB use cases described in Section IV. As mentioned in Section IV-C, the delay constraints cannot be guaranteed to be satisfied at all times regardless of the value of N_{RB}. Our first step would be to find the delay violation probability corresponding to N_{RB} and a real-time network resource allocation scheme.

For details of channel gain parameters for different wireless communication streams, refer to Section IV-A. All of the channels' gains are thought to be controlled by random variables that are not related to each other. The uplink delays from the cameras to the 5G base station and the downlink delays from the base station to the AGVs and surveillance room are monotonic functions of their respective channel gains. Various uplink and downlink delays depend on the channel gains and the distance from the base station. To estimate N_{RB} that can maintain 99.999% availability, we consider the situation when all AGVs are the furthest from the base station. So, we can find the probability distributions governing these uplink and downlink delays if the probability distributions of the channel gain parameters are known.

The probability distribution followed by the uplink transmission delay of the n^{th} video feed from the k^{th} camera to the 5G base station is indicated by $p^{k}_{\text{UL}}(\cdot)$. The corresponding cumulative distribution function (cdf) is $P^{k}_{\text{UL}}(\cdot)$. The probability distributions that govern the downlink delays from 5G base station to i^{th} AGV and the surveillance system are represented by $p^{i}_{\text{DL}}(\cdot)$ and $p^{\text{VR}}_{\text{DL}}(\cdot)$ respectively. The corresponding CDFs are denoted by $P^{i}_{\text{DL}}(\cdot)$ and $P^{\text{VR}}_{\text{DL}}(\cdot)$. Now we can calculate the probability distributions of the overall uplink and downlink delays (defined in Section IV-B). The CDFs governing the uplink and downlink delays are given by

$$\begin{aligned} P_{\text{UL}}(\cdot) &= \prod_{k=1}^{C} P^{k}_{\text{UL}}(\cdot), \\ P_{\text{DL}}(\cdot) &= P^{\text{VR}}_{\text{DL}}(\cdot) \times \prod_{i=1}^{M} P^{i}_{\text{DL}}(\cdot). \end{aligned} \tag{3}$$

The corresponding probability distribution functions are $p_{\text{UL}}(\cdot)$ and $p_{\text{DL}}(\cdot)$, respectively.

In this work, it is assumed that the processing delay on the edge server is of constant value for a given number of AGVs. Let θ be the delay threshold requirement on the factory floor. The theoretical delay violation probability (denoted by P_{Δ}) can be written as

$$\begin{aligned} P_{\Delta} &= P\left[\Delta(n) \leq \theta\right] \\ &= P\left[\delta_{\text{UL}}(n) + \delta_{\text{DL}}(n) \leq \theta - \delta_{\text{PROC}}\right] \\ &= \int_{y=0}^{\theta-\delta_{\text{PROC}}} p_{\text{UL}}(y) \left(\int_{x=0}^{\theta-\delta_{\text{PROC}}-y} p_{\text{DL}}(x)dx \right) dy \\ &= \int_{y=0}^{\theta-\delta_{\text{PROC}}} p_{\text{UL}}(y) P_{\text{DL}}(\theta - \delta_{\text{PROC}} - y)dy. \end{aligned} \tag{4}$$

As discussed in Section II, the delay violation probability (P_{Δ}) should be ≤ 0.001. With access to more RBs, we can reduce transmission delays by allocating more network resources to the camera system, AGVs, and surveillance system. Given the system parameters of a factory floor, we can estimate a lower bound on the number of RBs (N_{RB}) using this probabilistic approach. For estimation purpose, we consider the channel gain parameters (from the 5G base station to the camera system and the AGVs) to follow an exponential distribution with mean value λ. Uplink and downlink delays are determined by the available RBs, the channel conditions, and distances from the 5G base station. In the planning phase of the Industry 4.0 system, the average distance between an AGV and the 5G base station is used to calculate the downlink delays.

To find out the lower bound of N_{RB}, we fix a range $[1, \hat{N}_{\text{RB}}]$. Round trip delay is a monotonically decreasing function of N_{RB}. We use binary search in the range $[1, \hat{N}_{\text{RB}}]$ to find the value of N_{RB} that allows the probability of round trip delay violation to remain below 0.001. Once we fix the value of N_{RB}, we can assign the RBs to the camera system, the AGVs, and the maintenance room using a similar method described in Subroutine 1 in Section VI. Then, we can calculate the round-trip delay violation probability using Equation (4). If the chosen range is not enough to get a value of N_{RB} that keeps the probability of violation of the round trip delay below 0.001, we double the value of $\hat{N}_{\text{RB}}$ and search the new range. Following this process, we can obtain a lower bound of N_{RB}. This gives us a good estimate of the network resource requirements for an industrial setup. Please refer to Section VII-A for the simulation results.

B. Operational phase: Real-time network resource allocation framework

At the beginning of each video transmission from the cameras on the factory floor, the channel conditions inside the factory are estimated by the 5G base station, and the RBs are allocated to the cameras, AGVs, and surveillance system. The resource allocation scheme must satisfy two primary objectives of our system. Our first objective is to keep the end-to-end delay associated with n^{th} video transmission ($\Delta(n)$) within the required threshold θ. This will ensure a smooth VR experience in the surveillance room and convey motion controls to the AGVs in time to prevent any collisions. The second objective of our problem is to make the factory floor as efficient as possible. The combined optimization problem for the first N video transmissions from the factory floor can be written as

$$\begin{aligned} &\min\ \alpha_1 \sum_{n=1}^{N} \mathbf{1}_{\Delta(n)>\theta} + \alpha_2 \left(1 - \eta(N)\right) \\ &\Leftrightarrow\ \alpha_1 \sum_{n=1}^{N} \mathbf{1}_{\Delta(n)>\theta} + \alpha_2 \left(1 - \frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{i=1}^{M} d_i(k_n)}{v_{\max} M (k_{n+1} - k_n) T_S}\right) \\ &\text{s.t.} \sum_{k=1}^{C} x_k(n) \leq N_{\text{RB}} \quad \forall\ n \in [0, N] \\ &\qquad y_{\text{VR}}(n) + \sum_{i=1}^{M} y_i(n) \leq N_{\text{RB}} \quad \forall\ n \in [0, N], \end{aligned} \tag{5}$$

where $\alpha_1, \alpha_2 > 0$ are parameters of the scalarization and $\mathbf{1}_{\Delta(n)>\theta}$ is an indicator function.

