Task-Oriented Joint Design of Communication and Computing for Internet of Skills

Burak Kizilkaya

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

July 2023

Abstract

Nowadays, the internet is taking a revolutionary step forward, which is known as *Internet of Skills*. The Internet of Skills is a concept that refers to a network of sensors, actuators, and machines that enable knowledge, skills, and expertise delivery between people and machines, regardless of their geographical locations. This concept allows an immersive remote operation and access to expertise through virtual and augmented reality, haptic communications, robotics, and other cutting-edge technologies with various applications, including remote surgery and diagnosis in healthcare, remote laboratory and training in education, remote driving in transportation, and advanced manufacturing in Industry 4.0.

In this thesis, we investigate three fundamental communication requirements of Internet of Skills applications, namely *ultra-low latency*, *ultra-high reliability*, and *wireless resource utilization efficiency*. Although 5G communications provide cutting-edge solutions for achieving ultra-low latency and ultra-high reliability with good resource utilization efficiency, meeting these requirements is difficult, particularly in long-distance communications where the distance between source and destination is more than 300 km, considering delays and reliability issues in networking components as well as physical limits of the speed of light. Furthermore, resource utilization efficiency must be improved further to accommodate the rapidly increasing number of mobile devices. Therefore, new design techniques that take into account both communication and computing systems with the task-oriented approach are urgently needed to satisfy conflicting latency and reliability requirements while improving resource utilization efficiency.

First, we design and implement a 5G-based teleoperation prototype for Internet of Skills applications. We presented two emerging Internet of Skills use cases in healthcare and education. We conducted extensive experiments evaluating local and long-distance communication latency and reliability to gain insights into the current capabilities and limitations. From our local experiments in laboratory environment where both operator and robot in the same room, we observed that communication latency is around 15 ms with a 99.9% packet reception rate (communication reliability). However, communication latency increases up to 2 seconds in long-distance scenarios (between the UK and China), while it is around 50-300 ms within the UK experiments. In addition, our observations revealed that communication reliability and overall system performance do not exhibit a direct correlation. Instead, the number of consecutive packet drops emerged as the decisive factor influencing the overall system performance and user quality of experience. In light of these findings, we proposed a two-way timeout approach. We discarded stale packets to mitigate waiting times effectively and, in turn, reduce the latency. Nevertheless, we observed that the proposed approach reduced latency at the expense of reliability, thus verifying the challenge of the conflicting latency and reliability requirements.

Next, we propose a task-oriented prediction and communication co-design framework to meet conflicting latency and reliability requirements. The proposed framework demonstrates the task-oriented joint design of communication and computing systems, where we considered packet losses in communications and prediction errors in prediction algorithms to derive the upper bound for overall system reliability. We revealed the tradeoff between overall system reliability and resource utilization efficiency, where we consider 5G NR as an example communication system. The proposed framework is evaluated with real-data samples and generated synthetic data samples. From the results, the proposed framework achieves better latency and reliability tradeoff with a 77.80% resource utilization efficiency improvement compared to a task-agnostic benchmark. In addition, we demonstrate that deploying a predictor at the receiver side achieves better overall reliability compared to a system that predictor at the transmitter.

Finally, we propose an intelligent mode-switching framework to address the resource utilization challenge. We jointly design the communication, user intention recognition, and mode-switching systems to reduce communication load subject to joint task completion probability. We reveal the tradeoff between task prediction accuracy and task observation length, showing that higher prediction accuracy can be achieved when the task observation length increases. The proposed framework achieves more than 90% task prediction accuracy with 60% observation length. We train a DRL agent with real-world data from our teleoperation prototype for mode-switching between teleoperation and autonomous modes. Our results show that the proposed framework achieves up to 50% communication load reduction with similar task completion probability compared to conventional teleoperation.

Declaration

I certify that the thesis presented here for a PhD degree of the University of Glasgow is solely my own work other than where I have clearly indicated that it is the work of others and that the thesis has not been edited by a third party beyond what is permitted by the University's PGR Code of Practice.

The copyright of this thesis rests with the author. No quotation from it is permitted without full acknowledgement.

I declare that the thesis does not include work forming part of a thesis presented successfully for another degree.

I declare that this thesis has been produced in accordance with the University of Glasgow's Code of Good Practice in Research.

Signature: *Burak Kizilkaya*
Date: *July 2023*

List of Publications

1. **Burak Kizilkaya**, Changyang She, Guodong Zhao, and Muhammad Ali Imran. "Intelligent Mode-Switching Framework for Long Distance Teleoperation." (2023), Under Review.

2. **Burak Kizilkaya**, Olaoluwa Popoola, Guodong Zhao, and Muhammad Ali Imran. "On the Implementation of 5G-based Low-Latency Teleoperation: Two-way Timeout Approach." (2023), Under Review.

3. **Burak Kizilkaya**, Changyang She, Guodong Zhao and Muhammad Ali Imran, "Task-Oriented Prediction and Communication Co-Design for Haptic Communications." IEEE Transactions on Vehicular Technology, doi: 10.1109/TVT.2023.3247442.

4. **Burak Kizilkaya**, Chanyang She, Guodong Zhao, and Muhammad Ali Imran. "Task-Oriented Prediction and Communication Co-Design.", DOI: `http://dx.doi.org/10.5525/gla.researchdata.1392`, [Data Collection], (2023).

5. Rongyu Yu, **Burak Kizilkaya**, Zhen Meng, Emma Li, Guodong Zhao, and Muhammad Ali Imran. "Robot Mimicry Attack on Keystroke-Dynamics User Identification and Authentication System." In IEEE International Conference on Robotics and Automation (ICRA), London, UK, (2022).

6. **Burak Kizilkaya**, Ekaterina Khirakova, Elliot Cooper, Oluwakemi Olusholaq, Guodong Zhao, Bridgette Wessels, Jennifer Challinor, and Muhammad Ali Imran. "Robotics-in-Care: Capabilities, Challenges, and Opportunities." In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, (2022).

7. **Burak Kizilkaya**, Guodong Zhao, Yusuf A. Sambo, Liying Li, and Muhammad Ali Imran. "5G-enabled education 4.0: Enabling technologies, challenges, and solutions." IEEE Access 9 (2021): 166962-166969.

8. **Burak Kizilkaya**, Bo Chang, Shuja Ansari, Yusuf A. Sambo, Guodong Zhao, and Muhammad Ali Imran. "Age of Control Process for Real-Time Wireless Control." In Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1-5. IEEE, (2021).

9. **Burak Kizilkaya**, Zhen Meng, Guodong Zhao, Liying Li, and Muhammad Ali Imran. "Towards 5G-Enabled Education: Remote Laboratory and Training Prototype." In UK-RAS Conference for PhD Students & Early-Career Researchers on Robotics at Home, pp. 53-54, (2021).

10. Bo Chang, **Burak Kizilkaya**, Liying Li, Guodong Zhao, Zhi Chen, and Muhammad Ali Imran. "Effective Age of Information in Real-time Wireless Feedback Control Systems." Science China Information Sciences 64, no. 2 (2021): 1-14.

11. **Burak Kizilkaya**, Bo Chang, Guodong Zhao, and Muhammad Ali Imran. "Wireless Control for Life-Critical Actions." Engineering and Technology for Healthcare (2020): 153.

Contents

List of Tables

List of Figures

List of Acronyms

3GPP	3rd Generation Partnership Project
5G	5th Generation Cellular Communications
5G NR	5G New Radio
AWGN	Additive White Gaussian Noise
AoI	Age of Information
AI	Artificial Intelligence
AI-CoCoCo	AI-aided Communication-Control Co-design
AR	Augmented Reality
BS	Base Station
CPU	Central Processing Unit
CNN	Convolutional Neural Networks
CF	Cost Function
DQN	Deep Q-Learning
DRL	Deep Reinforcement Learning
DoF	Degrees of Freedom
DoS	Denial-of-Service
DT	Digital Twin
DTW	Dynamic Time Warping
E2E	End-to-End
eMBB	enhanced Mobile Broadband
EPC-AKA	Evolved Packet System Authentication and Key Agreement
ELU	Exponential Linear Unit
XR	Extended Reality

FNN	Feedforward Neural Networks
FCI	Franka Control Interface
GRU	Gated Recurrent Unit
GPR	Gaussian Process Regression
GAN	Generative Adversarial Networks
GPU	Graphical Processing Unit
GUI	Graphical User Interface
tanh	Hyperbolic Tangent
ITS	Intelligent Transport Systems
IoT	Internet of Things
IK	Inverse Kinematics
JND	Just Noticeable Difference
kNN	k-Nearest Neighbours
LSTM	Long Short Term Memory
ML	Machine Learning
MDP	Markov Decision Process
mMTC	massive Machine Type Communications
MSE	Mean Squared Error
mmWave	millimeter Wave
MEC	Mobile Edge Computing
MNO	Mobile Network Operator
MLP	Multi Layer Perceptron
NTP	Network Time Protocol
NCS	Networked Control Systems
PDR	Packet Delivery Ratio
PPC	Packetized Predictive Control
PID	Proportional Integral Derivative
QoE	Quality of Experience
QoS	Quality of Service

ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks
RL	Reinforcement Learning
RRMSE	Relative Root Mean Squared Error
RMSE	Root Mean Squared Error
SNR	Signal to Noise Ratio
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TimeGAN	Time-series Generative Adversarial Networks
TCP	Transmission Control Protocol
TTI	Transmission Time Interval
URLLC	Ultra Reliable Low Latency Communications
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UE	User Equipment
VR	Virtual Reality

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisors: Dr. Guodong Zhao, Prof. Muhammad Ali Imran, and Prof. Hadi Heidari. My deepest gratitude goes to Dr. Guodong Zhao for his continuous support and guidance throughout my Ph.D. studies. I have learned a lot from him. My sincere thanks also go to Prof. Muhammad Ali Imran. His warm and friendly attitude and expert advice always inspired me to make the right decisions. He was more than a supervisor to me; he was a mentor. My grateful thanks also go to Prof. Hadi Heidari for his support throughout my studies.

I would also like to thank Dr. Changyang She for his insightful discussions and valuable suggestions during the last year of my Ph.D. He was always supportive, responsive, and encouraging.

My sincere thanks also go to all Communication, Sensing, and Imaging (CSI) Research Group members who became very close friends and family to me: Dr. Shuja Ansari, Dr. Yusuf Sambo, Dr. Olaoluwa Popoola, Dr. Ahmad Taha, Dr. Qammer Abbasi, Dr. Sajjad Hussain, Dr. Wasim Ahmad, Dr. Kenechi Omeke, Dr Muhammad Ali Jamshed, Liyuan Qi, Daniel Mitchell Waqas Nawaz, Mahmoud Shawky and Syed Basit. I would like to also thank Dr. Guodong Zhao's group: Zhen Meng, Yufeng Diao, Xiangmin Xu, Rongyu Yu, Xinyu Qi, Kan Chen, and Jiaming Yang. Our group meetings and discussions inspired me a lot.

I would also like to thank examiners Prof Chenguang Yang and Dr Oluwakayode Onireti for their insightful discussion and helpful suggestions. Their comments have helped me to a great extent in improving both the quality and the clarity of my thesis.

My grateful thanks also go to Prof. Martin Lavery and Prof. Edward Wasige for their comments and suggestions in my annual progress meetings, which helped me a lot to steer right direction.

My sincere thanks also go to Prof. Enver Ever and Prof. Ali Cevat Taşıran for their everlasting mentorship and encouragement.

My heartfelt gratitude goes to my family. They are always supportive and beside me. This achievement would not be possible without them. My warmest thanks go to Duygu Madenci for her patience and support. She is always supportive, encouraging, and patient.

Chapter 1

Introduction

1.1 Internet of Skills

The internet has been and will be one of the most transformative technologies of the modern age by revolutionizing how we communicate, work, learn, teach, and live [8]. In the beginning, i.e., at the early stage of the internet, it was primarily used for time-sharing of big and powerful computers, enabling multiple users to access a single machine, i.e., a desktop computer, simultaneously from different geographical locations.

With technological advancements, the internet has become mobile and more accessible with a broader range of devices, including cell phones, laptops, and other mobile devices [9]. The mobile internet allows users to access the internet and, in turn, information at anytime, anywhere.

In recent years, the internet has taken another leap forward with the Internet of Things (IoT) [10, 11]. The IoT is defined as the network of devices, including vehicles, home appliances, sensors, actuators, and any item with embedded electronics and software which enables connected *things* to exchange information for different purposes such as monitoring, surveillance, and automation, to name a few.

Nowadays, the internet is taking another revolutionary step forward, which is known as *Internet of Skills* [12, 13]. The Internet of Skills is a concept that refers to a network of sensors, actuators, and machines that enable knowledge, skills, and expertise delivery between people and machines, regardless of their geographical locations. This concept allows an immersive remote operation and access to expertise through virtual and augmented reality, haptic communications, robotics, and other cutting-edge technologies [14, 15]. The Internet of Skills aims to provide a near-real experience of the remote location to enable real-time interactions, allowing people to perform tasks remotely without requiring physical presence as before. There are various applications of the Internet of Skills, including remote surgery and diagnosis in healthcare [16], remote laboratory and training in education [17], remote driving in transportation [18], and advanced manufacturing in Industry 4.0 [19].

An Internet of Skills system, as shown in Fig. 1.1, has three main domains, namely the

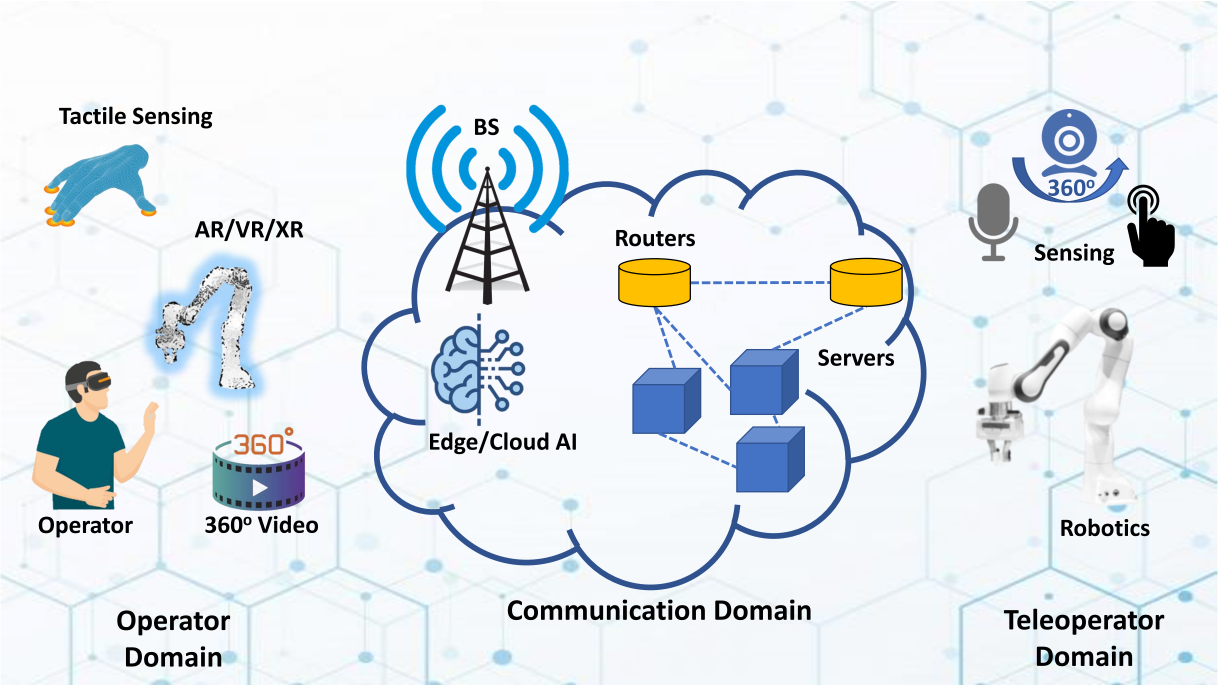


Figure 1.1: Internet of Skills

operator domain, communication domain, and teleoperator domain. In the operator domain, the skills of the operator are sampled by an operator device equipped with the haptic interface. Accurately sampled operator skills are converted to specific commands or instructions and need to be transmitted to the teleoperator at the teleoperator domain via the communication domain with ultra-low latency and ultra-high reliability. The received commands/instructions are executed in the teleoperator domain to reproduce the operator's skills. Concurrently, the real-time status of the remote environment is fed back to the operator domain [13]. For a fully immersive and interactive Internet of Skills system, real-time multimodal sensing, including audio, video, haptic feedback, and real-time control of remote teleoperator, are vital. Specifically, the *end-to-end* (E2E) delay should be around 1 ms, and reliability should be higher than 99.999%. These requirements are very challenging to meet and require interdisciplinary research in robotics, communications, and artificial intelligence (AI).

In this thesis, we conduct an interdisciplinary study in robotics, communications, and AI to fulfill the communication and quality of service (QoS) requirements of Internet of Skills systems. In this section, we first identify key enabling technologies of the Internet of Skills. Then, we investigate technical requirements from a communications perspective. Considering technical requirements, we emphasize the role of fifth-generation cellular communications (5G) by explaining current capabilities and limitations. With the knowledge of current limitations and capabilities, we explain our motivation along with our design perspective. Finally, we state our objectives, tasks required to achieve our objectives, and research contributions.

1.2 Key Enabling Technologies

In this section, we provide key enabling technologies of the Internet of Skills concept to investigate requirements and current capabilities in terms of communications and QoS , which are summarized in Table 1.1.