Clearly, this optimization problem depends on the sequence of decisions made by the controller on the edge server. As we do not participate in the design of the motion controller, the problem cannot be solved using existing optimization methods. Instead of solving the problem over a horizon (N number of transmissions), we tackle a myopic version of the problem, which is tractable, and its solution provides us with a real-time resource allocation policy. The optimization problem corresponding to n^{th} video transmission can be written as

$$\begin{aligned} &\min\ \alpha_1 \mathbf{1}_{\Delta(n)>\theta} + \alpha_2 \left(1 - \eta(n)\right) \\ &\text{s.t.} \sum_{k=1}^{C} x_k(n) \leq N_{\text{RB}} \\ &\qquad y_{\text{VR}}(n) + \sum_{i=1}^{M} y_i(n) \leq N_{\text{RB}}. \end{aligned} \tag{6}$$

The first term of the optimization problem is an indicator function. To minimize it, we need to satisfy the inequality constraint pertaining to the indicator function. In the second term of the optimization problem, the numerator of the efficiency term depends on the distance covered by the AGVs. A particular AGV continues to move with its current velocity until it receives new motion controls or faces an obstruction (whichever occurs first). The denominator, on the other hand, decreases as the end-to-end delay decreases. By decreasing $\Delta(n)$, we effectively reduce the second term of our optimization problem. Therefore, we need to minimize the end-to-end delay to minimize (6). The end-to-end delay has been defined in Section IV-B as the sum of the uplink, processing, and downlink delays. The uplink and downlink delay are defined as the maximum of individual uplink and downlink delays, respectively. So, our objective function is a non-convex integer nonlinear programming.

VI. Network resource allocation policy in real-time

In this section, we develop a real-time network resource allocation policy that solves the myopic version of the optimization problem described in Section V-B. We need to approximate the delay functions described in Section IV-A using the Shannon-Hartley theorem [33] to obtain an explicit expression for (6). We can write down the approximate delay functions [33], [34] as:

$$\delta^k_{\text{CAM}}\left(x_k(n);h_k(n)\right) \approx s_{\text{camera}} \Big/ \left[12\Delta f x_k(n) \log_2\left(1+\frac{P_{\text{cam}} h_k(n) (r^k_{\text{cam}})^{-\alpha}}{12\Delta f x_k(n) N_0}\right)\right], \tag{7a}$$

$$\delta^i_{\text{AGV}}\left(y_i(n);g_i(n)\right) \approx s_{\text{control}} \Big/ \left[12\Delta f y_i(n) \log_2\left(1+\frac{P_{\text{B}} g_i(n) (r^i_{\text{agv}}(n))^{-\alpha}}{12\Delta f y_i(n) N_0}\right)\right], \tag{7b}$$

$$\delta_{\text{VR}}\left(y_{\text{VR}}(n);g_{\text{VR}}(n)\right) \approx s_{\text{VR}} \Big/ \left[12\Delta f y_{\text{VR}}(n) \log_2\left(1+\frac{P_{\text{B}} g_{\text{VR}}(n) r^{-\alpha}_{\text{VR}}}{12\Delta f y_{\text{VR}}(n) N_0}\right)\right]. \tag{7c}$$

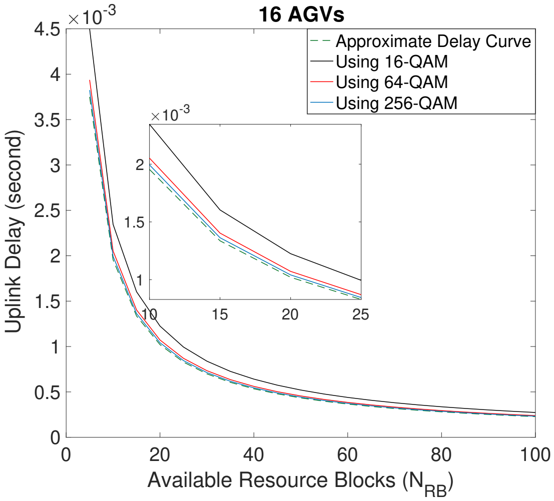


Fig. 3: Uplink delay vs the number of available RBs for different coding schemes

The subcarrier spacing of each RB is denoted by Δf. The spectral density of the noise inside the factory floor is N_0. The transmission power levels of the cameras and the 5G base station are indicated by P_{cam} and P_{B}, respectively. The path-loss factor inside the factory is α. The distance from the surveillance room and the k^{th} camera to the base station is represented by r_{VR} and r^k_{cam}. The distance of the i^{th} AGV from the base station when that AGV receives motion controls related to the n^{th} video transmission is denoted by $r^i_{\text{agv}}(n)$. The size of the messages generated on the cameras is s_{camera}. The size of the messages retrieved by the AGVs and the surveillance system is s_{control} and s_{VR} respectively. In Section II, we have mentioned that a 5G base station is placed near the factory premises to consistently provide an efficient wireless communication medium. According to [28], industrial automation will require local area networks, deployed in factory premises to consistently meet URLLC performance targets. In Fig. 3, we compare the uplink delay (camera system to the 5G base station) corresponding to different modulation schemes, with the approximate delay curve (obtained from the Shannon-Hartley theorem [33], [34]). For simulation purposes, the number of AGVs and cameras is taken to be 16 and 4 respectively. We use the Channel Quality Indicator parameter provided by SimuLTE [35], to use the desired modulation schemes for the industrial system implemented in OMNet++. From Fig. 3, we can see that with an improved modulation scheme, we gradually approach the approximate delay curve (the minimum delay that can be achieved by the Shannon-Hartley theorem). 256-QAM very nearly achieves the optimal delay and is within 1% of the limit. So, under the assumption that the industrial system uses a private 5G network free from external interference, our approximations work quite well. In the next two subsections, we give the key insight behind our resource allocation policy and the corresponding algorithm.

A. Network resource allocation policy

As discussed in Section V-B, we need to minimize the end-to-end latency to minimize both terms of the real-time optimization problem. End-to-end delay consists of uplink, processing, and downlink delays. In Section IV-B, uplink and downlink delays are defined as $\delta_{\text{UL}}(n) = \max_{k=1}^{C} \delta^k_{\text{CAM}}\left(x_k(n);h_k(n)\right)$ and $\delta_{\text{DL}}(n) = \max\left\{\delta_{\text{VR}}\left(y_{\text{VR}}(n);g_{\text{VR}}(n)\right), \max_{i=1}^{M} \delta^i_{\text{AGV}}\left(y_i(n);g_i(n)\right)\right\}$, respectively. Our non-convex integer nonlinear problem is effectively a combinatorial search problem. We need to prove the following proposition before describing our approach.