1.2.1 High Quality 360^{o} Video Streaming

Users need to experience the remote environment as close to actual physical attendance as possible to be fully satisfied with the experience. To achieve this, high-quality 360° video streaming will be used in emerging Internet of Skills applications, which will be observed via headsets to provide a real-time experience of a remote environment, irrespective of distance. For example, we need to consider the human eye as a reference to achieve optimum user experience. The human eye has the capability of focusing on close and far objects, sees under low and high light, and has a very wide view even without moving the head (horizontally 150° and vertically 120°) which requires 720 million pixels of display [20]. That is why the high resolution 360° video streaming is one of the key enabling technologies of future Internet of Skills applications to enable human eye-quality streaming, in turn, real attendance experience in a remote environment. However, this requires very high data rates and ultra-low latency, which will be delivered by 5G and beyond communications.

1.2.2 Augmented Reality (AR), Virtual Reality (VR), and Extended Reality (XR)

The new Internet of Skills paradigm requires remote interactivity to achieve an immersive experience, which would enable users to hear, see, feel, and interact with the remote environment. AR, VR, and XR will be the key enabling technologies of the aforementioned interactive Internet of Skills environment. AR mainly focuses on the real environment by applying virtual information on top of it. In contrast, VR focuses on a virtual environment without reality aspect, creating a fully synthetic virtual environment. On the other hand, XR is considered as an integrated cyber-physical environment which becomes indistinguishable by user. AR, VR, and XR can provide common environments for users, who are physically in different locations, for interaction and collaboration. Users will be able to access, meet, communicate, and interact with the remote cyber-physical environment. The 5G mobile communication will play an enabler role considering stringent communication requirements such as low latency and high data rates.

1.2.3 Tactile Sensing

Current perception capabilities of remote operations such as video and audio alone cannot provide the envisaged experience of the Internet of Skills. Users need to feel and interact with the environment to achieve a complete sense of presence. The tactile sensing paradigm provides real-time sensing of the remote environment to complete the remote attendance puzzle, which includes seeing, hearing, feeling, controlling, and communicating interactively. With recent advances in haptic communications, it is possible to feel the texture of remote surfaces in real-time, which provides a real sense of presence. However, typical haptic sensors are sampled and transmitted with a frequency of more than 1kHz to ensure the stability of the system [21]. In this sense, the system requires stringent latency, reliability, and data rate requirements from the communication point of view, which will be enabled with ultra reliable low latency communications (URLLC) and enhanced mobile broadband (eMBB) features of 5G.

1.2.4 Teleoperation

There needs to be more than just attending remote environments over video and audio for Internet of Skills applications. Users will have control over the remote environment to achieve interactivity which requires teleoperation capability with the integration of robotics by real-time control and automation. However, real-time teleoperation comes with stringent communication requirements to achieve seamless operation. Moreover, real-time control requires dedicated bandwidth, ultra-low latency, and ultra-high reliability, which cannot be supported by pre-5G communication technologies.

1.2.5 Artificial Intelligence (AI) and Machine Learning (ML)

The ubiquitous, flexible, adaptive, and personalized environment will be the core features of the Internet of Skills which is expected to be enabled by the integration of AI and ML on communication networks via mobile edge computing (MEC) technology. However, this approach increases the communication overhead and introduces more stringent communications requirements. Therefore, pre-5G technologies become insufficient to support the new Internet of Skills ecosystem, which will require on-the-fly AI and ML with minimal delay by residing AI/ML algorithms at the edge of the network. All previously described enabling technologies will rely on AI/ML to deliver high-quality user experience in the Internet of Skills. This puts stringent constraints on communication networks, which in turn have to rely on AI/ML to self-optimize and support dynamic user demands. These communication needs include high data rates to enable the fast exchange of a massive amount of data, ultra-low latency to mitigate outdated data problems, and ultra-high reliability to ensure data integrity. Furthermore, wider coverage and ultra connectivity with other devices will further enable and develop future on-the-fly AI and ML.

Table 1.1: Connectivity Requirements of key enabling technologies [1–3]

Enabling Technology	**Data Rate**	**Latency**	**Packet Loss Rate**
Virtual Reality	> 530 Mbps	< 10 ms	$< 10^{-6}$
Remote Interaction	100 - 200 Mbps	< 2 ms	$< 10^{-5}$
High Quality 360° Video Streaming	> 120 Mbps	< 20 ms	$< 10^{-5}$
Tactile Sensing	-	1 ms	10^{-6}

1.3 Technical Requirements

Considering today's technological capabilities in terms of communication, it is possible to discuss some technical requirements to realize the Internet of Skills. We summarize the technical requirements of Internet of Skills systems from a communication perspective considering enabling technologies as shown in Table 1.2.

1.3.1 Ultra-high Reliability

Reliability of the wireless links is one of the most important issues, especially in critical applications, since wireless links are prone to failures and packet losses, directly affecting the system's performance. For applications such as remote surgery and remote diagnosis, a wireless link failure probability of 10^{-5} or less is expected to allow seamless performance over wireless communications. Ultra-reliability is vital since the action itself is life-critical and severe consequences result from failure [3]. In addition, response efficiency of real-life control can be achieved with ultra-reliable communications as well as upgraded hardware to support ultra-reliability [22]. Since new Internet of Skills systems include haptic feedback, reliable communication becomes more crucial. Reliability failures may lead to incorrect diagnoses, treatments, injuries, and even death in life-critical scenarios [23].

1.3.2 Ultra-low Latency

To ensure the stability and performance of the system, another critical requirement is low latency, i.e., low end-to-end delay. In Internet of Skills systems, multimodal feedback requires transmission in every millisecond. Since the current communication technologies (i.e., pre-5G) lag behind the latency requirements of Internet of Skills systems, the forthcoming 5G technology is expected to become the solution for such requirements with shorter transmission time intervals (TTI) [24] and grant-free uplink transmission schemes [25], to name a few. However, it is very challenging to meet the latency requirements of Internet of Skills systems considering

Table 1.2: Technical Requirements of Internet of Skills Applications

	Latency	**Packet Loss Rate**	**Reference**
Scalar Data (heart rate, blood pressure, temperature, respiration rate)	< 250 ms	$< 10^{-3}$	[27]
Multimedia Data (audio, video)	< 150 ms	$< 10^{-3}$	[28], [29], [30]
Haptic Feedback Data	< 1 ms	$< 10^{-5}$	[29], [30], [31]

long-distance communications since propagation delay becomes the dominant factor. Therefore, new design methodologies need to be investigated to address the low-latency challenge.

1.3.3 Security and Privacy

Internet of Skills applications, especially remote surgery and diagnosis applications, require high security and privacy since life-critical actions are performed over wireless links. In this case, the security requirements of the communication systems are as necessary as reliability and low latency. Internet of Skills systems can face some malicious actions such as Denial-of-Service (DoS) attacks by User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) flooding, malicious code injections into the application by a buffer overflow, altering transmitted packets between the operator and teleoperator robot illegitimately, and replaying some legitimate packets [26]. Such malicious actions can lead to severe problems for the patient's health and the system's performance. To overcome security issues, new network security approaches should be investigated by collaborating with developing 5G technology. In addition, the tradeoff between the security of the system and latency requirements should be investigated to ensure that new security schemes meet both security and QoS requirements [21].

1.3.4 Edge Artificial Intelligence

Edge computing techniques are popular for increasing resource utilization efficiency since computing activity is performed close to the site where data are generated (i.e., on the edge). To overcome latency, reliability, or packet loss problems, artificial intelligence (AI) solutions can be deployed to the edge of the network to realize predictive control and caching. For example, in the presence of packet loss, predictive packets can be used from the buffer to keep the performance of the system stable in real-time control systems [3]. It becomes inevitable that scientists will research and discuss new machine learning and AI techniques to use wireless network resources optimally as well as to overcome problems of wireless communications such as packet loss and latency.

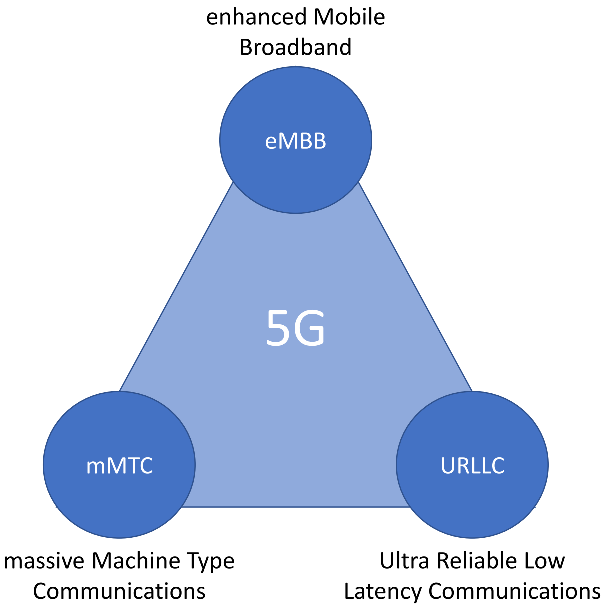

enhanced Mobile Broadband
eMBB
5G
mMTC
URLLC
massive Machine Type Communications
Ultra Reliable Low Latency Communications

Figure 1.2: Three main services of 5G [6, 7].

1.4 Role of 5G

The key technologies described in the previous section will enable new Internet of Skills applications. In this section, we provide connectivity requirements of key enabling technologies by emphasizing the role of 5G in fulfilling these requirements considering three main services of 5G as seen in Fig. 1.2.

1.4.1 Data Rate and Capacity

Internet of Skills applications have high data rate demands due to use cases that involve AR/VR/XR and 360° video streaming. Pre-5G cellular communication systems are incapable of supporting the high data rate requirements, mainly due to limited spectrum. 5G New Radio (5G NR), on the other hand, will provide new spectrum opportunities, especially in millimeter wave (mmWave) frequency bands (24 GHz to 100 GHz) [32]. mmWaves provide high data rates and larger bandwidth, while the coverage is one of the concerns. Therefore, 5G communications will make use of different frequency bands for different use cases. For example, massive Machine Type Communications (mMTC) will use frequency bands below 1GHz, which will provide wide area coverage for relevant applications where coverage is one of the main concerns. On the other hand, eMBB will make use of high frequency bands to provide higher data rates for specific applications such as live 360° video streaming. In addition, it is also possible to meet high data rate and capacity requirements with the help of AI and ML driven digital twins. Digital twin (DT) is the digital representation of a physical system by providing two-way or one-way

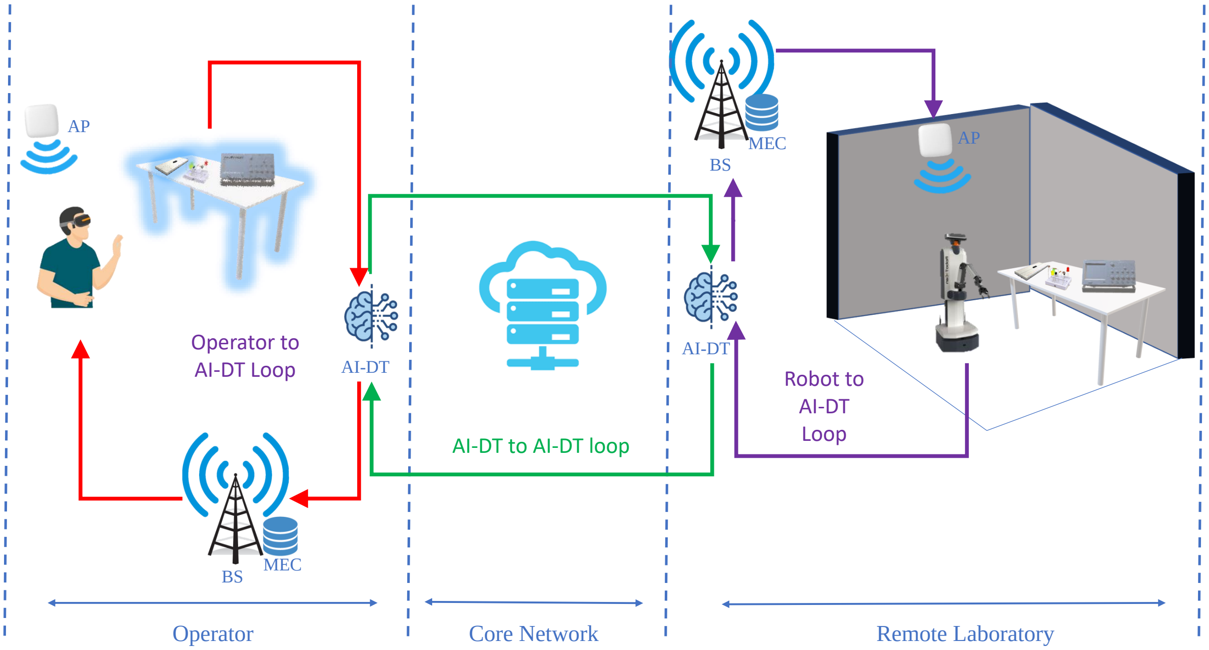


Figure 1.3: Remote laboratory with AI enabled digital twin (AI-DT) has three control loops. The operator to AI-DT and robot to AI-DT control loops provide instant interaction to the operator and the robot. The AI-DT to AI-DT control loop is to synchronize the two ends of the system and to mitigate poor communication in the core network.

communication between a physical and digital system which makes it possible to have real experience with lower requirements over digital model [33]. For example, in remote laboratory and training use cases in education, it is convenient to deploy the DT concept since the task is not life-critical as in healthcare or surveillance for disaster management, where guaranteed network performance is vital. For example, as shown in Fig. 1.3, DT can continue to operate with the help of integrated AI to mitigate mobile network limitations in case of poor network performance in a remote laboratory environment.

1.4.2 Reliability and Latency

Teleoperation, real-time remote control, and tactile sensing require ultra-low latency and ultra-high reliability in mobile communications to enable the seamless operation and high Quality of Experience (QoE). Less than 1 ms end-to-end latency and packet loss rate below 10^{-5} are required to guarantee high quality and smooth remote operation [1], especially in mission-critical scenarios. Smooth remote operation can be achieved with AI/ML advanced mechanisms. AI-aided Communication-Control Co-design (AI-CoCoCo) is one of the solutions to mobile network limitations by relaxing latency and reliability requirements based on dynamic requirements of the control process [34]. For example, some control packets may have outdated information or not carry crucial information for the control task. Therefore, it is possible to relax latency and reliability requirements with AI-CoCoCo considering the value of each packet concern-ing various communication and control related metrics such as latency, reliability, and Age of Information (AoI) instead of providing the same resources to all packets.

1.4.3 Edge/Cloud Computing

With the new data-driven and AI-enabled Internet of Skills environment, mobile communications have become more critical to exchange vast amounts of data. However, the increasing number of connected smart devices and the increased demand for bandwidth bring about new challenges for mobile communication systems. MEC [35] feature of the 5G mobile communications provides cloud storage and computing capabilities available at the edge of the mobile network to enable data processing at close proximity to the user. It enables the exchange of inference instead of raw data, which minimizes the amount of data traffic, and isolates network data from the core network to enable less resource consumption. For example, as shown in Fig. 1.3, MEC-enabled remotely controlled laboratories can benefit from computing resources available on the edge of the network. Instead of transmitting raw sensory information from the operator to the controller over the communication network, MEC enables computation on the operator side and transmission of control commands to the remote robot, which would dramatically increase wireless resource utilization efficiency.

1.4.4 Security and Privacy

The Internet of Skills applications will have access to personal data that could also contain information on users' behaviours and skills, which raises significant concerns about security and privacy. In addition, the increasing number of user devices towards a fully connected Internet of Skills ecosystem amplifies the consequences of security breaches. More secure authentication and encryption schemes are required to guarantee security and privacy in the Internet of Skills. The mutual authentication scheme is one of the solutions to this problem [36]. Both the user equipment and the network perform mutual authentication using Evolved Packet System Authentication and Key Agreement (EPC-AKA). AKA works based on symmetric-key authentication, which is more efficient than public-key based mechanisms. In the Internet of Skills, mutual authentication is vital to ensure the security of facilities as well as the privacy of the user. Both user and remote facility need to authenticate to ensure that the user is authorized to access the provided resources and that resources are legitimate entities to access the information provided by the user. In addition, it is also essential to ensure that authenticated users continue to use the system after the authentication process. Therefore, it becomes necessary to ensure perpetual authentication with the help of AI and ML, where users can be certified continuously according to behaviour or previous user data.

1.5 Motivation

Considering discussions in previous sections, we consider three main communication requirements of Internet of Skills applications in this thesis, namely *ultra-low latency*, *ultra-high reliability*, and *wireless resource utilization efficiency*. Although 5G communications provide state-of-art solutions to achieve ultra-low latency and ultra-high reliability with good resource utilization efficiency, it is challenging to guarantee these requirements, especially in long-distance communications (more than 300 km), considering delays and reliability issues in networking components as well as physical limits of the speed of light. In addition, resource utilization efficiency needs to be further improved to facilitate rapidly increasing number of mobile devices. To fulfill these three requirements, new design methodologies are in urgent need considering both communication and computing systems to satisfy conflicting delay and reliability requirements with improved resource utilization efficiency. One of the main challenges is the design methodology of communication and computing systems for the emerging Internet of Skills paradigm where computing and communication systems are tightly coupled and interact with each other more than before[1]. There are two main design approaches in the existing literature; *i)independent/separate design, ii)joint/co-design* [37]. Table 1.3 gives the comparison of two approaches for networked control systems.

In the independent design approach, computing and communication systems are designed separately. For example, in networked control systems, control systems are designed first. Then, according to the requirements of the control systems in terms of communication, such as data rate, latency, and reliability, the communication system is designed to fulfill the requirements specified [38]. In other words, communication parameters are tuned to fulfill control system requirements. This approach simplifies the system design and creates flexibility so control systems are generally compatible with different communication systems. However, the main problem with the independent design is that control requirements need to be fulfilled by providing required communication resources which may lead to vast use of wireless resources [37], since the independent design uses communication as a tool to fulfill control requirements, yet it does not consider communication resource efficiency as a design parameter. As a result, the created system fulfills the control requirements, but it may not operate optimally in terms of resource (e.g., wireless and energy resources) utilization efficiency.

In the co-design approach, both computing and communication parameters are tuned jointly to achieve better overall performance. Instead of considering two separate sub-systems, the co-design approach considers the system as a whole and optimizes the performance parameters jointly by considering the tradeoff between communication and computing sub-systems [37]. For example, in networked control systems, the system performance relies not only on control

[1] Here, the communication system refers to a wireless communication system in Internet of Skills applications including radio and networking components, whereas computing system refers to robotics, control, and, machine learning systems in Internet of Skills applications

Table 1.3: Independent Design vs Co-Design

	Independent design	**Co-design**
Complexity	Less complex, easier to model	More complex and relatively difficult to model
Overall performance	High sub system performance but low overall performance	High overall performance since parameters are jointly tuned
Wireless resource usage	Low optimization of wireless resources since requirements are considered separately	High optimization of wireless resources since both wireless resources and computing parameters are jointly optimized considering tradeoff between them
Flexibility	More flexible sub systems	Flexibility is less since two sub systems are modeled as one system
Failure detection/diagnosis	Failure detection and diagnosis are relatively easy since the complexity of the system is low	It's challenging to detect and diagnose failures because of system's complexity

performance but also on the performance of the communication system [38]. In such a case, only fulfilling the requirements of the control system does not leverage the performance of the whole system to higher levels. In independent design, it is more flexible and less complex to design the systems, whereas it is more challenging in co-design. However, considering two separate subsystems degrades overall system performance and consumes vast amounts of wireless resources. In short, the main goal of the co-design approach is to consider both computing and communication parameters jointly to design a better system in terms of overall system performance since the system's overall performance depends not only on sub-system performances.

In this thesis, we apply a co-design approach considering computing and communication sub-systems considering the benefits of the co-design approach on overall system performance.

1.6 Objectives

The main objective of this thesis is to establish a communication and computing co-design framework to reduce the *E2E communication latency* and improve the *communication reliability* and *resource utilization efficiency* for Internet of Skills applications. The following research plan and tasks are considered to achieve the main objective.

- Design and implement 5G-enabled Internet of Skills testbed. Using the testbed, conduct real-world communication latency and reliability experiments to investigate current communication capabilities and limitations.

- Identify specific research challenges and formulate the research problem.
- Propose novel design frameworks for the Internet of Skills and validate them on the testbed to achieve the main research objective.

1.7 Research Contributions

Considering the aforementioned research plan, the following research contributions arose from this thesis.

- **5G-enabled Internet of Skills Testbed:** We design and implement the 5G-enabled Internet of Skills testbed (Chapter 3) with haptic feedback. We conduct experiments considering two important use cases in *education* and *healthcare*. In education, we design an Education 4.0 use case that enables students worldwide to connect remote laboratory resources in the University of Glasgow during the Covid-19 lockdown and conduct experiments which provide an immersive learning environment and exchange of skills over communication networks. In healthcare, we consider remote dental inspection use case in which skilled dentists can conduct remote dental inspections from all around the world. Considering both Internet of Skills use cases, we conduct communication latency and reliability experiments to investigate the current capabilities and limitations of 5G communications for the Internet of Skills applications. Experiment results are used to better understand and investigate open research questions to formulate solid research problem considering practical systems.
- **Task-oriented Prediction and Communication Co-Design:** We propose a task-oriented prediction and communication co-design framework (Chapter 4) in the context of Internet of Skills systems, where the utilization efficiency of the communication system is maximized subject to the requirements of different operating tasks. In particular, in the scenario with limited real-world data samples, we generate synthetic data via time-series generative adversarial networks (TimeGAN). This allows us to obtain the relationship between Just Noticeable Difference (JND)[2] threshold and the prediction error probability that is below 10^{-5}. We take 5G NR as an example to demonstrate the proposed framework, where we compare the performance of the systems with and without task-oriented design. We further compare the performance difference of deploying the predictor at the transmitter and receiver sides. Specifically, the main contributions of this chapter are listed below.

[2] In the literature, user experience in haptic communications is characterized by JND threshold, which is defined as the minimum difference between the operator and teleoperator that can be perceived by human users [39].

- We propose a task-oriented prediction and communication co-design framework with a predictor at the receiver side. We derived an upper bound of the overall error probability in the framework by taking packet losses and prediction errors into account. From the upper bound, we reveal the tradeoff between the resource utilization efficiency and the overall reliability.
- To illustrate how to use this framework in practical system design, we take remote robotic control in 5G NR as an example. Then, we formulate an optimization problem to optimize bandwidth allocation and communication data rate subject to constraints on the E2E delay and overall reliability. An optimization algorithm is proposed to find the optimal solution.
- We collect real-world data from a teleoperation prototype. We further use TimeGAN to generate synthetic data for predictor training and testing. With both synthetic data and real-world data, we illustrate the tradeoff between the prediction horizon and prediction error probability, and further evaluate the overall reliability. Our results show that the proposed task-oriented prediction framework can save up to 77.80% bandwidth compared with a benchmark design that is task-agnostic.

- **Intelligent Mode-switching Framework:** We propose an intelligent mode-switching framework (Chapter 5) for long-distance teleoperation in the context of Internet of Skills systems, where the communication load of the communication system is minimized subject to requirements on task completion probability. In particular, we design a general framework by jointly considering the communication and decision-making systems to switch between autonomous and teleoperation modes. User intention recognition is done at the operator side with a Convolutional Neural Networks (CNN)-based classification model. Based on user intentions, a deep reinforcement learning (DRL) agent is trained and deployed at the operator side to switch between the autonomous and teleoperation modes considering communication load and task completion probability. A real-world data set is collected from our Internet of Skills testbed to train and test both user intention recognition and DRL algorithms. We further compare our results with conventional teleoperation. Specifically, the main contributions of this chapter are listed below.

  - We propose an intelligent mode-switching framework for long-distance teleoperation in the context of Internet of Skills systems by jointly considering the communication system and the mode-switching system to switch between autonomous and teleoperation modes. To efficiently use available wireless resources, we formulate an optimization problem to minimize communication load subject to joint task completion probability.
  - To illustrate how to use the proposed framework in practical system design, we implement a CNN-based classification model for user intention recognition. Based on

user intentions, a DRL agent is trained with Deep Q-Learning (DQN) algorithm for decision-making on mode-switching. We collect real-world data from our Internet of Skills testbed to train and test both user intention recognition and DRL algorithms.

- We reveal the tradeoff between communication reliability and task completion probability as well as the tradeoff between operator experience coefficient and task completion probability. Our results show that the proposed intelligent mode-switching framework can achieve up to 50% communication load reduction with similar task completion probability compared to conventional teleoperation without mode-switching.

1.8 Thesis Organisation

In this section, we provide the thesis organisation for the rest of the thesis, which can be given as follows.

- In Chapter 2, we provide background information and discuss the existing literature. Specifically, we first provide fundamental tradeoffs on communication reliability, latency, and data rate considering short packet communications in 5G NR and haptic communications. Then, we provide fundamental machine learning models used in this thesis, explaining the underlying working mechanisms of CNN, Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), reinforcement learning (RL), and Generative Adversarial Networks (GAN). Furthermore, existing literature on co-design approaches for communication packet loss and latency compensation is surveyed and discussed along with research gap analysis to identify the research problem. Similarly, existing literature on co-design approaches for autonomous teleoperation and mode-switching is surveyed and discussed along with a separate research gap analysis section to investigate open research questions.

- In Chapter 3, we provide design and implementation details of the 5G-enabled Internet of Skills testbed. We first explained the operator, teleoperator, and communication domains, along with implementation details. Then, we present two Internet of Skills use cases in healthcare and education. First, we provide details of the remote dental inspection use case in healthcare, which can be used by a dentist to inspect a patient's teeth remotely. We conducted communication reliability and latency measurements and proposed a two-way timeout approach by dropping stale packets for better reliability and latency performance. Our experimental results show that the proposed approach achieves a 65% reduction in latency and around 40% reduction in jitter. Moreover, we provide details of the Education 4.0 use case in education, where we explain design and implementation details by comparing them with existing implementations in the literature. We conduct local (in the lab) and long-distance (within the UK and between UK and China with more than 200 students) latency measurements and discuss current capabilities and limitations.

- In Chapter 4, we propose a task-oriented prediction and communication framework for Internet of Skills applications. We take packet losses in communications and prediction errors in prediction into account and derive an upper bound for the overall error probability. Then, we reveal the tradeoff between the resource utilization efficiency and the overall reliability. We formulate an optimization problem to optimize the resource utilization efficiency subject to constraints on the E2E latency and overall system reliability. Then, we propose an optimization algorithm to find the optimal solution. To evaluate our framework and train our prediction algorithms, we collect real-world data and generate synthetic data using GANs. We show that 77.80% resource reduction can be achieved with the proposed framework compared to a task-agnostic benchmark.

- In Chapter 5, we propose an intelligent mode-switching framework for long-distance teleoperation in the context of Internet of Skills systems. The proposed framework jointly considers communication and mode-switching system parameters and switches between autonomous and teleoperation modes to reduce communication load subject to joint task completion probability. We implement a CNN-based classification algorithm for user intention recognition and DQN based DRL algorithm for decision making, which are trained and tested with real-world data from our Internet of Skills testbed. We reveal the tradeoffs between communication reliability and task completion probability as well as operator experience and task completion probability. We show that 50% communication load reduction can be achieved by switching between autonomous and teleoperation modes compared to conventional teleoperation.

- In Chapter 6, we conclude the thesis with concluding remarks and future research directions.

Chapter 2

Background and Literature Review

In this chapter, we provide background information and review the existing literature. In Section 2.1, we provide fundamental background on short packet communications, explaining basic tradeoffs on reliability, latency and data rate for wireless communications. In Section 2.2, we introduce and discuss haptic communications. In Section 2.3, fundamental machine learning models that are used in communications research are presented. Section 2.4 provides extensive literature review for co-design approaches in existing literature for latency and packet loss compensation. Lastly, Section 2.5 presents existing literature in autonomous teleoperation for Internet of Skills emphasizing co-design approach for better overall performance. In Sections 2.4 and 2.5, research gap analyses are conducted considering existing studies to highlight contributions of this thesis.

2.1 Short Packet Communications in 5G NR

In communications, packet transmission process starts with *channel coding*, in which the information bits that need to be transmitted are mapped into a continuous-time signal. Therefore, the *channel code* can be defined as the mapping between the information bits and the signal. The packet length can be described as $N = WT$ for a continuous-time signal with duration T and bandwidth W [40]. Therefore, short packet communications can be described as the communications with short packet length N. One of the main objectives of the communication system design is to achieve as large as possible rate with a very small error probability, ε. Rate is defined as the rate of information bits to the packet length, i.e., $R = b/N$. The fundamental upper bound on maximum achievable rate without error is firstly introduced by Shannon [41] in 1948 which is known as Shannon capacity. For an Additive White Gaussian Noise (AWGN) channel, Shannon capacity is expressed as

$$R = W\log_2(1+\gamma) \quad \text{(bits/sec)}, \tag{2.1}$$

where γ is the signal-to-noise ratio (SNR). This is an asymptotic result with an assumption of infinitely large packet length, i.e., $N \to \infty$, which is extensively used in communication systems over decades considering no constraints on packet length and communication delay. Over last decade with introduction of 5G, new application areas emerged necessitating low latency communications which can be achieved with shorter packet length. In this case, Shannon capacity is no longer accurate for short packet communications since short packets do not hold the assumption of infinitely large packet length. For example, in Internet of Skills applications, the packet length is very short, e.g., 32 bytes or even shorter according to 3^{rd} Generation Partnership Project (3GPP) specifications [24]. In this case, more refined analysis of maximum achievable rate became a must to analyse communication systems for such applications. However, finding closed-form expression of achievable rate is very challenging and considered as a NP-hard problem [42] in the literature. Fortunately, Polyanskiy et al. [43] provided an approximation to find tight bounds of achievable rate for short packets in 2011. According to [43], the maximal achievable rate for AWGN channels for short packet length can be accurately approximated as

$$R \approx C - \sqrt{\frac{V}{N}} Q^{-1}(\varepsilon) \text{ (bits/s/Hz)}, \tag{2.2}$$

where $C = \log(1+\gamma)$ is the Shannon capacity, $\gamma = \frac{\alpha g P}{N_0 W}$ is the received SNR at base station (BS), α is the large-scale channel gain, g denotes the small-scale channel gain, P denotes the transmit power, N_0 is the single sided noise spectral density, $V = \left[1 - \frac{1}{(1+\gamma)^2}\right]$ is the channel dispersion, $N = WT$ is the packet length (i.e., blocklength), T is the transmission duration, W is the bandwidth, and $Q^{-1}(.)$ is the inverse of the Gaussian Q-function given as $Q(x) \triangleq \int_x^\infty \frac{1}{2\pi} e^{-t^2/2} dt$. It is worthy noting that (2.2) approaches Shannon capacity when $N \to \infty$. From this result, Shannon capacity cannot be achieved in short packet communications and error probability cannot be assumed as zero, i.e., error probability is not negligible. From (2.2), the error probability can be derived as

$$\varepsilon \approx Q\left(\frac{TWC - b + \log(TW)/2}{\sqrt{TWV}}\right), \tag{2.3}$$

which shows the fundamental tradeoffs between transmission duration T, error probability ε, and the required bandwidth W (i.e., spectrum efficiency).

In this thesis, we use fundamental results of maximum achievable rate for short packet communications in our analyses considering the Internet of Skills applications.

2.2 Haptic Communications

In Internet of Skills applications, multi-modal sensory feedback is necessary to enable immersive remote operation capabilities. In recent years, haptic feedback is introduced as a complementary information to audio and video feedback in remote operation applications [44]. Haptic term refers to two types of perception in the current literature, namely kinesthetic and tactile [45]. Kinesthetic perception includes the information of forces, torques, velocity, and position. On the other hand, tactile perception includes the information of object textures, surface frictions, and other modalities sensed by human skin. Haptic information, compared to other sensory information such as audio and video, is more sensitive to communication imperfections such as delay and reliability issues, especially in real time applications [44]. Therefore, haptic feedback enabled real-time applications, e.g., Internet of Skills, comes with very stringent communication requirements. Haptic communications considers these requirements which are well aligned with URLLC communications in 5G [21]. In the literature, the most commonly used performance metrics for haptic communications are latency, jitter, packet loss and data rate [21, 44].

In this thesis, we consider requirements of haptic communications in prototype design and implement haptic feedback enabled teleoperation system. Haptic term refers to force feedback (kinesthetic perception) in our implementation and design.

2.3 Machine Learning for Communications

Machine learning is a subfield of computer science with more than 70 years of history [46]. The main idea is to learn from real-life data to be able to forecast future for more precise decision making without any human intervention, i.e., without programmed instructions. In other words, machine/computer uses real-life data and develops a model which captures the trends and patterns in the given data. Using trained model, the machine becomes capable of carrying tasks without needing to be programmed explicitly. Therefore, the fuel of the machine learning can be considered as data. Advanced algorithms and methods are used to efficiently analyse the data. Over the last decade, data availability increased rapidly with the advancements in sensors (i.e., hardware) and communications (e.g., Internet of Things technology) which enable to monitor and collect data for various phenomenon in a cheap and efficient way, which in turn, produced the fuel for machine learning algorithms. In addition, advancements in computation hardware (e.g., high performance central processing units (CPUs) and graphical processing units (GPUs)) enable processing of large amount of data and development of advance machine learning techniques by providing required computation power for machine learning [47] which allow wide application of machine learning in various areas such as in healthcare for monitoring and early diagnosis [48], in finance for future market forecasts [49], and in education for precision education [50], to name a few.

Machine learning is divided in three categories considering the training approaches, namely supervised learning, unsupervised learning and reinforcement learning [51]. In supervised learning, data samples $(\mathbf{x}_i, y_i), i \in [1,N]$ are used to train machine learning algorithms. Each element $\mathbf{x}_i$ is a feature vector which describes the i-th example in the data sample. For each example, y_i is called label which can be a continuous value for *regression* models and discrete (i.e., category/class) for *classification* models which are two basic types of supervised learning. In regression, given the input feature $\mathbf{x}_i$, the ML model outputs a continuous value y_i. On the other hand, classification models are trained to output a discrete value y_i which belongs to finite set of classes $\{1,2,3,...,C\}$. In unsupervised learning, data samples $(\mathbf{x}_i), i \in [1,N]$ do not have labels where the main aim is to understand the underlying patterns in the data. For example, clustering is one of the well studied area in which unsupervised learning is widely used. Reinforcement learning, on the other hand, aims to find a policy, π, (similar to model in supervised learning) by perceiving state of the environment, s, and taking some action, a, which results in getting reward, r, and change in state of the environment. Machine learns the optimal policy, π^*, which maximize the expected reward, $\mathbb{E}[r]$.