Proposition 1. *Let $x_1,\ldots,x_n$ be n positive real numbers such that $x_1+\ldots+x_n \leq N$. Consider n continuous and* *differentiable decreasing functions $f_i(x_i)\ \forall\ i \in [1, n]$. If the minimum value of $\max_{i=1}^{n} f_i(x_i)$ is v at $(x'_1, \ldots, x'_n)$, then $x'_1 + \ldots + x'_n = N$ and $f_1(x'_1) = \ldots = f_n(x'_n) = v$.*

Proof: Let $s' = \sum_{i=1}^{n} x'_i$. If $s' < N$, then we can define a new set of real numbers $x''_i = x'_i + (N - s')/n$ and use them as arguments for decreasing functions. Therefore, this will lower the value of $\max_{i=1}^{n} f_i(x_i)$ which contradicts the assumption that the minima occurs for $(x'_1, \ldots, x'_n)$.

Assume $f_1(x'_1) = \ldots = f_n(x'_n) = v$ is not true. Let the two largest unequal values (in descending order) be $f_i(x'_i)$ and $f_j(x'_j)$. There exists a $\delta < x'_j$ such that $f_i(x'_i) > f_i(x'_i + \delta) > f_j(x'_j - \delta) > f_j(x'_j)$. Once again, we can decrease the value of $\max_{i=1}^{n} f_i(x_i)$. This contradicts the assumption. We can continue choosing a sequence of δ until all values become identical.

Subroutine 1: Network resource allocation for end-to-end delay threshold $\hat{\theta}$

```
$\theta_0 = \hat{\theta} - \delta_{\text{PROC}}$, $\phi = 0.5$, $\Delta_{\text{UL}} = \theta_0(1-\phi)$, $\Delta_{\text{DL}} = \theta_0\phi$
for $it \leftarrow 1$ to $iter_1$ do
    for $k \leftarrow 1$ to $C$ do
        $x_k(n)$ := The lowest integer value that makes the approx. delay function (7a) less than $\Delta_{\text{UL}}$
    for $i \leftarrow 1$ to $M$ do
        $y_i(n)$ := The lowest integer value that makes the approx. delay function (7b) less than $\Delta_{\text{DL}}$
    $y_{\text{VR}}(n)$ := The lowest integer value that makes the approx. delay function (7c) less than $\Delta_{\text{DL}}$
    Check RB constraint for both uplink and downlink
    if Constraint violated on both sides then
        Delay violation cannot be avoided.
        Return Null
    else if Constraint violated on one side then
        Use Binary Search to tune $\phi$.
        Update $\Delta_{\text{DL}}$ and $\Delta_{\text{UL}}$
    else
        Return $y_{\text{VR}}(n), \{y_1(n), \cdots, y_M(n)\}, \{x_1(n), \cdots, x_C(n)\}$
Return $y_{\text{VR}}(n), \{y_1(n), \cdots, y_M(n)\}, \{x_1(n), \cdots, x_C(n)\}$
```

Algorithm 1: Network resource allocation corresponding to n^{th} video transmission

```
Input: $C$, $M$, $\theta$, $N_{\text{RB}}$, $\Delta f$, $\delta_{\text{PROC}}$, $P_{\text{cam}}$, $P_{\text{B}}$, $s_{\text{camera}}$, $\alpha$, $s_{\text{control}}$, $s_{\text{VR}}$, Channel gains, iter
$\hat{\theta}_{\text{low}} = \delta_{\text{PROC}} + \epsilon$, $\hat{\theta}_{\text{high}} = \theta$
Keep doubling $\hat{\theta}_{\text{high}}$ till we find a RB allocation using Subroutine 1.
$\hat{\theta} = (\hat{\theta}_{\text{high}} + \hat{\theta}_{\text{low}})/2$
for $it \leftarrow 1$ to $iter_2$ do
    Use Binary Search to adjust $\hat{\theta}$ within the range $[\hat{\theta}_{\text{low}}, \hat{\theta}_{\text{high}}]$ and use Subroutine 1 to find RB allocation.
Return the final RB allocation
```

The approximate delay functions are continuous and differentiable decreasing functions of the RBs. Using the Proposition 1, our objective should be to allocate RBs among cameras in such a way that their respective delays to reach the 5G base station are as close to the same as possible. Similarly, the VR video feed should reach the surveillance system around the same time the AGVs receive their motion controls. In the next subsection, we give a detailed explanation of our RB allocation method.

B. Algorithm for network resource allocation

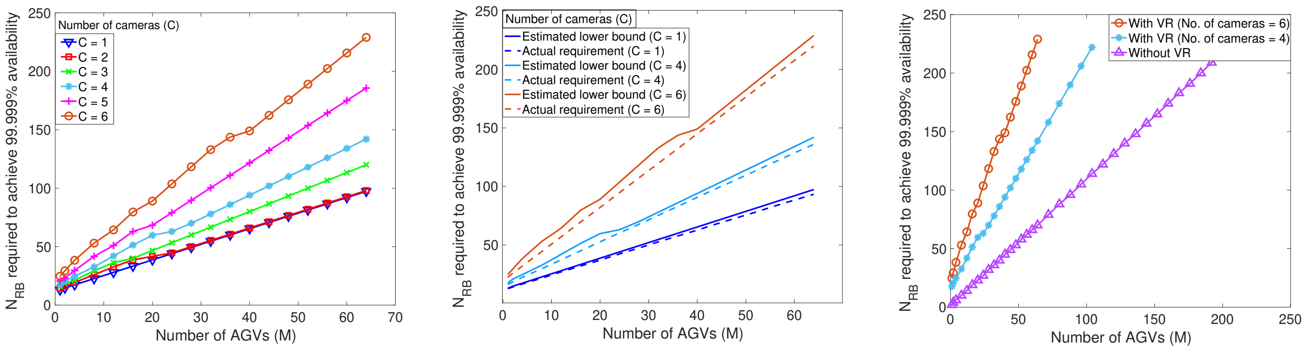
Given an end-to-end delay threshold (denoted $\hat{\theta}$), Algorithm 1 is used to find an RB allocation that satisfies the constraint. The total allowable transmission delay (denoted by θ_0) is calculated by subtracting the processing delay from the end-to-end delay threshold $\hat{\theta}$. We allocate a portion (denoted ϕ) of the allowable transmission delay for uplink transmissions and the rest for downlink transmissions. Initially, ϕ is set to 0.5. During iterations of the algorithm, the ϕ value is adjusted to increase/decrease transmission time on one side (uplink or downlink) as necessary due to the constraint on the availability of the RBs. For each iteration of the algorithm, we try to assign RBs in such a way that the uplink and downlink communication delays match their respective target value. Note that all transmission delays have the form $a_1/(x\log(1 + \frac{a_2}{x})) = \Delta_{\text{t}}$, where x corresponds to the number of RBs for a particular transmission. The solution to this equation is given by

$$x = -\frac{a_1 a_2}{a_1 + a_2\Delta_{\text{t}}\text{ProductLog}\left(-\frac{a_1\exp\left(-\frac{a_1}{a_2\Delta_{\text{t}}}\right)}{a_2\Delta_{\text{t}}}\right)}, \quad (8)$$

where the ProductLog(z) function [36], also known as the Lambert W function, gives the solution for $we^w = z$.