Machine learning is widely applied in communications in various areas such as channel prediction [52, 53], traffic prediction [54, 55], mobility and trajectory prediction [56, 57], QoS and QoE prediction [58, 59], and optimal policy approximation [60, 61].

In this thesis, we use machine learning for trajectory prediction, task classification, policy optimization and synthetic data generation.

For trajectory prediction and task classification, there are three main well established models, namely convolutional neural networks, recurrent neural networks, and long short term memory. Convolutional neural networks [62] is a well known deep learning model for both classification and regression tasks. Recurrent neural networks, on the other hand, are widely used deep learning algorithm, especially for sequential and time varying data [63]. Similarly, long short term memory [64] is a variant of RNN which is designed to overcome vanishing gradient problem and learn long term dependencies in a sequence. It is worth noting that the prediction performance of RNN, LSTM, and CNN depend on the datasets [65–68]. In general, RNN and LSTM outperform CNN in time-series data. But for some datasets, where the time-series data change suddenly, CNN can be better than RNN and LSTM [69]. With these considerations, we design trajectory prediction and task classification algorithms by using CNN, RNN, and LSTM. Then, we choose the best performing model for different tasks.

For synthetic data generation, there are two main approaches, namely autoregressive models, and generative adversarial networks (GANs) [70]. Autoregressive models focus on predicting future elements in a sequence by looking at past elements. They try to factor the distribution of sequences into smaller parts called conditionals. However, they lack true randomness, and one cannot generate entirely new sequences from them without additional information. GANs, on the other hand, are advanced technique to model high dimensional distributions of data [71]. GANs are widely used for synthetic data generation in which real data collection is hard, costly or takes huge amount of time. They are successfully used in image processing and computer vision [72, 73], and text generation [74, 75] as well as data augmentation [76, 77]. However, GANs don't take full advantage of the information in past elements like autoregressive models do, and it can be challenging to capture step-by-step dependencies in the data. Therefore, we apply Time-series GAN [70] to generate synthetic time-series data, which is the special variant of GAN framework for time-series data generation.

In this section, we provide background information and fundamental concepts for convolutional neural networks, long short term memory, and recurrent neural networks as well as basics of reinforcement learning. In addition, we provide basics of Time-series generative adversarial networks which is a GAN for time-series data. These methods are used for trajectory prediction, policy optimization and synthetic data generation in the following chapters.

2.3.1 CNN for Prediction

Convolutional neural networks [62] is a well known deep learning model which consists of mainly convolution layers, pooling layers and fully connected layers as seen in Fig. 2.1. In convolution layers, element-wise convolution operation is applied to the input features to generate the feature representations of inputs (i.e., *feature maps*) with a given kernel ψ. Then, non-linear activation function is applied to obtain the output of the convolution layer. For input $\mathbf{x}_i$ and kernel ψ_i, the output of the convolution layer is a feature matrix $\mathbf{Y}_i$ which can be computed from the following steps for resulting feature at location (k, j) in $\mathbf{Y}_i$,

$$\mathbf{Z}_i^{k,j} = \mathbf{W}_{\psi_i} * \mathbf{x}_i^{k,j} + \mathbf{b}_{\psi_i}, \tag{2.4}$$

$$\mathbf{Y}_i^{k,j} = \Phi(\mathbf{Z}_i^{k,j}), \tag{2.5}$$

where $\mathbf{W}_{\psi_i}$ is the weight matrix, $\mathbf{b}_{\psi_i}$ is the bias matrix, and $\Phi(\cdot)$ is the non-linear activation function, and '$*$' is the convolution operator. After performing convolution operation, the pooling layer is deployed to decrease the number of features, i.e., the resolution of the feature map. One of the well known pooling operation is max-pooling [78]. In max-pooling operation, new feature map is computed by traversing the output of convolution layer and calculating the maximum of each subsection of the convolution output according to the kernel size. After computing feature maps with several convolution and pooling layers, computed features are flatten to be fed to fully connected layer (i.e., dense layer) as seen in the Fig. 2.1 which is trained for estimations.

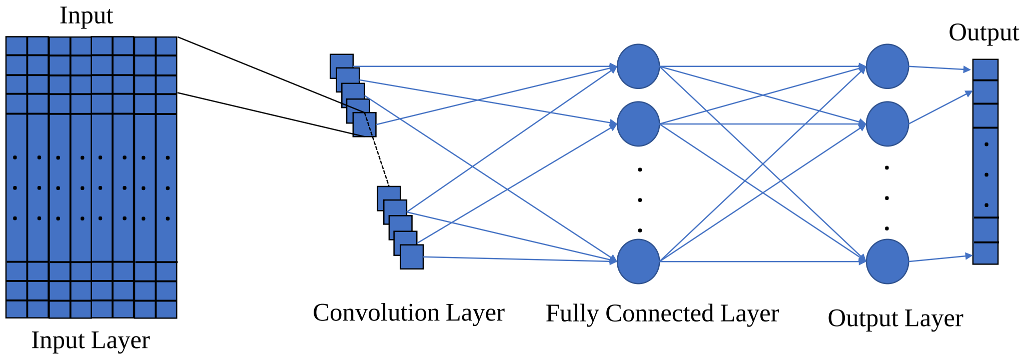


Figure 2.1: Convolutional neural networks

2.3.2 RNN for Prediction

Recurrent neural networks are widely used deep learning algorithm, especially for sequential and time varying data [63]. RNNs process the inputs with arbitrary lengths by maintaining a hidden state, $\mathbf{h}_t$. Hidden state depends on the input $\mathbf{x}_t$ at current time t and the previous hidden state, $\mathbf{h}_{t-1}$. This can be tought as the memory of the network for previous computations. The feedback loop that feeds the hidden state to the network as an input for the next time step is the key feature of the RNNs which allows RNNs to exhibit temporal dependencies between elements of a sequence [79]. RNN model updates the output and hidden states by applying the following steps,

$$\mathbf{o}_t = \sigma(\mathbf{W}_\text{o}[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_\text{o}), \tag{2.6}$$

$$\mathbf{h}_t = \sigma(\mathbf{W}_\text{h}\mathbf{o}_t + \mathbf{b}_\text{h}), \tag{2.7}$$

where $\mathbf{o}_t$ is the output of the RNN cell, $\mathbf{h}_t$ is the new hidden state, $\mathbf{W}_\text{o}$ and $\mathbf{W}_\text{h}$ are weight matrices, $\mathbf{b}_\text{o}$ and $\mathbf{b}_\text{h}$ are bias terms, and $\sigma(\cdot)$ is the activation function. Depending on the specific application requirements, various activation functions can be used such as sigmoid acivation function, hyperbolic tangent (tanh) activation function, rectified linear unit (ReLU) activation function, and exponential linear unit (ELU) activation function. One of the main problem of the RNNs is the vanishing gradient problem which is known as the problem of back propagating errors in time.

2.3.3 LSTM for Prediction

Long short term memory [64] is a variant of RNN which is designed to overcome vanishing gradient problem and learn long term dependencies in a sequence. Each LSTM cell has three

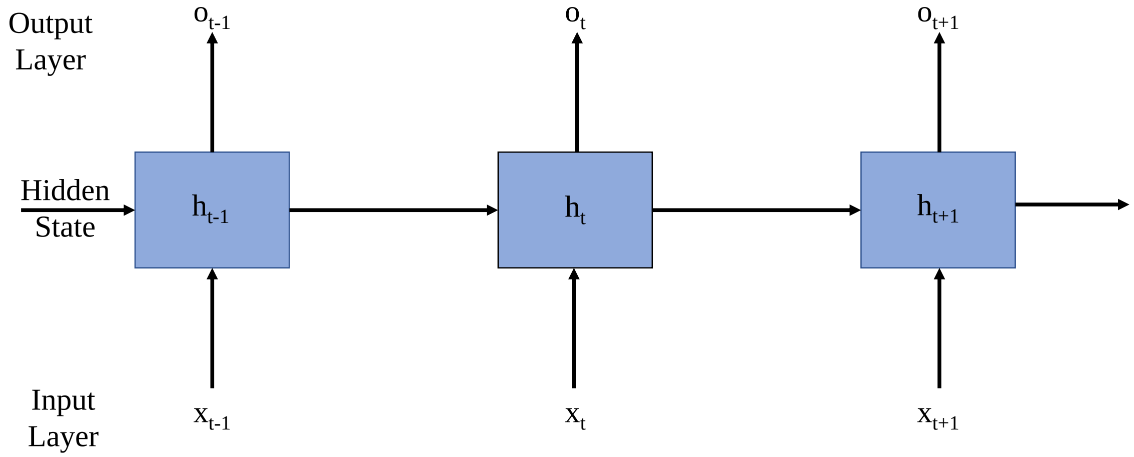


Figure 2.2: Recurrent neural networks

types of gates to control the flow of information, namely the input gate $\mathbf{i}_t$, forget gate $\mathbf{f}_t$, and output gate $\mathbf{o}_t$. In addition, LSTM cell has a cell state which can be thought as a long term memory. The cell state is updated using the information from input and forget gates and output of the timestep is computed by passing updated cell state to the output gate. Each LSTM cell takes three inputs at each time slot: the feature observed in the current slot $\mathbf{x}_t$, the previous LSTM cell state (i.e., long-term memory unit) $\mathbf{L}_{t-1}$, and the previous hidden state (i.e., the short-term memory unit) $\mathbf{h}_{t-1}$. Then, LSTM model updates the output, hidden state, and cell state by applying following steps.

$$\mathbf{f}_t = \sigma(\mathbf{W}_\mathrm{f}[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_\mathrm{f}), \tag{2.8}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_\mathrm{i}[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_\mathrm{i}), \tag{2.9}$$

$$\tilde{\mathbf{L}}_t = \tanh(\mathbf{W}_\mathrm{x}[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_\mathrm{x}), \tag{2.10}$$

$$\mathbf{L}_t = \mathbf{f}_t\mathbf{L}_{t-1} + \mathbf{i}_t\tilde{\mathbf{L}}_t, \tag{2.11}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_\mathrm{o}[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_\mathrm{o}), \tag{2.12}$$

$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{L}_t), \tag{2.13}$$

where $\mathbf{f}_t$ is the forget gate which decides what information will be kept from the previous cell state, $\mathbf{i}_t$ is the input gate which decides what information will be added to the cell state of the network, $\mathbf{o}_t$ is the output gate of the LSTM cell, $\mathbf{h}_t$ is the new hidden state of the network, $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_x$, $\mathbf{W}_o$ are the coefficient matrices, $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_x, \mathbf{b}_o$ are the bias terms, and $\sigma(\cdot)$ is the activation function. As in RNNs, sigmoid, tanh, ReLU and ELU can be used as an activation function depending on the specific application scenario.

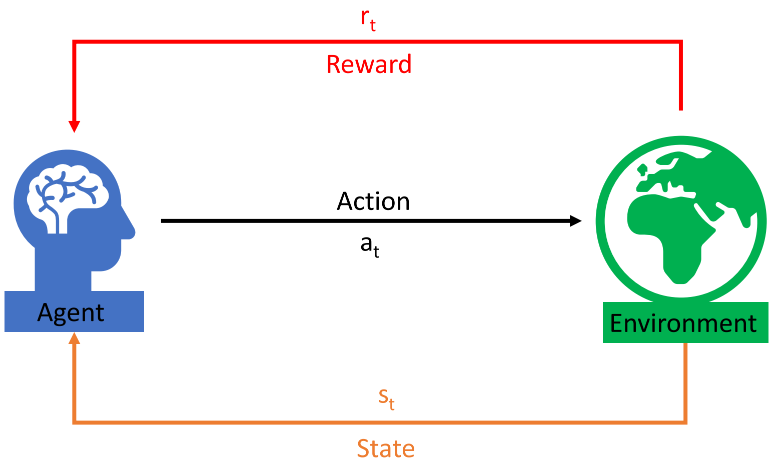


Figure 2.3: Typical reinforcement learning process

2.3.4 Reinforcement Learning

Reinforcement learning [80] is a machine learning technique where an agent learns to make decisions by interacting with a dynamic environment to maximize the reward. The main aim is to learn an optimal policy which can be taught as a mapping between states of the dynamic environment and actions of the agent that maximizes expected reward.

One of the widely used mathematical frameworks to model the reinforcement learning problems is Markov Decision Process (MDP) [81]. A set of states and a set of actions are used to define an MDP along with transition and reward functions. The transition function characterizes the probability of transitions from one state to another for a given action. On the other hand, the reward function characterizes the reward for taking an action in a given state. The Markov property is the fundamental assumption of the MDP. In other words, a process is an MDP if it satisfies the Markov property. Markov property, which is also known as *memoryless* property, states that the probability of future states only depends on the current state and action. In other words, it is not depend on the history of states and actions. Before explaining the training process and some well known reinforcement learning algorithms, we define fundamental concepts.

- **Environment:** Environment is the external world or system. It can be taught as an information source which is used by the agent to learn and make decisions. For example, it can be a physical system that the agent interacts with in an autonomous driving scenario.

- **Agent:** Agent is an entity which learns to make decisions by interacting with the environment. For example, it can be the intelligent system in an autonomous driving scenario which observes the state of the road conditions and take actions accordingly for driving the car.

- **State:** State is the current condition of the environment that the agent interact with. The state can be anything depending on the environment such as sensor readings from autonomous vehicle.

- **Action:** Action is a decision made by the agent considering agent's current policy to affect/change the state of the environment that the agent is in. For example, agent can accelerate, decelerate, or keep stable the velocity of the car depending on the state in an autonomous driving scenario.

- **Reward:** Reward is the numerical feedback which shows how desirable the action taken for a particular state. It can be positive, negative or zero and can be immediate for an action or delayed depending on the specific application. For example, agent can receive no reward during the driving, receives negative rewards for traffic rule violations and receives positive reward once arrive the destination in an autonomous driving scenario.

- **Policy:** Policy is the mapping of the states to the actions which characterizes the behaviours of the agent in a given environment. Using the policy, agent take actions depending on the current state. The policy of the agent can be deterministic or stochastic.

Given the MDP assumption and the definitions, the goal of the reinforcement learning is to find an optimal policy π^* that maximizes expected reward $\mathbb{E}[r]$. During the learning process, the agent interacts with the dynamic environment by observing the current state. According to current state observation, s_t, agent takes action, a_t, considering the current policy π_t. For each action, agent receives a reward, r_t, based on the reward function and considering the current state. After each action, environment changes to the next state, s_{t+1}. Depending on the observed state, action, received reward, and next state, the agent updates its policy in every training step. After applying many training steps, the policy of the agent converges to an optimal policy, π^*, and agent learns to make decisions for the environment that it is trained for. This process is achieved using various reinforcement learning algorithms such as policy optimization algorithms [82], and Q-learning [83].

Q-Learning

Q-learning is a model-free reinforcement learning algorithm, in which agent does not have an explicit model of the environment which characterizes the underlying dynamics of the environment. Instead, agent learns the optimal policy by interacting with the environment and using experiences, without assuming knowledge of the environment [83]. The main goal of the Q-learning is learning to approximate the optimal Q-function, which is known as action-value function. Q-function characterizes the expected cumulative reward for a particular action in a given particular state by following the optimal policy. In Q-learning algorithm, agents learns a Q-function which estimates the expected cumulative reward. To apply Q-learning, systems needs to satisfy Markov property, i.e., future states does not depend on history of states and actions. In each iteration of learning process, Q-function is updated using the Bellmann equation, based on the current observed state, action and the reward which can be expressed as

$$Q(s_t,a_t) := Q(s_t,a_t) + \alpha(r_t + \gamma \max_a(Q(s_{t+1},a)) - Q(s_t,a_t)), \tag{2.14}$$

where s_t is the state at time t, a_t is the action at time t, α is the learning rate, r_t is the expected immediate reward, γ is the discount factor, s_{t+1} is the next state, a is the set of possible actions, and $Q(.,.)$ is the Q-function. This update policy is suitable for reward based framework which agent tries to maximize the reward. In penalty based framework, the $\max(.)$ operator in the above equation should be replaced with $\min(.)$ operator to minimize the penalty, i.e., negative reward. The pseudocode of Q-learning is given in Algorithm 1.

Algorithm 1 Q-Learning

```
Input: Initialize $Q(s,a)$, $s$, $a$, $r$
for episode = 1,... do
    for each iteration = 1,... do
        Choose an action $a_t$ randomly with probability $\varepsilon$, or obtain the action with highest
        Q-value with probability $1-\varepsilon$ ($\varepsilon$-greedy policy)
        Take the action $a_t$
        Compute the reward $r_t$ and next state $s_{t+1}$
        Update Q-function by applying (2.14)
        Update the current state $s_t = s_{t+1}$
    end for
end for
```

2.3.5 Generative Adversarial Networks

Generative adversarial networks (GANs) are advanced technique to model high dimensional distributions of data [71]. GAN structure consists of two neural networks that are trained simultaneously in a competition or game like framework. These neural networks are called generator and discriminator. Intuitively, generator tries to generate synthetic data samples that are not distinguishable from the genuine samples. On the other hand, discriminator tries to distinguish the fake/synthetic samples from the genuine ones. The main aim is to train a generator which can generate synthetic data samples whose distribution is similar to genuine data distribution. GANs are widely used for synthetic data generation in which real data collection is hard, costly or takes huge amount of time. They are successfully used in image processing and computer vision [72, 73], and text generation [74, 75] as well as data augmentation [76, 77].

The main natural question would be why we need to generate synthetic data although we can have access to huge amount of data with the advancements in communications and sensor technologies. There would be several reasons to generate synthetic data such as data augmentation, privacy and security concerns on the available data and data de-biasing and balancing [84]. For example, data augmentation would be required in some cases where real world data available is limited to be used to train machine learning algorithms due to cost and time limitations of data collection process. In addition, the data to be used may contain sensitive information which needs to be protected such as medical records of patients. Furthermore, the data in hand may be imbalanced or biased which may affect the training. In such circumstances, synthetic data generation is a solution which enable us to perform statistical analyses and train machine learning models on synthetic data that are statistically similar to the real data.

In this thesis, we use GANs for synthetic time-series data generation. In the following, we provide fundamental principles of GAN for time series data generation.

To generate synthetic time-series data, one can apply Time-series GAN [70], which is one of the variants of GAN framework for time-series data generation. In TimeGAN framework, there are four neural networks that are trained jointly. They are called embedding neural network, recovery neural network, sequence generator, and sequence discriminator.

To explain how the training works, let us denote the parameters of the embedding network, recovery network, sequence generator, and sequence discriminator as θ_{em}, θ_{re}, θ_{g}, and θ_{d}, respectively. In addition, let's denote the time-series input features as $\mathbf{x}_{t:t'}$ in feature space $\mathscr{X}$, and the latent variable as $\mathbf{v}_{t:t'}$ in latent space $\mathscr{V}$. In the training, embedding and recovery networks are trained to learn mappings from feature space to the latent space. Embedding network takes the sequence of features as an input and outputs the sequence of latent variables which can be given as

$$\mathbf{v}_{t:t'} = \vartheta_{\text{em}}(\mathbf{x}_{t:t'}|\theta_{\text{em}}). \tag{2.15}$$

Similarly, the recovery network do the reconstruction of input features from latent variable by taking latent variable as its input and outputting the sequence of input features as follows

$$\hat{\mathbf{x}}_{t:t'} = \vartheta_{\text{re}}(\mathbf{v}_{t:t'}|\theta_{\text{re}}). \tag{2.16}$$

The generator network, on the other hand, is trained to learn generating synthetic sequences in latent space by taking a sequence of random variables $\mathbf{z}_{t:t'}$ from a distribution which is known. The generator network can be expressed as

$$\hat{\mathbf{v}}_{t:t'} = \vartheta_{\text{g}}(\mathbf{z}_{t:t'}|\theta_{\text{g}}), \tag{2.17}$$

where $\hat{\mathbf{v}}_{i:i'}$ is the generated synthetic sample in latent space. The discriminator network is trained to classify the fake/synthetic and real samples. In other words, it is trained as a classification network in the latent space which tries to distinguish the synthetic sequences from the real sequences. The discriminator network takes a sequence of latent variable (it can be a synthetic sequence or real sequence) and provides a binary output which can be expressed as

$$\tilde{y} = \vartheta_{\mathrm{d}}(\mathbf{v}_{t:t'}|\theta_{\mathrm{d}}), \tag{2.18}$$

where $\tilde{y} = 1$ means that the sequence is classified as a real/genuine sequence and $\tilde{y} = 0$ means that the sequence is classified as a fake/synthetic sequence.

In the training process, the aforementioned four neural networks are jointly trained considering three loss functions: reconstruction loss, $\mathscr{L}_{\mathrm{R}}$, unsupervised loss, $\mathscr{L}_{\mathrm{U}}$, and supervised loss, $\mathscr{L}_{\mathrm{S}}$. Reconstruction loss is used to measure the difference between the reconstructed features, $\hat{\mathbf{x}}_{t:t'}$, and the original features, $\mathbf{x}_{t:t'}$. In other words, this is the loss to measure the performance of embedding and recovery networks which can be expressed as

$$\mathscr{L}_{\mathrm{R}} = \mathbb{E}_{\mathbf{x}_{t:t'}}\left[\sum_i ||\mathbf{x}_i - \hat{\mathbf{x}}_i||_2\right]. \tag{2.19}$$

Unsupervised loss is used to maximize the probability of correct classification for the discriminator network which can be given as

$$\mathscr{L}_{\mathrm{U}} = \mathbb{E}_{\mathbf{x}_{t:t'}}\left[\sum_i \log(\tilde{y}_i)\right] + \mathbb{E}_{\hat{\mathbf{x}}_{t:t'}}\left[\sum_i \log(1-\tilde{y}_i)\right]. \tag{2.20}$$

Supervised loss, on the other hand, is used to measure the discrepancy between real and synthetic data distributions in latent space which can be given as

$$\mathscr{L}_{\mathrm{R}} = \mathbb{E}_{\mathbf{x}_{t:t'}}\left[\sum_i ||p(\mathbf{v}_i) - \hat{p}(\mathbf{v}_i)||_2\right]. \tag{2.21}$$

where $p(\mathbf{v}_i)$ and $\hat{p}(\mathbf{v}_i)$ are real and synthetic data distributions in latent space.

The generator and discriminator networks are trained jointly and iteratively by solving the following optimization problem,

$$\min_{\theta_{\mathrm{em}},\theta_{\mathrm{re}}} \quad \alpha\mathscr{L}_{\mathrm{S}} + \mathscr{L}_{\mathrm{R}}, \tag{2.22}$$

where $\alpha \geq 0$ is a hyperparameter.

The embedding and recovery networks are trained to minimize $\mathscr{L}_{\mathrm{S}}$ and maximize $\mathscr{L}_{\mathrm{U}}$ which yields to following optimization

$$\min_{\theta_g} \quad \eta\mathscr{L}_{\mathrm{S}} + \max_{\theta_d}\mathscr{L}_{\mathrm{U}}, \tag{2.23}$$

where $\eta \geq 0$ is a hyperparameter that balances two losses.

2.4 Co-Design for Latency and Packet Loss Compensation

In this section, we review the existing co-design studies for latency and packet loss compensation, considering communication models, prediction models, performance evaluation metrics, and prediction accuracy performances. Table 2.1 provides a categorization of the reviewed studies.

In the existing literature [4, 85–97], the prediction has been considered as a promising approach to meet stringent latency and reliability requirements by deploying predictors to reduce user experienced delay and improve system reliability.

In [4], prediction and communication co-design is proposed for ultra-reliable low latency communications in 5G. The proposed co-design approach reduces user experienced latency by predicting future packets and transmitting them in advance. Here, user experienced latency is the difference between communication latency and the prediction horizon of the prediction algorithm. With the proposed approach, authors show that achieving zero user-experienced latency is possible. However, it should be noted that the user-experienced latency is not the same as communication latency since the communication latency cannot be zero. In the study, the authors use short packet communications model to characterize the transmission errors in the system. A linear prediction method is used to derive prediction error probability. The probability of less than 10^{-5} JND violation is achieved. User-experienced delay and reliability are used as a performance metric by jointly optimizing prediction horizon, bandwidth allocation, and delay components. It is demonstrated with simulations and experiments that zero user-experienced delay is possible with the joint design of prediction and communication systems. In addition, prediction and communication co-design improves the tradeoff between user-experienced delay and reliability compared to no prediction case.

Tong et al. in [85] proposed communication and control co-design considering packetized predictive control (PPC) for real-time cyber physical systems. The main idea is to predict future packets at the remote controller and send them to the actuator over unreliable wireless links. After receiving the predicted packets, the receiver executes the current command and caches the remaining sequence of commands to be used in case of packet loss. Let us consider that the remote controller transmits the predicted sequence with length K, and then the system can tolerate up to $K-1$ consecutive packet losses. Short packet communications are used to model the wireless channel. Prediction length and wireless resources are jointly optimized to minimize the power consumption. However, the prediction algorithm and prediction error are not specified in the study. Similarly, authors proposed communication control co-design for PPC in [97]. Short packet communications are considered, while an explicit prediction model is not specified. Information freshness is used as a performance metric, while wireless energy efficiency is minimized subject to system outage probability. Results show that the proposed approach achieves a better tradeoff between energy efficiency and information freshness.

The needle insertion use case is considered in 5G remote robotic surgery in [86]. A Gaussian mixture regression model is used to predict haptic feedback to a surgeon in a needle insertion process. The prediction model is deployed at the surgeon (receiver) side to predict haptic feedback in case of packet loss in the communication network. 5G network is considered, although a specific model is not used to characterize the communication system. From the results, the proposed method achieves a prediction time under 1 ms with around 10^{-3} prediction error.

In [87], a two-stage prediction framework is proposed for haptic feedback prediction in a remote driving setting to assist the driver in poor network conditions. The communication system is assumed to be LTE with some time delay, yet no channel model is specified. For the prediction, a single-track model is used, which is a mathematical model to predict the future trajectory of a vehicle. Study shows that the proposed approach provides better telepresence for remote driving.

Girgis et al. in [88] proposed a predictive actuation framework by predicting the missing state at the receiver using the previously received state for wireless networked control systems (NCSs). Communication and control systems are optimized jointly to achieve higher resource efficiency, i.e., increase the number of controlled plants with available wireless resources. Age of information, which is defined as the amount of time elapsed since the generation of the latest received information [98], is used as a metric to minimize along with transmission power. Gaussian process regression (GPR) is used as a prediction algorithm at the remote controller and plant end. The communication system is assumed as analog uncoded communications. From the results, the proposed framework achieves the stable control of more plants with similar wireless resources compared to the benchmark event-triggered scheduling.

In tactile internet, burstiness aware resource allocation is proposed [90]. Packet arrival processes are classified by their burstiness. After traffic state classification, resources are allocated depending on the traffic state. Short packet communications are considered as a communication model, while model-based and data-driven methods are used for classification. The Neyman-Pearson method is used as a model-based method, whereas the k-means unsupervised learning method is employed for a data-driven approach. The required bandwidth is minimized by jointly optimizing communication and prediction parameters subject to joint error constraints where both communication and prediction errors are considered. According to the results, the proposed approach reduces bandwidth requirement by up to 70%.

Considering emerging metaverse applications, sampling, communication, and prediction co-design is proposed in [91] to synchronize the physical world and its digital model on the metaverse. The authors proposed to jointly optimize the sampling rate and prediction horizon to minimize the communication load under different Mean Squared Error (MSE) tracking error constraints. A deep reinforcement algorithm is proposed to adjust the sampling rate and predic-

Table 2.1: Co-Design for Latency and Packet Loss Compensation

Study	Year	Communication Model	Prediction Model	Performance Evaluation	Prediction Accuracy
[4]	2019	Short packet communications	Linear prediction model	User experienced delay and reliability	JND violation probability $< 10^{-5}$
[85]	2018	Short packet communications	Not specified	Energy consumption	Not specified
[86]	2020	Not specified	Gaussian mixture model	Prediction time and accuracy	$<$ 1ms and $< 10^{-3}$
[87]	2016	Not specified	Single track vehicle model	Steering wheel reversal rate	Not specified
[89], [88]	2020, 2021	Analog uncoded communications	Gaussian process regression	AoI & resource efficiency	Not specified
[90]	2018	Short packet communications	Neyman-Pearson method and k-means method	Resource efficiency	10^{-2}
[91]	2022	Short packet communications	MLP and DRL	Communication Load	MSE $< 0.002^o$ for tracking error
[92]	2022	Not specified	RNN	Prediction performance	2.75 mm head motion prediction error for 100 ms latency
[93]	2020	Not specified	LSTM and MLP	Prediction performance	around 4% pixel mismatch
[94]	2021	Short packet communications	Linear prediction method and RL	AoI	Not specified
[95], [96]	2022, 2022	Short packet communications	Linear state transition model	AoI	Not specified
[97]	2019	Short packet communications	Not specified	Information freshness & energy efficiency	Not specified

tion horizon with the inclusion of expert knowledge. Short packet communications are considered in communication, while Multi Layer Perceptron (MLP) is deployed on the receiver side for prediction. The simulation and experimental results show that the proposed approach reduces communication load with a reliability guarantee.

In [92], the head motion prediction method is proposed to compensate for latency in augmented reality glasses. The proposed solution considers an RNN model for predicting sequential hand movement trajectory. Communication latency is assumed to be constant, and prediction performance is presented considering the given communication latency. The study shows that the proposed prediction method can reduce the effects of communication latency by predicting head movement compared to the baseline without prediction. Similarly in [93], the motion prediction method is proposed for VR applications considering URLLC in 5G. The proposed approach conducts motion prediction and pre-rendering on edge to compensate for rendering latency. Multi-task LSTM and MLP models are used for motion prediction using real-world head and body motion datasets. The proposed framework achieves better prediction performance compared to baseline methods.

AoI oriented prediction framework for intelligent transportation systems is proposed in [94]. The effect of the prediction and prediction horizon is investigated for long-distance communications on the AoI. The general linear prediction method is used for prediction, and short packet communications are considered. Predicted packets are transmitted to the receiver in advance to compensate for the latency. The expression for average AoI is derived, and the relationship between prediction horizon and average AoI is investigated. According to the results, prediction is more suitable for short-distance communications regarding AoI performance. Therefore, an RL algorithm is trained to make a decision on when to predict or not predict. Results show that dynamic switching between predict and not predict modes provides better AoI performance. Similarly, the status prediction and data aggregation approach is proposed to improve AoI performance in short packet communication systems in [95]. Two transmission schemes are proposed considering prediction errors. In *predict-compare* transmission scheme, the predicted status update is compared with the actual update, and the transmission is terminated in case of prediction errors. In *predict-aggregate-compare* transmission scheme, two status updates are predicted with different prediction horizons and aggregated considering their time correlations. The prediction model utilizes a general linear state transition model. Proposed schemes provide 64% average AoI reduction compared to the baseline without prediction.

2.4.1 Research Gap Analysis

Deployment Strategy

Considering existing studies summarized in this section, the predictor is either deployed at the transmitter [4, 85, 87, 89] or at the receiver [86, 88]. Both deployment strategies have advantages and disadvantages. If the predictor is deployed at the transmitter, the advantage is that the historical information used in the prediction algorithm is accurate. The disadvantage is that either a prediction error or a packet loss in communication may result in a JND violation. If the predictor is deployed at the receiver, it can adjust the prediction horizon according to the communication delays of different packets. If the communication delay is satisfactory, there is no need to make any prediction. In this way, a JND violation happens when both the communication and the prediction fail. The disadvantage of this framework is that the historical information used in the prediction algorithm may not be accurate because some packets are lost or severely delayed. Nevertheless, the effects of deployment strategy have not been investigated in the existing literature and deserve further analyses.

Task Awareness

Existing studies optimized communication systems with given QoS requirements (such as communication delay and decoding error probability) [4, 85, 88, 89]. In haptic communications, the overall reliability is characterized by the probability that the difference between the operator and the teleoperator is larger than a threshold referred to as Just Noticable Difference (JND) [39]. The relationship between QoS and JND of different tasks in haptic communications remains unclear. If the communication system is designed to meet the given QoS requirements without considering the JND thresholds of different tasks, the communication resource utilization efficiency can be extremely low. For example, the JND threshold in remote surgery is much smaller than that in remote education. Even if the required JND violation probability is the same, the required prediction accuracy in these two systems is very different. To improve communication resource utilization efficiency, we should consider diverse JND thresholds of haptic tasks when we design the prediction and communication systems.

Prediction Reliability

In the existing literature, another issue is the evaluation of reliability. Unlike communication systems that are built upon fundamental theories, most prediction algorithms are data-driven. Since the JND violation probability should be lower than 10^{-5}, it takes a very long time to evaluate the probability of the rare event from real-world data. If we use model-based prediction, it is possible to derive the JND violation probability, but the mismatch between the theoretical models and practical systems is inevitable. Therefore, we need to develop innovative methods for estimating the reliability of prediction algorithms.

Considering research gap analysis conducted, we observe that existing studies did not investigate the effects of predictor deployment strategies, diverse task requirements, and prediction algorithm reliability in system design. In chapter 4, we propose a task-oriented prediction and communication co-design framework to address open research questions in the existing literature.

2.5 Co-Design for Autonomous Teleoperation

In this section, we review the existing literature on autonomous teleoperation considering user intention recognition techniques, task performance metrics, and decision-making algorithms, which are summarised in Table 2.2.

Autonomous teleoperation is extensively studied in the existing literature [99–111]. The main idea of autonomous teleoperation is predicting user intention and performing some parts of the task autonomously to decrease the demand on the operator and increase the task success rate. Hence, existing works can be categorized based on their user intention recognition techniques, task performance, and decision-making algorithms. User intention recognition is done either by model based methods [99, 103, 111], data-driven methods [104–110] or combination of both [100]. User intention recognition accuracy varies between 20% and 95%. On the other hand, some studies do not consider accuracy. Performance metrics change from task to task. However, task success rate and task completion time are the most popular objective measures in addition to subjective measures such as operators' mental and physical demands.

In [99], a manipulation assistance system is proposed for teleoperation to minimize the effects of communication delays, limited bandwidth, and environmental effects. The proposed system captures user intention and assists the user in two modes, namely shared control and autonomous control. In shared control, the system corrects the movement of the remote teleoperator while tasks are executed autonomously in autonomous mode. Task parameterized hidden semi-markov model is used to learn the manipulation tasks from operator demonstrations. The proposed solution provides improvements in task-execution time.

In another study [100], authors propose Stochastic Assistive Teleoperation System to increase the operator efficiency and throughput. In the proposed approach, the system provides visual scene estimations of applicable actions to allow the operator to choose from possible actions. A combination of the Markov process and RNN model is considered to generate applicable action sequences. The proposed system increases operator throughput by 92% by performing 93% of the task autonomously. Similarly, Gao et al. [103] propose an assistive teleoperation approach for search and rescue robot teleoperation. User intention is recognized with a Gaussian mixture regression model with a recursive Bayesian filter. Based on user intention recognition, the system assists the user by blending user input and inference from user intention recognition.