Algorithm 1 is used to minimize our original objective, *i.e.*, to minimize the end-to-end latency. Given the end-to-end delay requirement of our system (θ), we try to find an RB allocation that satisfies the constraint, using Subroutine 1. If no such allocation is possible, we again search for an RB allocation using a delay threshold of 2θ. We continue to double until we can find an RB allocation. Once we are successful, we find the upper bound of the minimum end-to-end delay. Then we use the binary search method to adjust $\hat{\theta}$ within the range and pass it on to the Subroutine 1 to find the corresponding RB allocation. As we go through the iterations of Algorithm 1, we gradually approach the minimum possible end-to-end delay and find the corresponding RB allocation. The network resource allocation is done at the 5G base station, which is virtually co-located with the edge server [28] as the edge server and the 5G base station have a wired connection between them. So, the network resource allocation algorithm can run on the edge server. The following data are needed to run the algorithm.

- **The channel conditions inside the factory floor:** In 5G NR, a User Equipment (UE) reports the Channel State Information (CSI) to the 5G base station. The periodicity



(a) Required N_{RB} vs. Number of AGVs (b) Estimated N_{RB} vs Number of AGVs (c) N_{RB} vs M with or without VR

Fig. 4: N_{RB} vs Number of AGVs

of the CSI report flow can be of three types - periodic, aperiodic, and semipersistant [37]. In our simulation of the industrial system, we have used a periodic CSI report with a periodicity of 50 ms. Control packets for CSI reporting are sent via a narrowband channel. Channel conditions inside a factory floor do not change very frequently [38], so the periodicity of the CSI reporting might be a bit larger. We have added the details about CSI reporting in the revised version of the manuscript.

- **The positions of the AGVs:** Based on the most recent video feed from the camera system on the factory floor and the motion control decisions taken on the edge server, the positions of the AGVs are readily available at the edge server.

The number of iterations used in Subroutine 1 and Algorithm 1 is indicated by iter_1 and iter_2 respectively. For each call to Subroutine 1, the binary search is executed iter_1 times. Therefore, the maximum number of computations for each call to Subroutine 1 is of order $O(\text{iter}_1 \cdot (C + M))$. In Algorithm 1, the binary search is performed most iter_2 times, and at each step, Subroutine 1 is called. The constant terms that are not related to the design of the industrial system (such as - iter_1 and iter_2), can be disregarded in the time complexity analysis. So, the overall worst-case complexity of Algorithm 1 is $O(C + M)$.

VII. Simulation results

The simulation results section is categorized into two subsections. In the first subsection, we demonstrate the scalability issues that may arise from the number of cameras used on the factory floor to create the VR 360° video feed for remote surveillance of the automation process, and the number of AGVs involved in the manufacturing process. In the second subsection, we compare the performance of the resource allocation scheme developed in Sections VI-A and VI-B with the existing method. The values of the parameters used in our simulation set-up used are given in Table IV. We simulate the different communication streams and the movement of the AGVs on the factory floor using OMNet++ version 6.0 integrated with SimuLTE that provides a framework mimicking 5G network. The machine used for the simulations is a MacBook Pro with 3.1 GHz Dual-Core Intel Core i5 processor, Intel Iris Plus Graphics 650 1536 MB, and 8 GB RAM.

TABLE IV: Simulation setup

Parameter	Value
Area of factory floor	500m×500m
Surveillance camera resolution	720 × 480 at 30 fps [39]
Data generated by each camera	25 Mbps
Size of the motion control messages sent to each AGV	500 bytes [3]
Subcarrier spacing (Δf) of the RBs	15 kHz
Dimension of each AGV	750mm(L)×550mm(W) ×200mm(H) [40]
Maximum speed of each AGV	4 m/s
Parameters of the scalarization (Sec. V-B)	0.5 each

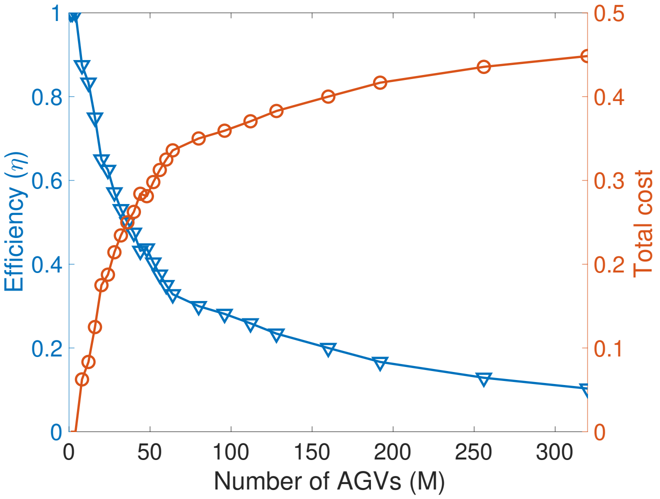
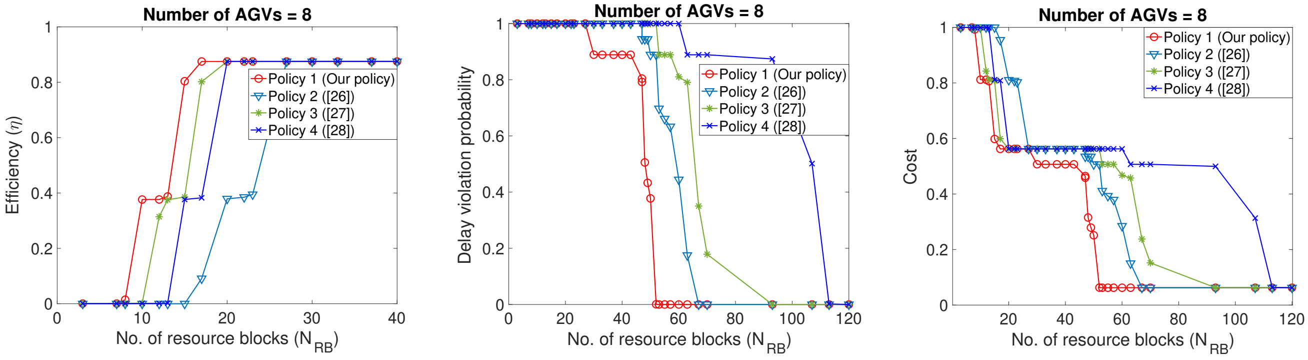