Policy blending approach is proposed for assistive teleoperation for manipulation tasks in [101]. In policy blending, user input and robot policy are blended with a state dependent function to perform manipulation tasks. Full autonomy and direct control are two extremes depending on the blending function parameters. By adjusting the parameters of the blending function, the mode of operation can be adjusted to achieve good performance in terms of task execution time.

Zein et al. [104] proposed autocomplete teleoperation approach for unmanned aerial vehicles (UAV) teleoperation. A machine learning model is utilized at the operator end to recognize user intention as motion primitives such as lines, arcs, and full circles. If the recognized motion primitive is correct, the operator commands to switch to autonomous mode to autocomplete the task. Support Vector Machine (SVM) is used for user intention recognition which achieves around 80% accuracy. Compared to conventional teleoperation, the proposed approach reduces the average distance covered and the average time spent to finish a track with a UAV. In [105], authors improve the autocomplete teleoperation approach [104] by introducing mixed reality. In addition, a deep learning model, which consists of CNN and RNN layers, is considered instead of an SVM classifier. Prediction accuracy is improved by 10% with a reduction in average distance covered and average time spent to finish a track by teleoperating a UAV.

Table 2.2: Autonomous Teleoperation Literature

Study	Year	Idea	Performance Metric	Prediction Model	Prediction Accuracy	Decision Making	Comm. Persp.
[99]	2017	Shared /Autonomous control modes	Task performance, robustness, task execution time	Task-parameterized hidden semi Markov model	x	User	x
[100]	2022	Assistive teleoperation system	Command automation rate, robustness, scalability and runtime performance	Combination of Markov process and RNN	Not specified	User	x
[101]	2012	Assistive teleoperation	Execution time	Policy blending	x	User	x
[103]	2014	Assistive teleoperation	Estimation accuracy	Gaussian mixture regression	80-97%	User	x
[104, 105]	2020, 2021	Autocomplete teleoperation	Average distance covered, average time required	SVM and Conv-LSTM	79%, 88%	User	x
[106]	2021	Semi-autonomous teleoperation	Task success under delay, recognition accuracy	LSTM	95%	User	x
[107, 108]	2019, 2021	Transfer learning for task recognition	Classification accuracy over observation length	LSTM, CNN and SVM	95%, 85%	-	x
[109, 110]	2021, 2022	Personalized Autocomplete	Number of mistakes over time	CNN and GRU	80% to 90%	User	x

Semi-autonomous teleoperation is proposed for remote surgery under communication delays in [106]. Surgeon intention is recognized using a deep learning model which utilizes an LSTM model. The deep learning model recognizes the surgeon's intention and sends high-level commands to the robot, which executes the received commands. The proposed approach achieves around 95% surgeon intention recognition accuracy and around 80% task success rate.

In [107], a transfer learning approach is proposed for surgeon intention recognition using kinematics and images to transfer knowledge to the real robot acquired from a surgical simulator. SVM and Random forest models are used, which achieve around 95% recognition accuracy with up to 30% transfer accuracy gain. Similarly, authors propose a transfer learning approach for surgeon intention recognition for remote surgery teleoperation scenario in [108]. Surgeon intention recognition is performed using an LSTM model. Prediction accuracy of around 85% is achieved. It is also shown that the transfer learning approach requires 40% less training data than no-transfer learning.

In [109], a personalized version of the autocomplete teleoperation [104] is proposed for UAV teleoperation. Personalization is achieved by transfer learning using user input and partial feedback (user's feedback to switch or not switch to autocomplete mode) to minimize the number of mistakes over time. A combination of CNN and gated recurrent unit (GRU) models is used for user intention recognition, which reduces the number of mistakes by 30%. Authors improve their approach in [110] by introducing incremental learning to adapt the previously trained user intention recognition model to the new user. From the input of the new user, an exemplar set is created, which is used to update the user intention recognition model, which enables adaptation of new users and improves the previous model [109] by 20%.

2.5.1 Research Gap Analysis

Operator Overhead

Considering the existing studies, decision-making for mode-switching is generally assumed to be done by the operator [99–111], which brings an extra degrees of freedom (DoF) to be controlled by the operator. The operator needs to make concurrent decisions on mode-switching alongside the task that he/she is performing, which may prevent the operator from focusing on the task. This is almost impossible in life-critical tasks like remote surgery, which requires full attention to the task. Although task auto-completion can enhance user experience and task success, the mental and physical demand introduced by mode-switching decisions is not taken into account in the existing literature. Hence, it is an open problem how mode-switching can be performed autonomously and seamlessly.

Communication Perspective

On the other hand, the communication perspective is not investigated, although some studies discuss the performance under communication delays [99, 106]. Making strong assumptions on communication delay and reliability by assuming constant delay or zero packet loss is unrealistic and cannot be applied to practical systems. Moreover, the task success rate cannot be captured fully by only considering user intention recognition accuracy without considering communication imperfections. Therefore, communication errors and their impact on task success rates need to be considered in system design. Furthermore, communication errors can lead to task failure or even system failure, especially in critical applications. Hence, autonomous teleoperation systems need to be designed jointly by considering tight interactions between communication and computing (e.g., prediction and control) systems. Therefore, how a communication system can be designed to accommodate autonomous teleoperation in an efficient manner is still an open research question that needs to be addressed.

Considering research gap analysis conducted, existing studies in the literature operator overhead is not taken into account in autonomous teleoperation systems which may lead to lower task completion probability. In addition, effects of communication imperfections on task completion probability are not investigated. In chapter 5, we propose an intelligent mode-switching framework for autonomous teleoperation systems to address open questions in the existing literature.

Chapter 3

Design and Implementation of 5G-based Teleoperation Prototype

3.1 Overview

We design and implement a 5G-enabled teleoperation prototype with haptic feedback[1] for the emerging Internet of Skills applications. At the operator domain, 3D systems haptic device is deployed as a haptic controller, capable of 6-DoF positional sensing and 3-DoF force feedback. The human operator uses the haptic device to control the remote robotic arm on the teleoperator domain over the communication domain. We deployed Franka Emika Panda robotic arm as a teleoperator, a 7-DoF serial manipulator with 1kHz control and sensor sampling capabilities.

In this chapter, we explain the flow of information in each domain and describe the prototype setup. Then, we conduct experiments considering two important use cases in education and healthcare. In education, we design an Education 4.0 testbed, which enables students worldwide to connect remote laboratory resources and conduct experiments that provide an immersive learning environment and exchange of skills over communication networks. In healthcare, we consider remote dental inspection use case in which skilled dentists can conduct remote dental inspections worldwide. Considering both Internet of Skills use cases, we conduct communication latency and reliability experiments to understand the current capabilities and limitations of 5G communications for the Internet of Skills applications. We proposed a two-way timeout approach by dropping stale packets for better reliability and latency performance. We consider a joint Cost Function (CF), defined as the cost of successfully transmitting a packet. Our experimental results show that the proposed approach achieves a 65% reduction in latency and around 40% reduction in jitter. Experiment results are used to understand better and investigate open research questions to formulate solid research problems considering practical systems.

[1] The demonstration video: https://youtu.be/c3onK5Vh6QE

3.2 Operator Domain

A local computer at the operator domain samples 3D positions, $\mathbf{p} = \{p_x, p_y, p_z\}$ of the end effector of the haptic device at the haptic device working space. As seen in Fig. 3.1, the robot and haptic device coordinate systems do not match. To avoid the mismatch, we investigate their local coordinate systems and match their movement axes where the Y-axis of the haptic device corresponds to the X-axis of the robotic arm (i.e., towards up), the Z-axis of the haptic device corresponds to the Y-axis of the robotic arm (i.e., towards to user), and X-axis of the haptic device corresponds to Z-axis of the robotic arm (i.e., towards right). Then, positions are scaled up to match the workspace of the robotic arm and haptic device since one unit of movement in the haptic workspace does not correspond to one unit of movement in robot space, and mapping is required. The mapping between haptic device workspace and robotic arm workspace can be given as

$$\mathbf{p}^{'} = \begin{cases} p_x^{'} = (p_y/d) + a \\ p_y^{'} = (p_z/d) + b \\ p_z^{'} = (p_x/d) + c \end{cases} \tag{3.1}$$

where $\mathbf{p}^{'} = \{p_x^{'}, p_y^{'}, p_z^{'}\}$ is the 3D position of the end effector of the robotic arm. The constants a, b, and c are the offsets between haptic device end effector and robotic arm end effector in X, Y, and Z axes. The constant d is the scaling factor between haptic device work space and robotic arm work space. Computed position values, $\mathbf{p}^{'}$, with fixed quaternion rotations (since we only consider position control) are then fed into the inverse kinematics (IK) solver of the robotic arm, which is based on the Denavit–Hartenberg [112] parameters of the Franka Emika Panda robotic arm [113] which can be given as follows.

$$\theta = f_{ik}\left(\mathbf{p}^{'}, \mathbf{q}\right) \tag{3.2}$$

where $\theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7\}$ are computed joint angles, $\mathbf{q} = \{q_w, q_x, q_y, q_z\}$ are fixed quaternion rotations, and $f_{ik}(.)$ is the inverse kinematics solver function. The same computation is performed in each sampling cycle, and computed joint angles are transmitted to the teleoperator.

3.3 Communication Domain

In the communication domain, we deploy a UDP server at 5G base station MEC unit. It is worth to note that UDP is used to guarantee the required control frequency since it is more suitable than TCP for real-time applications considering latency requirements [114]. The haptic controller and

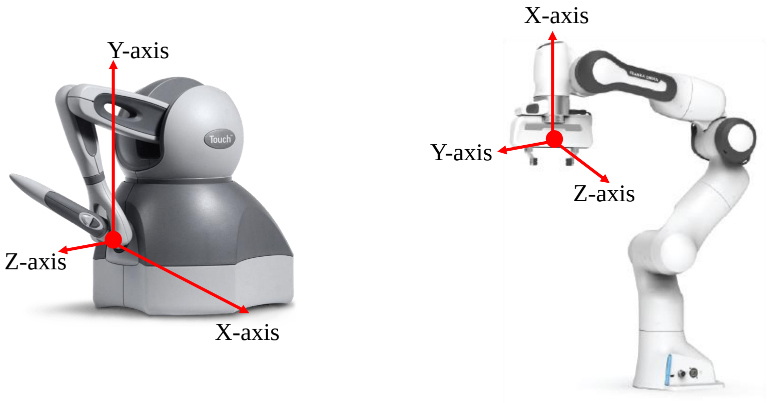


Figure 3.1: Coordinate systems of Franka Emika Panda robotic arm and 3D systems haptic device.

the robotic arm computers are equipped with 5G dongles to access the local BS. At the UDP server, we utilize two separate ports to communicate with two clients (i.e., the haptic device computer and the robotic arm computer). In addition, we deploy two buffers which are used to keep the latest received packets from the robot and haptic device. The haptic device computer transmits the latest computed joint angles to the server, and then the server writes them into an angle buffer, as shown in Fig. 3.2. The latest joint angles are transmitted to the robotic arm whenever a new packet request is received. New packets are requested by transmitting the latest sample to the server. In other words, each client transmits the latest sample, then the server receives the latest sample and updates the corresponding buffer. In return, the server transmits the content of the buffer to the client. If the client is a haptic device, the angle buffer is updated, and the content of the force buffer is returned. If the client is a robotic arm, the force buffer is updated, and the content of the angle buffer is returned.

3.4 Teleoperator Domain

Franka Emika Panda robotic arm and a local computer reside in the teleoperator domain. Received joint angles from the server are written into a buffer for execution. The main reason to use the buffer is that the frequency of the real-time control loop is 1kHz, which means that the control loop needs to be fed with new joint angles in every 1 ms. This requires new packet arrival in every 1 ms, which is not possible in physical implementation because of the communication and computation latencies. Therefore, we utilize a buffer at the robot end, which act as an information source for the control loop and enable us to run the control loop with a frequency of 1kHz.

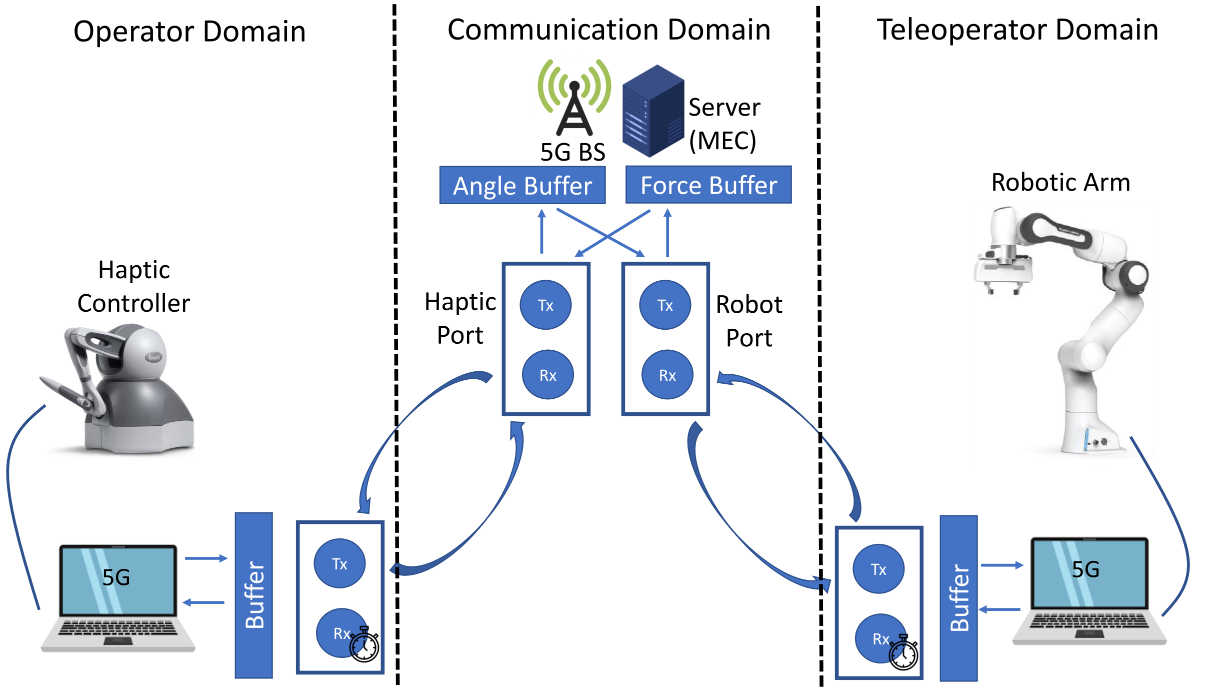


Figure 3.2: 5G-enabled teleoperation system with haptic feedback

On the other hand, we update the buffer with the latest received joint angles whenever a new packet arrives from the server. We employ a proportional integral derivative (PID) control algorithm on a local computer which computes joint angular velocity commands for received joint angles considering the difference between received joint angles and the current state of the robotic arm. Computed commands are executed using provided functions of *libfranka*, which is a C++ implementation of Franka Control Interface (FCI) [113]. This process is repeated in each control loop to enable real-time control. In addition, the robotic arm's built-in sensors are used to sample external forces applied to the robot, which are transmitted as feedback to the server, which enables us to control the robot in real-time with haptic feedback (i.e., sense of touch).

3.5 Remote Dental Inspection Use Case: Two-way Timeout Approach

Some serious studies have been conducted in teleoperation for life-critical actions in healthcare. For example, the first known remote surgery was demonstrated in 2001 with constant latency of 155 ms and zero packet loss [115]. The patient was in Strasbourg, France, and the surgeon was in New York. Another example was demonstrated in 2016 by Ericsson and King's College London using 5G technology. The most recent example of remote surgery was carried out in China in 2019. The patient was 3000 km away from the surgeon, and brain surgery was performed using 5G technology with the collaboration of Chinese PLA General Hospital (PLAGH), China Mobile, and Huawei 5G technology [116].

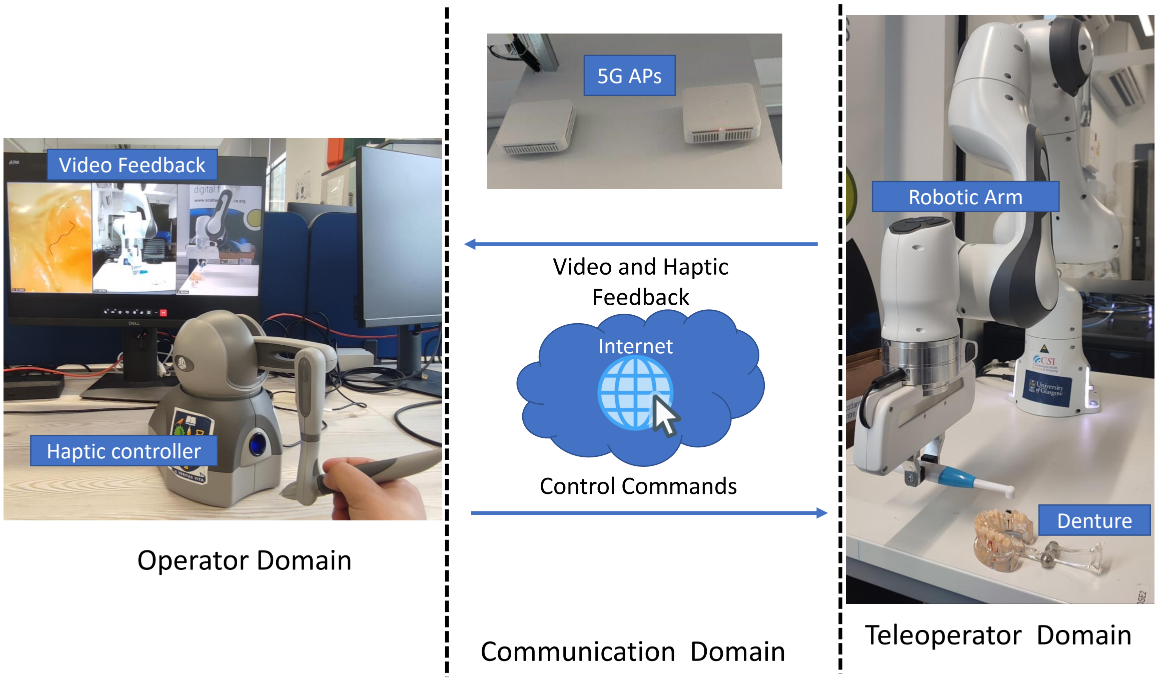


Figure 3.3: Remote dental inspection prototype

In this section, we describe the remote dental inspection prototype[2] system and propose the two-way timeout approach for communication latency reduction. Remote dental inspection (see Fig. 3.3) ensures that everyone has equal access to appropriate dental care by removing geographical, personal, and physical barriers. Our remote dental inspection prototype allows patients to get remote specialist care while also allowing specialists to undertake real-time examinations remotely. This type of application decreases the burden of treatment by reducing travel time and improving access to dental services. Furthermore, it saves the dentist's time and decreases carbon emissions.

3.5.1 Two-way Timeout Approach

As seen in Fig. 3.3, our prototype consists of three main domains. At the operator domain, a controller equipped with a haptic interface is used by a human operator to control a robotic arm at the teleoperator domain. The haptic controller samples the operator's inputs (i.e., control commands) and transmits them via the communication domain to accomplish a task. In the communication domain, a server is utilized to exchange packets between the operator and teleoperator domains. In the teleoperator domain, a remote robotic arm receives control commands and provides force feedback over the communication domain. Generally, UDP is used to guarantee the required control frequency since it is more suitable than TCP for real-time applications [114].

In conventional UDP, clients request packets and wait for the response from the server. Wait-

[2] Demonstration video: https://youtu.be/afJFUwUW6Dg

ing time is related to the latency and packet loss experienced in the communication domain. Since UDP is unreliable, packet losses are to be expected, which may result in communication deadlock. Deadlock can be described as the event that both the client and the server are in the receiving (Rx) state. In most cases, this is caused by packet loss in the communication domain. For example, a client transmits a packet to the server and waits for the response in the Rx state. If the transmitted packet drops, the server will also be in the Rx state, resulting in a packet exchange deadlock. In the sequel, we propose a technique to minimize communication latency and the chance of a deadlock. As shown in Fig. 3.2, two clients, a haptic device and a robotic arm, communicate with the UDP server to exchange control commands and haptic feedback, respectively. In this scenario, each client sends the most recent sample to the server and waits for the most recent feedback or control command. As a result, clients can be in either the Rx or transmitting (Tx) state. We use a timeout technique at each client end (i.e., two-way timeout) to limit the waiting time in the Rx state and to mitigate the event of a deadlock. In other words, the client transmits the most recent sample and waits until the timeout threshold, τ, is reached. If no packet is received before then, it switches to the Tx state to transmit a new sample. As a result, the client is not required to wait in the Rx state for any longer than the timeout threshold τ. Please note that the timeout threshold, τ, must be adjusted according to application requirements. For example, if the required latency for an application is 10 ms, then waiting more than 10 ms is a waste of time because any packet received after the threshold will be stale. In other words, if a packet is not received before the timeout, the receiver assumes it is lost. Fresh information can thus be retrieved without wasting time waiting for stale information.

Let us denote the communication latency from the haptic device to the server as d_{hs} and the communication latency from the server to the haptic device as d_{sh}. Similarly, denote the communication latency from the robotic arm to the server and from the server to the robotic arm as d_{rs} and d_{sr}, respectively. Then, E2E communication latency (i.e., round trip latency) can be given as

$$d_{e2e} = d_{hs} + d_{sr} + d_{rs} + d_{sh}. \tag{3.3}$$

The proposed timeout approach limits d_{sr}, and d_{sh} by the timeout threshold τ, i.e., $d_{sr} \leq \tau$, and $d_{sh} \leq \tau$. It is worth noting that τ is a task-dependent parameter which needs to be determined as a design parameter for each specific task. For time critical tasks, τ is expected to be smaller compared to tasks which can tolerate larger delay. In this case, the E2E communication latency becomes

$$d_{e2e} \leq d_{hs} + d_{rs} + 2\tau. \tag{3.4}$$

While the timeout strategy ensures that there is no deadlock in the communication channel, it also means that packets may be discarded or dropped if they do not arrive before the timeout expires. This implies that the proposed approach has a lower level of reliability. To account for this, we denote the reliability as γ and compute the reliability using Packet Delivery Ratio (PDR), which can be expressed as

$$\gamma = R/T, \tag{3.5}$$

where R is the total number of packets received by the receiver and T is the total number of packets transmitted by the transmitter. To capture the overall performance, we consider a joint Cost Function (CF), defined as the cost of successfully transmitting a packet, which is inspired by retransmission scheme in URLLC to improve the reliability [4, 24]. For given E2E latency d_{e2e}, and reliability γ, CF can be defined as

$$CF = \frac{1}{\gamma} d_{e2e} \,(ms), \tag{3.6}$$

where $\frac{1}{\gamma}$ is the number of repetitions required to transmit a packet with probability 1, and d_{e2e} is the time required to transmit a packet.

3.5.2 Latency Measurement Setup

One of the main challenges in latency measurements is the unsynchronized end devices. Since the E2E latency is typically in the range of a few milliseconds (e.g., 1-10 milliseconds), ensuring that the end devices are synchronized to the same clock is critical for achieving reliable and accurate results. One method for synchronizing end devices is to use *Network Time Protocol (NTP)* servers, in which each device synchronizes its local clock with the same NTP server, allowing time synchronization. However, even this solution has a synchronization error of tens to hundreds of milliseconds (ms) depending on the NTP server load and other communication channel characteristics [117], which is insufficient for our experiments. To overcome this difficulty, we collect exchanged packets with timestamps at the server, allowing us to perform the measurements on the same machine and eliminate the time synchronization issue. Let us denote t_c^i as the time the i-th packet received by the server from the client c. Then, the E2E latency between the server and the client c for the i-th packet becomes

$$d_{e2e,c}(i) = t_c^{(i)} - t_c^{(i-1)}, \tag{3.7}$$

where $d_{e2e,c}$ is the time difference between two consecutive packets, which measures the round trip time required from server to client and client to server. Then, the mean E2E latency for

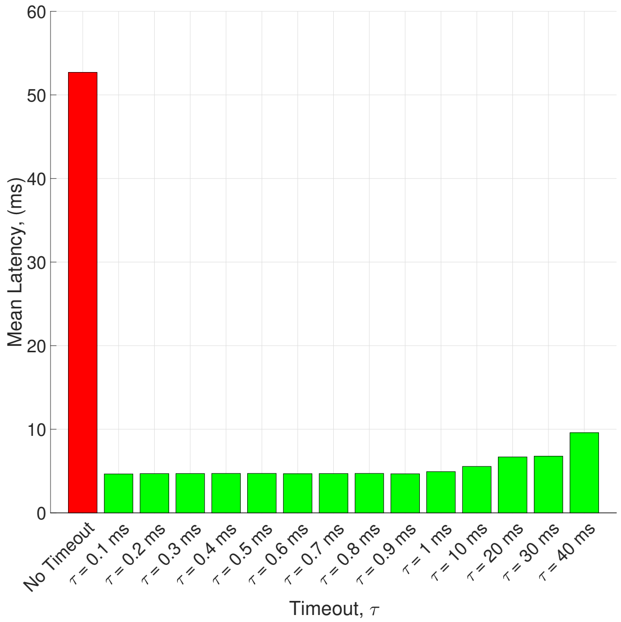


Figure 3.4: Latency comparison between the proposed approach and the benchmark UDP without a timeout.

received N packets from client c becomes

$$d_{mean,c} = \frac{1}{N}\sum_{i=1}^{N} d_{e2e,c}(i). \tag{3.8}$$

On the server side, control and feedback packets are exchanged over two dedicated ports. Furthermore, two threads are running concurrently to provide real-time communication. One thread handles server and haptic device communication. Another thread runs for server and robot communication. The server buffers the received control packets from the haptic device and transmits the most recent feedback packet back. Similarly, received feedback packets are buffered, and the server returns the most recent available control packet to the robot. Since both the control and feedback buffers have a capacity of one packet, the server always has the most recently received control and feedback packets. Furthermore, the server records packet traffic from both the haptic device and the robot. Every received packet is recorded with a timestamp, providing a chronological record of packet exchanges.

3.5.3 Experimental Results

In this section, we provide our results. We conduct measurements on our 5G-enabled teleoperation testbed. Latency and packet loss measurements are conducted, as discussed in earlier sections. In Fig. 3.4, we provide an average E2E latency comparison between the proposed approach and the benchmark UDP without a timeout. We conducted experiments for different

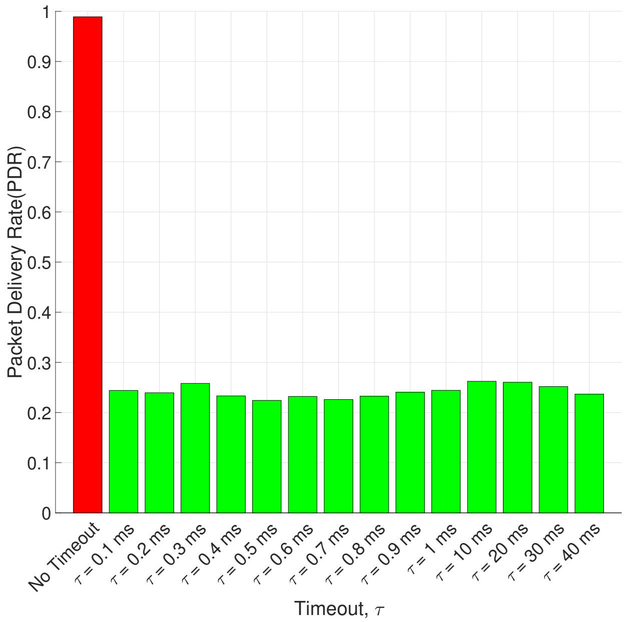


Figure 3.5: Packet delivery rate comparison between the proposed approach and the benchmark UDP without a timeout.

timeout values and for the benchmark UDP where we exchanged 10,000 packets in every experiment. As seen from the figure, the proposed approach outperforms the benchmark UDP in terms of E2E latency, where the proposed approach achieves mean E2E latency of around 5 ms while the benchmark UDP achieves mean E2E latency of around 50 ms. This is reasonable since the proposed approach reduces the waiting time in the Rx state and mitigates the long waiting time delays. On the other hand, E2E latency increases with increasing timeout period τ. This is also reasonable since the E2E latency is related to the communication latency and the timeout period.

In Fig. 3.5, we provide the reliability comparison in terms of PDR. As discussed in previous sections, the proposed approach has a negative impact on reliability. This is due to the fact that the waiting time in the Rx state is limited by the timeout period τ, which allocates very little time for packet reception. However, in order to reveal overall performance, we must consider a joint metric CF.

CF results are provided in Fig. 3.6 to compare the performance of the proposed approach with the benchmark UDP without a timeout. As seen from the figure, the proposed approach outperforms the benchmark UDP. In other words, the proposed approach requires less than 20 ms to successfully transmit a packet, whereas the benchmark UDP requires more than 50 ms. The results show that the proposed approach can achieve up to 65% less E2E latency under identical network conditions.

Furthermore, we provide the latency measurements in Fig. 3.7 for the first 300 packets to demonstrate the difference between the proposed approach and the benchmark UDP in terms

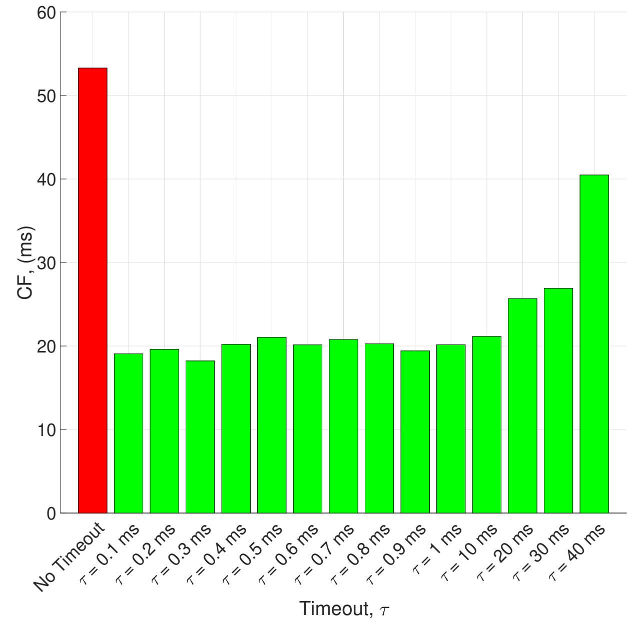


Figure 3.6: Cost function comparison between the proposed approach and the benchmark UDP without a timeout.

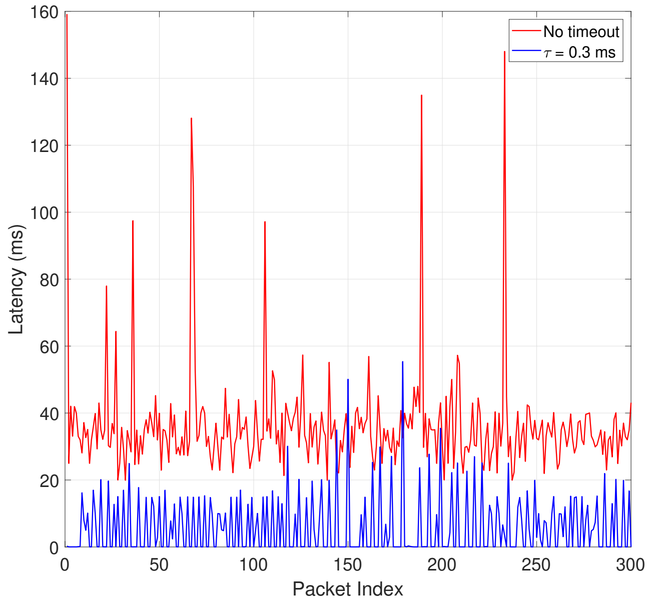
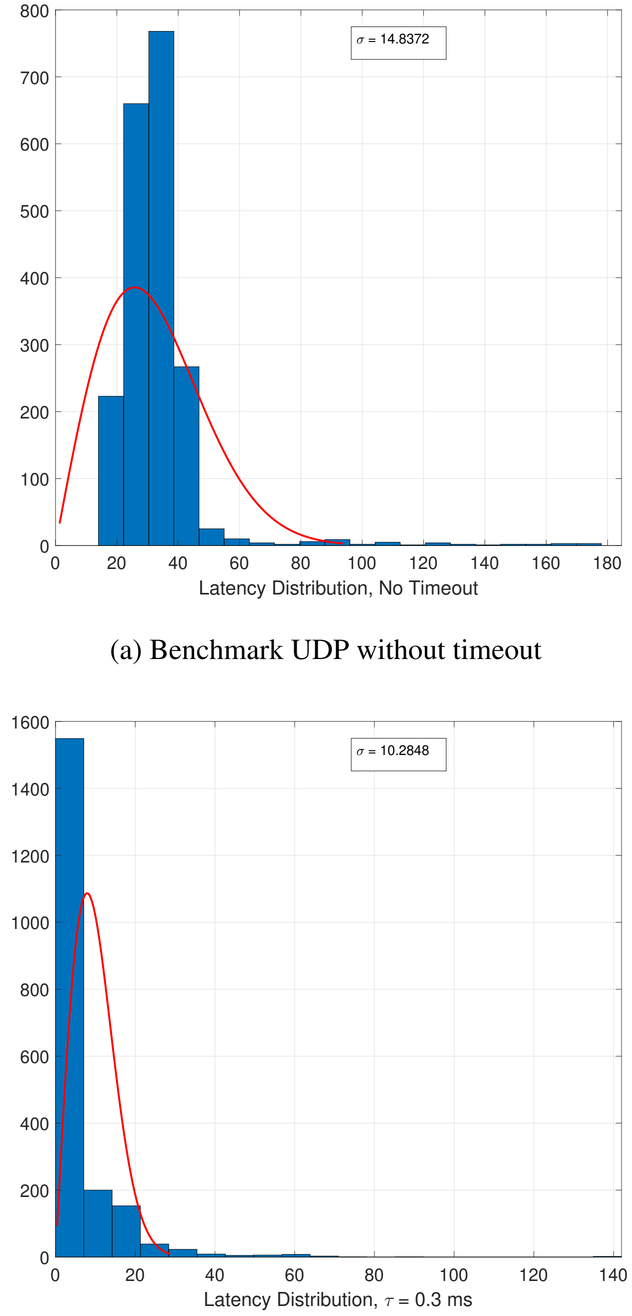


Figure 3.7: Latency values of the first 300 packets for the proposed approach with $\tau = 0.3$ ms and the benchmark UDP without a timeout.



(a) Benchmark UDP without timeout

(b) Proposed approach with $\tau = 0.3$ ms

Figure 3.8: Latency distributions of the proposed approach with $\tau = 0.3$ ms and the benchmark UDP without a timeout.