Fig. 5: Efficiency and Cost vs Number of AGVs

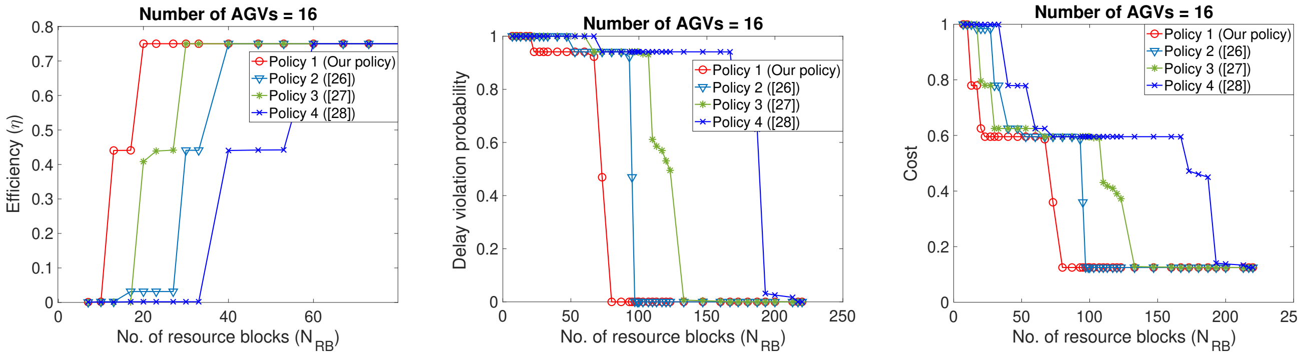
A. Scalability of Industry 4.0 system

Network resource requirement of the Industry 4.0 system considered in this work is heavily dependent on the number of cameras used to monitor the factory floor. Scalability in terms of bandwidth requirement is a crucial issue for the factories of the future. Another critical scalability problem arises from the number of AGVs operating on a factory floor, which affects



(a) η vs N_{RB} (b) Delay violation probability vs N_{RB} (c) Cost vs N_{RB}

Fig. 6: Comparison of performance metrics for 8 AGVs on the factory floor. We compare our policy (Policy 1) with three existing resource allocation policies: Policy 2 [25] (centralized resource allocation based on throughput conditions of industrial devices); Policy 3 [26] (fractions of available RBs allocated to URLLC and eMBB applications; and Policy 4 [27] (greedy allocation strategy).



(a) η vs N_{RB} (b) Delay violation probability vs N_{RB} (c) Cost vs N_{RB}

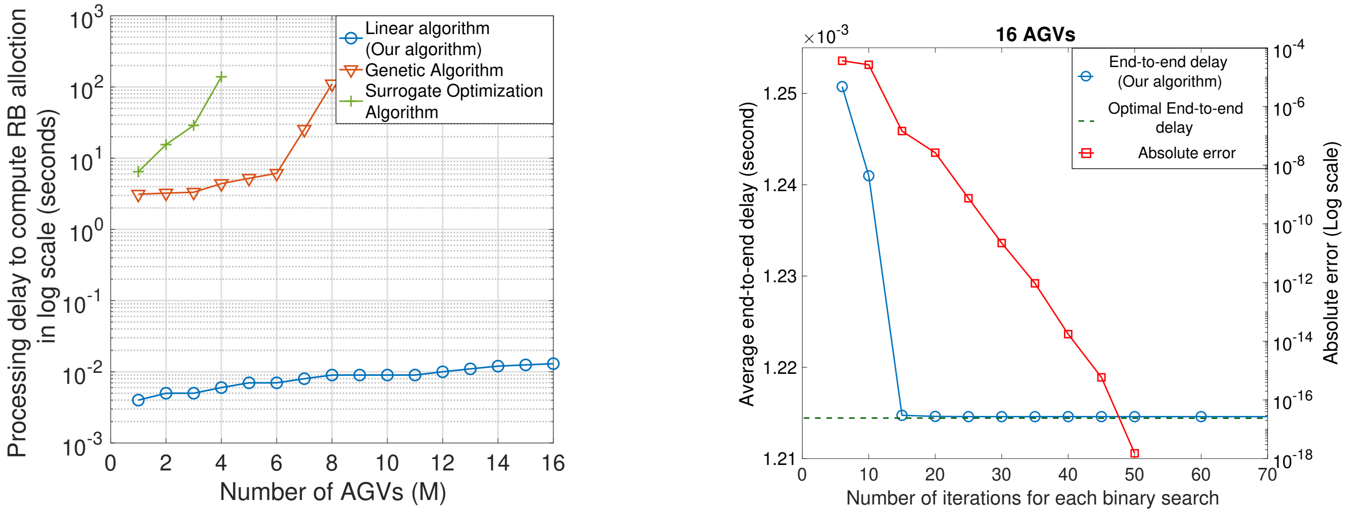
Fig. 7: Comparison of performance metrics for 16 AGVs on the factory floor. We compare our policy (Policy 1) with three existing resource allocation policies: Policy 2 [25] (centralized resource allocation based on throughput conditions of industrial devices); Policy 3 [26] (fractions of available RBs allocated to URLLC and eMBB applications; and Policy 4 [27] (greedy allocation strategy).

the safety condition of the floor. The following results provide an understanding of the scalability of our model.

1) Effect of number of cameras used to create VR 360° *video:* For a smooth VR 360° video experience, VR developers and industries concur that application end-to-end latency should be less than 20 ms in order for the Motion-To-Photon latency (MTP) to be indiscernible [11], [12], [41]. In our work, the end-to-end latency threshold (θ) is kept at 15 ms. The processing delay to create VR 360° video by stitching video feed from the cameras on the factory floor depends on both the quality of the video feed from cameras, the quality of 360° video, and the number of cameras. The video-stitching method described in [42] requires on average 13 ms to stitch videos from 6 cameras. The amount of data generated by the cameras on the factory floor is linearly dependent on the number of cameras, and the corresponding delay follows a nearly linear curve [43].

In our simulation, as the number of cameras increases, the time it takes to process video stitching and blending increases. Because the end-to-end delay must be met, the total transmission time must be shortened. With increasing number of cameras, the amount of data to be transmitted to the 5G base station increases. The increased amount of data, along with less time for transmission, calls for increased demand for resource blocks. For a fixed number of AGVs, Fig. 4a showes how the demand for the total number of RBs (N_{RB}) gradually increases as more cameras are used to create the VR 360° video. We use the resource allocation scheme described in Algorithm 1 to allocate RBs to cameras, AGVs, and the surveillance system. It is evident from Fig. 4a that N_{RB} increases as we continue to add more AGVs to the factory floor. We need to add more RBs to accommodate more motion control inputs that need to be delivered to the additional AGVs in order to maintain the end-to-end delay threshold.

In Fig. 4b, we compare the estimated number of RBs, calculated using the method described in Section V, with the actual number of RBs required to achieve 99.999% availability. Although the lower bound is not very strict when the number of AGVs and cameras is higher, it still provides a good understanding of the scalability of a factory floor. In Fig. 4c, we show that the scalability in terms of the number of AGVs nearly doubles when the VR maintenance system is excluded from the Industry 4.0 system. In the absence of a VR system, the limitation on scalability arises from the physical architecture of 5G NR.



(a) Processing delay to allocate network resources using different algorithms vs Number of AGVs

(b) End-to-end delay accuracy vs Number of iterations for each binary search in Subroutine 1 and Algorithm 1

Fig. 8: Performance of our resource allocation policy

2) Effect of number of AGVs on efficiency and cost of the automation process: In this sub-subsection, we show how the value of the objective function (referred to here as the cost of the Industry 4.0 system), defined in Section V-B, changes as we increase the number of AGVs on the factory floor. Both scalarization parameters of the objective function (Equation (5)) are set to 0.5. The first term of the objective function is the deficiency (defined as $1-$*efficiency*) of the system. As described in Section IV-C, if efficiency is reduced, the chance of a collision event on the factory floor increases. The second term of the objective function is related to the number of round-trip delay violation events. These delay violations negatively affect the VR video feed delivery to the maintenance room, as well as motion control delivery to AGVs. For simulation purpose, the number of cameras on the factory floor is taken to be 4.

The threshold of delay violation probability for our problem is less than 0.001, as discussed in Section II. Using the method described in Section V, we estimate the lower bound for the number of total resource blocks (N_{RB}) necessary to achieve this criterion. With knowledge of the lower bound of N_{RB}, we provide a sufficient number of RBs to ensure that the round-trip delay violation is kept under 0.001. As a result, the second term of the objective function (see Section V-B) contributes very little to the overall cost.

As described in Section IV-C, the efficiency of the automation process is defined as the ratio between the total distance covered by the AGVs dictated by the motion controller on the edge server and the distance covered by the AGVs at maximum velocity for the duration of the automation process. The locations of the starting and end points of the AGV routes contribute to the fluidity of the automation process. For the reproducibility of our simulation, we generate the starting and end points of the AGVs using a random seed in OMNet++ once and keep the locations fixed throughout the simulation process. When there is a single AGV on the factory floor, the efficiency is very close to 1 (Fig. 5). As we increase the number of AGVs, the efficiency drops sharply to about 65 AGVs, and then the rate of decrease in efficiency slows (Fig. 5). An increase in the number of AGVs increases the number of interactions among the AGVs. As a result, the controller must slow down the AGVs (and, on occasion, completely stop the AGVs) to avoid collisions. Therefore, efficiency decreases and is expected to reach zero when there are so many AGVs that the factory floor comes to a standstill. So, with an increase in the number of AGVs, the efficiency approaches 0, i.e. deficiency approaches 1, and the overall cost approaches just very slightly above 0.5. Due to the finite area of the factory floor in our simulation setup, arbitrarily increasing the number of AGVs makes little sense, as most AGVs are stuck during the automation process. So, the maximum number of AGVs in our simulation is 300. Therefore, the theoretical upper limit of the cost (slightly greater than 0.5) is not visible in Fig. 5.

B. Performance of our resource allocation policy

In this subsection, we demonstrate the performance of our resource allocation scheme, as well as the efficiency of our algorithm to implement this policy. Finally, we show the trade-off between processing time and accuracy of our algorithm.

1) Comparison of different network resource allocation policy: In this sub-subsection, we compare how well or worse our resource allocation policy (refer to Section VI-A) holds compared to policies in the existing literature. Especially, we compare our resource allocation policy with the policies described in [25], [26], and [27]. In the rest of this subsection, we denote our policy as Policy 1 and the policies described in [25], [26] and [27] as Policy 2, Policy 3, and Policy 4, respectively.

- *Policy 2:* In [25], the authors proposed a centralized resource allocation policy where the network resources are distributed among the industrial devices (sensors,

actuators, etc.) such as to maximize the sum throughput. Although a heuristic approach is given to find the solution to this optimization problem in [25], it is not clear whether it gives the optimal solution.
- *Policy 3:* In [26], a fraction (denoted by α) of available RBs are cyclically allocated to URLLC applications. The rest of the RBs are allocated to the eMBB applications, once again in a round-robin fashion. In our system, during downlink communications, we can classify the transmission to the surveillance system as URLLC coupled with the eMBB application, whereas the transmissions to the AGVs can be deemed as URLLC applications.
- *Policy 4:* The authors of [27] proposed a greedy allocation strategy for the allocation of RB to active industrial devices. The greedy strategy is based on a metric that depends on three key parameters: (i) the amount of data in the queues of the devices, (ii) the data generation rate at the devices, and (iii) the previous RB allocation history.

In Fig. 6, we consider 8 AGVs on the factory floor. We implement the aforementioned policies in OMNet++. For each policy, we use 10 different random number generating seeds to simulate Industry 4.0 under different channel conditions and run each iteration for 1 hour. We track the number of delay violation events for each iteration and finally calculate the delay violation probability for different policies. The first term of equation (5) can be computed using equation (2). The window size for the calculation of *efficiency* is taken to be 100. Finally, we can calculate the cost function using Equation (5). Throughout the simulation process, the number of cameras is set to be 4.

From Fig. 6a it is clear that we can reach the maximum possible efficiency if enough RBs are available on the factory floor. As mentioned in. Section II, we control the data injection process on the factory floor. As a result, our resource allocation policy requires far fewer RBs to achieve maximum efficiency. For a fixed number of RBs available on the factory floor, a fixed proportion to the surveillance system is not optimal. It may cause under-utilization of RBs by allocating more than required RBs to either the surveillance system or the AGVs. For similar reasons, we can see that our policy achieves the delay violation requirement with fewer RBs compared to Policy 2 and 3 (Fig. 6b). As a result, we can also achieve the minimum possible overall cost using fewer RBs (Fig. 6c). Similarly, for 16 AGVs on the factory floor, our resource allocation policy performs better (see Fig. 7).

2) Performance comparison of different algorithms for implementing our resource allocation policy: In this sub-subsection, we compare the performance of different algorithms to solve the network resource allocation problem in Section VI. We compare our algorithm (described in Section VI-B) with two existing algorithms to solve a global optimization problem: (1) genetic algorithm and (2) surrogate optimization algorithm. For the simulation setup, we keep the number of AGVs and the number of cameras to be 8 and 4 respectively. In Fig. 8a, we plot the processing time (on a logarithmic scale) needed to solve the network resource allocation problem using different algorithms. The approximate delay functions (details in Section VI) are monotonically decreasing functions of the number of RBs associated with them. If the number of RBs can take real values (instead of strict integers), the approximate delay functions follow the Proposition 1. Our algorithm is designed based on the approximate delay functions and Proposition 1, and can efficiently find the solution to the network resource allocation problem that minimizes our objective described in Equation (6). As the number of AGVs increases, both the Genetic Algorithm and Surrogate Optimization Algorithm do not converge to a solution even after an hour. Therefore, we truncated the curves in Fig. 8a. A generic global optimizer is much slower and cannot be used to assign RBs in real time.

3) Accuracy of our algorithm: In Fig. 8b, we show how the average end-to-end delay achieved by our resource allocation policy is affected as we vary the number of iterations for the binary searches used in Subroutine 1 and Algorithm 1. We consider 16 AGVs on the factory floor for this particular scenario. The number of cameras is taken to be 4. To obtain the baseline for comparing the accuracy of our policy (shown in dotted line in Fig. 8b), we solve the myopic version of our optimization problem (Equation (6)) while allocating network resources in the OMNet++ simulation. It is evident form Fig. 8b, with the increase in the number of iterations for both binary searches, the absolute error of the average end-to-end delay decreases. Empirically, we get one more decimal digit of accuracy as we increase the number of iterations to 50 for both binary searches.

VIII. Conclusions and Future Works

In this work, we proposed a network resource allocation algorithm for a futuristic factory equipped with stationary robots, AGVs, and VR surveillance. First, we model a fully automated factory floor assisted by a camera-based monitoring system and VR surveillance. The video feed from the camera system is sent to the edge server depending on the channel and the safety conditions on the factory floor. Then, we studied two key QoS parameters: end-to-end delay and safety. We presented a statistical method to estimate the number of RBs necessary to satisfy QoS metrics 99.999% over time. Our resource allocation policy achieves the QoS objectives by employing fewer RBs compared to the existing network resource allocation policy for Industry 4.0 systems [26]. Our algorithm solves the resource allocation problem with linear complexity in the total number of AGVs and cameras. Our algorithm also performs better than well-known global optimizers, such as genetic and surrogate optimization algorithms. Our network resource allocation scheme based on a controlled data injection process to the 5G network uses 25% to 30% fewer network resources to achieve the desired QoS, compared to the existing resource allocation schemes.

In our future work, our aim is to analyze an automated industrial system where human workers can intervene remotely using an augmented reality system. The network resource allocation policy should be designed to handle the stochastic nature of events of human interaction.

References

[1] F. Yang and S. Gu, "Industry 4.0, a Revolution That Requires Technology and National Strategies," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1311–1325, 2021.

[2] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industrie 4.0," *Wirtschaftsinformatik*, vol. 56, no. 4, pp. 261–264, 2014.

[3] L. Grosjean, O. Dobrijevic, K. Landernäs, R. Kulläng, P. Falco, S. Azhar, A. Rostami, S. Schmitt, N. König, P. Mohanram, B. Sayrac, F. Parzysz, G. Madhusudan, A. M. G. Serrano, S. Cerovic, S. Destouet-Roblot, and S. Inca, "Forward Looking Smart Manufacturing Use Cases, Requirements," 2020.

[4] "5G for Connected Industries and Automation," https://www.zvei.org/en/press-media/publications/5g-for-connected-industries-and-automation-white-paper-second-editon, 2019.

[5] S. K. Rao and R. Prasad, "Impact of 5G Technologies on Industry 4.0," *Wireless personal communications*, vol. 100, no. 1, pp. 145–159, 2018.

[6] P. Ferrari, S. Rinaldi, E. Sisinni, F. Colombo, F. Ghelfi, D. Maffei, and M. Malara, "Performance Evaluation of Full-cloud and Edge-cloud Architectures for Industrial IoT Anomaly Detection Based on Deep Learning," in *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT)*. IEEE, 2019, pp. 420–425.

[7] R. Bacci di Capaci and C. Scali, "A Cloud-based Monitoring System for Performance Assessment of Industrial Plants," *Industrial & Engineering Chemistry Research*, vol. 59, no. 6, pp. 2341–2352, 2020.

[8] W. M. H. Heemels, A. R. Teel, N. Van de Wouw, and D. Nešić, "Networked Control Systems with Communication Constraints: Tradeoffs Between Transmission Intervals, Delays and Performance," *IEEE Transactions on Automatic control*, vol. 55, no. 8, pp. 1781–1796, 2010.

[9] T. Charalambous, A. Ozcelikkale, M. Zanon, P. Falcone, and H. Wymeersch, "On the Resource Allocation Problem in Wireless Networked Control Systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 4147–4154.

[10] Z. Miljković, N. Vuković, M. Mitić, and B. Babić, "New Hybrid Vision-Based Control Approach for Automated Guided Vehicles," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 1, pp. 231–249, 2013.

[11] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "VR is on the Edge: How to Deliver 360 Videos in Mobile Networks," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 30–35.

[12] K. Mania, B. D. Adelstein, S. R. Ellis, and M. I. Hill, "Perceptual Sensitivity to Head Tracking Latency in Virtual Environments with Varying Degrees of Scene Complexity," in *Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*, 2004, pp. 39–47.

[13] E. A. Oyekanlu, A. C. Smith, W. P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, D. J. Kuhn, J. D. Mcghinnis, S. C. Buonavita, N. A. Looper *et al.*, "A Review of Recent Advances in Automated Guided Vehicle Technologies: Integration Challenges and Research Areas for 5G-Based Smart Manufacturing Applications," *IEEE access*, vol. 8, pp. 202 312–202 353, 2020.

[14] R. Masoni, F. Ferrise, M. Bordegoni, M. Gattullo, A. E. Uva, M. Fiorentino, E. Carrabba, and M. Di Donato, "Supporting Remote Maintenance in Industry 4.0 Through Augmented Reality," *Procedia manufacturing*, vol. 11, pp. 1296–1302, 2017.

[15] L. Damiani, M. Demartini, G. Guizzi, R. Revetria, and F. Tonelli, "Augmented and Virtual Reality Applications in Industrial Systems: A Qualitative Review Towards the Industry 4.0 Era," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 624–630, 2018.

[16] Z. Guo, D. Zhou, Q. Zhou, X. Zhang, J. Geng, S. Zeng, C. Lv, and A. Hao, "Applications of Virtual Reality in Maintenance During the Industrial Product Lifecycle: A Systematic Review," *Journal of Manufacturing Systems*, vol. 56, pp. 525–538, 2020.

[17] I. Malỳ, D. Sedláček, and P. Leitao, "Augmented Reality Experiments with Industrial Robot in Industry 4.0 Environment," in *2016 IEEE 14th international conference on industrial informatics (INDIN)*. IEEE, 2016, pp. 176–181.

[18] J. Kovar, K. Mouralova, F. Ksica, J. Kroupa, O. Andrs, and Z. Hadas, "Virtual Reality in Context of Industry 4.0 Proposed Projects at Brno University of Technology," in *2016 17th international conference on mechatronics-mechatronika (ME)*. IEEE, 2016, pp. 1–7.

[19] J. Rosales, S. Deshpande, and S. Anand, "IIoT Based Augmented Reality for Factory Data Collection and Visualization," *Procedia Manufacturing*, vol. 53, pp. 618–627, 2021.

[20] T. M. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and M. Vilar-Montesinos, "A Fog Computing and Cloudlet Based Augmented Reality System for the Industry 4.0 Shipyard," *Sensors*, vol. 18, no. 6, p. 1798, 2018.

[21] D. Mourtzis, V. Siatras, J. Angelopoulos, and N. Panopoulos, "An Augmented Reality Collaborative Product Design Cloud-Based Platform in the Context of Learning Factory," *Procedia Manufacturing*, vol. 45, pp. 546–551, 2020.

[22] Y. Wang, T. Yu, and K. Sakaguchi, "Context-Based MEC Platform for Augmented-Reality Services in 5G Networks," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–5.

[23] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold, K. Sandström, and M. Behnam, "Delay Mitigation in Offloaded Cloud Controllers in Industrial IoT," *IEEE Access*, vol. 5, pp. 4418–4430, 2017.

[24] W. Dai, H. Nishi, V. Vyatkin, V. Huang, Y. Shi, and X. Guan, "Industrial Edge Computing: Enabling Embedded Intelligence," *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 48–56, 2019.

[25] I. O. Sanusi, K. M. Nasr, and K. Moessner, "Radio Resource Management Approaches for Reliable Device-to-device (D2D) Communication in Wireless Industrial Applications," *IEEE transactions on cognitive communications and networking*, vol. 7, no. 3, pp. 905–916, 2020.

[26] M. Mhedhbi, M. Morcos, A. Galindo-Serrano, and S. E. Elayoubi, "Performance Evaluation of 5G Radio Configurations for Industry 4.0," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2019, pp. 1–6.

[27] H. Peng, W. Tärneberg, and M. Kihl, "Latency-Aware Radio Resource Allocation Over Cloud RAN for Industry 4.0," in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–8.

[28] G. Brown, P. Analyst, and H. Reading, "Ultra-Reliable Low-Latency 5G for Industrial Automation," *Technol. Rep. Qualcomm*, vol. 2, p. 52065394, 2018.

[29] "5G NR Resource Block Definition and RBs Calculation," https://www.techplayon.com/nr-resource-block-definition-and-rbs-calculation/, accessed: 2023-06-29.

[30] H. Holma, A. Toskala, and T. Nakamura, *5G Technology: 3GPP New Radio*. John Wiley & Sons, 2020.

[31] "3rd Generation Partnership Project (Release 17)," https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/17.02.00_60/ts_138211v170200p.pdf, 2022, [Online; accessed 15-Nov-2022].

[32] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward Low-Latency and Ultra-Reliable Virtual Reality," *IEEE Network*, vol. 32, no. 2, pp. 78–84, 2018.

[33] C. E. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[34] I. Al-Mejibli and S. Al-Majeed, "Challenges of Using MIMO Channel Technology in 5G Wireless Communication Systems," in *2018 Majan International Conference (MIC)*. IEEE, 2018, pp. 1–5.

[35] A. Virdis, G. Stea, and G. Nardini, "SimuLTE - A Modular System-level Simulator for LTE/LTE-A Networks based on OMNeT++," in *2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)*. IEEE, 2014, pp. 59–70.

[36] J. H. Lambert, "Observationes Variae in Mathesin Puram," *Acta Helvetica*, vol. 3, no. 1, pp. 128–168, 1758.

[37] "5G/NR - CSI Report," https://www.sharetechnote.com/html/5G/5G_CSI_Report.html, accessed: 2023-06-29.

[38] D. Sexton, M. Mahony, M. Lapinski, and J. Werb, "Radio Channel Quality in Industrial Wireless Sensor Networks," in *2005 Sensors for Industry Conference*. IEEE, 2005, pp. 88–94.

[39] "Video Surveillance Bandwidth Requirements - Calculation of Utilization," https://www.mistralsolutions.com/articles/video-surveillance-bandwidth-requirements-calculation-utilization/, 2020, [Online; accessed 15-Nov-2022].

[40] "Automated Guided Vehicle," https://4jmsolutions.com/equipment/factory-automatization-solutions/smart-factory/automated-guided-vehicle/, 2022, [Online; accessed 15-Nov-2022].

[41] "Latency Mitigation Strategies (by John Carmack)," https://danluu.com/latency-mitigation/, 2013, [Online; accessed 15-Nov-2022].

[42] W.-T. Lee, H.-I. Chen, M.-S. Chen, I.-C. Shen, and B.-Y. Chen, "High-Resolution 360 Video Foveated Stitching for Real-time VR," in *Computer Graphics Forum*, vol. 36, no. 7. Wiley Online Library, 2017, pp. 115–123.

[43] N. Sidaty, P.-L. Cabarat, W. Hamidouche, D. Menard, and O. Deforges, "Performance and Computational Complexity of the Future Video Coding," in *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2018, pp. 31–36.