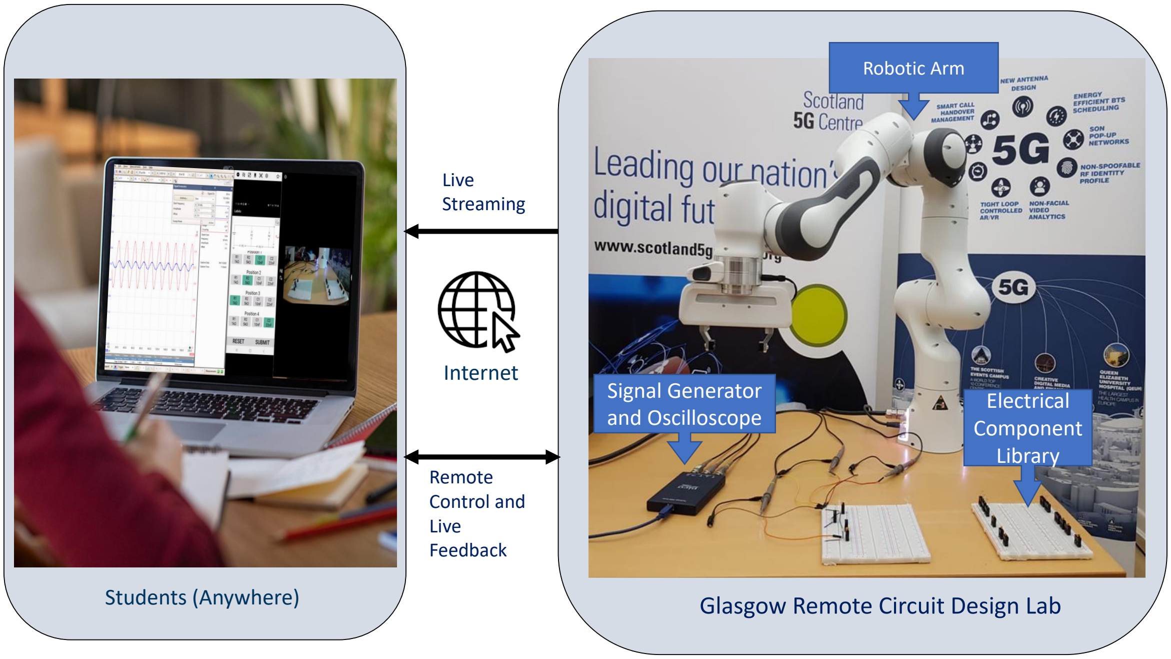

Live Streaming
Internet
Remote Control and Live Feedback
Students (Anywhere)
Robotic Arm
Signal Generator and Oscilloscope
Electrical Component Library
Glasgow Remote Circuit Design Lab

Figure 3.9: The structure of the remote circuit design lab

of jitter, which is the deviation of the latency. The figure shows that the proposed approach reduces jitter and provides more stable and predictable network conditions.To further evaluate the latency deviations, we provide latency distributions in Fig. 3.8 for the benchmark UDP (see Fig. 3.8a) and the proposed approach (see Fig. 3.8b). As seen from the figures, the proposed approach reduces the standard deviation of latency, σ, from 14.84 to 10.28 compared to the benchmark UDP which is consistent with the results in Fig. 3.7 and justifies them.

3.6 Remote Education Use Case: Education 4.0

In this section, we present one of the earliest prototypes of the Internet of Skills in Education which we call Education 4.0[3]. Education 4.0 prototype (see Fig. 3.9) offers an unrivaled experience of remote interaction to students all around the world. Physically, the lab is located at James Watt School of Engineering, University of Glasgow, UK. It is now accessible to students worldwide, where students take control of a robotic arm to conduct circuit design experiments remotely. The robotic arm is capable of assembling electrical circuits according to students' control commands by precisely placing electronic components (e.g., resistors, capacitors, etc.) on circuit boards. In addition, the remote lab prototype enables remote measurement of an actual circuit to complete lab tasks in circuit design courses.

[3] The demonstration video: https://youtu.be/RCR5l72HVuM

Table 3.1: Local Throughput, latency, and jitter measurements on 5G.

Throughput (Mbps)						**Latency (ms)**			**Jitter (ms)**		
Uplink			**Downlink**								
Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
53	54.4	56	457	526.5	553	14	14.8	16	1	2.99	5.09

3.6.1 Setup

Specifically, the remote lab enables three main capabilities. Students can observe the remote lab environment via high-quality video streaming, which provides more engagement in a remote environment. Students can control the remote robotic arm using a custom design control interface. Students are able to control lab equipment (digital signal generator and oscilloscope) to take remote measurements. To realize such an application, three main components are vital. High-quality streaming to observe remote environments, which requires video streaming software and hardware at both the user and robot end. Robotic control is vital to control the remote robotic arm, which requires control of the graphical user interface (GUI) at the user end and control of software at the robot end. Control of laboratory equipment (e.g., digital signal generator and oscilloscope) is also essential for remote experiment and measurement, which requires hardware at the laboratory and lab equipment GUI at the user end. Presented prototype can be controlled remotely by using our custom GUI. From the GUI, students select components, e.g., capacitors, and resistors, to assemble an electric circuit. After selecting components, control commands are transmitted to the remote robotic arm over a UDP-based server. A UDP-based server is used instead of a TCP-based server to further mitigate the overhead and, in turn, communication latency. Received control commands are executed by the robotic arm to assemble the circuit. In addition, real-time video feedback can be provided over different commercial video conferencing platforms such as Zoom and Microsoft Teams, allowing multiple attendees in one laboratory session and creating a more interactive environment.

3.6.2 Measurements

In the remote lab prototype, the 5G operating frequency is 3.75 GHz (i.e., it operates on band n78) with 100 MHz bandwidth. Experimental measurements on our local 5G testbed are provided in Table 3.1. In addition to 5G-based tests, we have done experiments by inviting students from UK and China to use our prototype, in which more than 200 students attended and tested the remote lab. In these experiments, the laboratory/robot end runs on 5G, and student access the remote lab over conventional internet which supports equitable learning by enabling students to access remote labs even from places where 5G is not available yet. In our remote experiments within the UK and between UK and China, we recorded communication latency of between 50 – 300 ms within the UK and up to 2 seconds between UK and China. Latency values are

Table 3.2: Existing Remote Laboratory Prototypes

Study	**Key Enabling Technologies**				
	High Quality 360^o Video Streaming	**AR/VR/XR**	**Tactile Sensing**	**AI/ML**	**Remote Interaction (Teleoperation)**
[118], [119], [120], [121], [122], [123], [124], [125]	X	X	X	X	✓
[126]	✓	✓	X	X	✓
[127]	X	X	X	✓	✓
[128]	X	✓	X	X	✓
Our prototype	✓	X	✓	✓	✓

unstable and change depending on location, internet speed, or other uncontrollable parameters in the public internet, making latency the main contributing factor to low-quality user experience. It is also worth noting that we do not provide packet loss results in Education 4.0 use case to avoid repetition considering that the reliability results are similar to remote dental inspection use case presented in previous subsection. URLLC is the most important enabler of this use case since reliable and low-latency communication is critical in such applications to ensure high quality of service. Therefore, this prototype is crucial to investigate current capabilities and further emphasize the necessity of 5G communications.

3.6.3 Discussions

The Education 4.0 prototype is crucial for demonstrating the communication requirements with real-life testbed implementation to further emphasize the necessity of 5G cellular communications. There are some existing remote laboratory prototypes in the literature. These studies are compared considering Education 4.0 key enabling technologies in Table 3.2. As seen from the comparison table, most studies only have remote interaction/control capability, while only some have AR/VR/XR capability. On the other hand, AI/ML and tactile sensing capability are not common in existing remote laboratory prototypes, which are some of the most important features of upcoming 5G-enabled Education 4.0.

Another important aspect is the reliability of the communication network. Similar to other industrial robots, the robot is sensitive to packet losses which affect the control performance. For example, it can compensate for up to 20 consecutive packet drops then the control process becomes unstable [113]. This implies that packet loss rate is very crucial in ensuring stable control, which cannot be guaranteed with current communication networks.

Another critical aspect of 5G-enabled Education 4.0 is the deployment and management strategies of the network to ensure high quality of service. There would be two strategies for the deployment and management of the network. The first is private campus networks, where universities would have their own and operate private networks by themselves. We already have well-known examples, such as the University of Glasgow, the University of Surrey, Shanghai University, the University of Tennessee, and Coventry University. The second strategy would be public networks which are owned by Mobile Network Operators (MNOs) instead of education institutions. MNOs would deploy, own, and manage the network. They can also create campus networks by using network slicing to ensure high quality of service to support 5G-enabled Education 4.0 use cases. On the other hand, one of the most critical challenges of setting up the 5G facility is the unavailability of User equipments (UEs)/dongles considering from the user perspective. Currently, the most advanced 5G UEs available are smartphones, and 5G dongle availability is quite limited, which limits 5G access and related testbed implementations. However, this challenge is expected to be solved with the rapid development of 5G hardware and the prevalence of 5G. In our case study, we have used our 5G dongles, which we built in-house at the University of Glasgow.

3.7 Chapter Summary

In this chapter, we present a 5G-enabled teleoperation prototype for Internet of Skills applications. We first provide three main domains, namely the operator domain, communication domain, and teleoperator domain. Each domain is explained in detail to provide fundamental design and implementation decisions. Furthermore, we present two Internet of Skills use cases in healthcare and education. Considering use cases, we perform communication latency and reliability experiments both locally in a laboratory environment and over the internet to investigate the current capabilities and limitations of the emerging Internet of Skills applications. We show that current communications can achieve around 15 ms E2E latency in local network settings, whereas 50 ms to 300 ms latency is observed within the UK experiments. On the other hand, the latency increases by up to 2 seconds in the experiments between UK and China, showing the effect of distance between operator and teleoperator on the latency. Around 99.9% packet reception rate is achieved in the experiments, showing the reliability of the communication network. The main concern in reliability is the number of packets lost consecutively. In other words, we observe that the prototype can tolerate a lower packet reception rate unless the number of consecutive packet losses is greater than the tolerable threshold[4]. In the remote dental inspection use case, we propose a two-way timeout approach to eliminate waiting time for stale packets by discarding the packets that are not delivered before a threshold. The proposed approach shows that the E2E latency can be improved by compromising reliability. In addition, discarding packets would affect the control performance, too. Retransmission of packets would

[4] For example, Franka Emika Panda robot can compensate for up to 20 consecutive packet drops, then the control process becomes unstable [113].

be a better approach in terms of reliability and control performance, compared to discarding packets. However, it would further increase the communication resource consumption. In brief, we observe that latency and reliability are two conflicting requirements of Internet of Skills applications. Therefore, we need to develop novel methods to achieve the stringent requirements. Considering the takeaways of this chapter, we propose two different frameworks in Chapter 4 and Chapter 5 for long-distance Internet of Skills applications to address latency and reliability challenges with co-design approach.

Chapter 4

Task-Oriented Prediction and Communication Co-Design Framework

4.1 Introduction

Haptic communications is the foundation for emerging Internet of Skills applications such as remote surgery and diagnosis in healthcare [16], remote laboratory and training in education [17], remote driving in transportation [18], and advanced manufacturing in Industry 4.0 [19]. Teleoperation is one of the most important components of the Internet of Skills systems. To provide high quality user experience, haptic communications need to meet stringent requirements on latency and reliability [21]. These requirements are well-aligned with URLLC in the fifth generation cellular networks [24]. Specifically, the E2E delay should be around 1 ms and reliability should be higher than 99.999%. However, it is very challenging to meet the latency and reliability requirements, especially when the communication distance between the operator and teloperator devices is longer than 300 km [3].

In the existing literature [4, 85–88, 94, 129], the prediction has been considered as a promising approach to meet stringent QoS requirements, such as latency and reliability. The user-experienced latency can be reduced by predicting future packets and transmitting them in advance. In addition, predictions can be used in case of packet loss to improve system reliability, as discussed in Chapter 2 in detail.

In a packetized predictive control system, predicted packets are sent to the receiver in advance and are used in case of packet loss [85]. The authors of this work jointly optimized prediction and communication systems to minimize wireless resource consumption. In [87], a two-stage prediction framework was proposed for haptic feedback prediction in remote driving to assist the driver in poor network conditions, where prediction errors were not considered [85, 87]. In [86], a haptic feedback prediction framework was proposed in remote surgery use case. Gaussian mixture regression model was used to predict haptic feedback to surgeon in a needle insertion process, where the prediction accuracy was measured by *Root Mean Square*

Error (RMSE) between predicted force and ground truth. In [88], authors proposed a predictive actuation framework by predicting missing state at receiver using previously received state. A model-based prediction algorithm was used to minimize average AoI and transmit power. However, in both [86] and [88], human perception capabilities were not taken into account.

Recently, user experience was considered in haptic communications, and was characterized by JND [39]. Here, JND is defined as the minimum difference between the operator and teleoperator that can be perceived by human users. Since human can only notice the difference when it is larger than a certain JND threshold, it is reasonable to design the predictor by considering the JND violation probability, which is defined as the probability that the difference between the predicted value and the ground truth is larger than a threshold. In [4], the authors used the JND violation probability as the performance metric to measure the reliability of the predictor and proposed a prediction and communication co-design framework to reduce user experienced delay, where a model-based prediction algorithm was deployed at the transmitter. It was demonstrated that prediction and communication co-design improves the tradeoff between user experienced delay and reliability compared to the communication system with no prediction. In [94], JND violation probability was used as the overall performance metric to illustrate the benefits of prediction on AoI in *Intelligent Transport Systems* (ITS). Similarly in [129], the tradeoff between prediction length and AoI was analyzed by using JND violation probability in prediction design. The results showed that the prediction algorithm used in the system can help to improve the AoI performance.

In practice, the JND threshold is a task dependent parameter with large variation. For example, the JND threshold value would be different when human operator controls a robotic arm for different tasks. The large JND threshold is expected when human operator has the full arm movement. In contrast, the small JND threshold is required in fine control. However, current design methods cannot capture such JND threshold dynamics since the fundamental relationship between JND threshold and QoS is not clear yet. As a result, current design methods suffer from over-provisioning of wireless resource that impedes their implementation in real-world systems.

In addition, it is very challenging to obtain the relationship between JND threshold and QoS. Unlike communication systems that are built upon fundamental theories, most of the prediction algorithms are developed via data-driven design. This raises a huge challenge when the required error probability is at the level of 10^{-5}, which requires a large number of real-world data samples to evaluate the probability of the rare event. If we use model-based prediction, it is possible to derive the prediction error probability, but the mismatch between the over simplified theoretical models and practical complex systems will lead to inaccurate results. Therefore, we need to develop innovative methods to overcome the above challenge.

In this chapter, we propose a task-oriented prediction and communication co-design framework in the context of teleoperation system, where the utilization efficiency of the communication system is maximized subject to the requirements of different operating tasks. In particular, in the scenario with limited real-world data samples, we generate synthetic data via TimeGAN. This allows us to obtain the relationship between JND threshold and the prediction error probability that is below 10^{-5}. We take 5G New Radio as an example to demonstrate the proposed framework, where we compare the performance of the systems with and without task-oriented design. We further compare the performance difference of deploying the predictor at transmitter and receiver sides. Specifically, the main contributions of this chapter are listed below:

- We propose a task-oriented prediction and communication co-design framework with a predictor at the receiver side. We derived an upper bound of the overall error probability in the framework by taking packet losses and prediction errors into account. From the upper bound, we reveal the tradeoff between the resource utilization efficiency and the overall reliability.

- To illustrate how to use this framework in practical system design, we take remote robotic control in 5G New Radio as an example. Then, we formulate an optimization problem to optimize bandwidth allocation and communication data rate subject to constraints on the E2E delay and overall reliability. An optimization algorithm is proposed to find the optimal solution.

- We collect the real-world data from a teleoperation prototype. We further use TimeGAN to generate synthetic data for predictor training and testing. With both synthetic data and real-world data, we illustrate the tradeoff between the prediction horizon and prediction error probability, and further evaluate the overall reliability. Our results show that the proposed task-oriented prediction framework can save up to 77.80% bandwidth compared with a benchmark design that is task-agnostic.

The rest of the chapter is organized as follows. In Section 4.2, we develop a general design framework for task-oriented prediction and communication co-design. In Section 4.3, we illustrate packet losses in communications by taking 5G New Radio as an example. In Section 4.4, we introduce prediction algorithms. In Section 4.5, we propose efficient resource allocation with task-oriented prediction. In Section 4.6, we present simulation and numerical results. Section 4.7 concludes this chapter. Notations used throughout the chapter are listed in Table 4.1 for clarification.

Table 4.1: Descriptions of Notations

Notation	Description	Notation	Description
$K_m(t)$	control command send by m-th transmitter at time slot t	E_m^{B}	effective bandwidth for the m-th link
n	number of features in a control command	b_m	number of bits per packet for the m-th link
D_m^{c}	E2E communication delay for the m-th link	W_m	bandwidth for the m-th link
D_m^{e}	user experienced delay for the m-th link	C_m	capacity for the m-th link
D_m^{r}	core network and backhaul delay for the m-th link	R_m	achievable rate for the m-th link
D_m^{q}	queuing delay for the m-th link	V_m	channel dispersion for the m-th link
D_m^{t}	transmission delay for the m-th link	γ_m	SNR for the m-th link
D_m^{ch}	channel coherence time for the m-th link	α_m	large-scale gain for the m-th link
D_m^{th}	queuing delay threshold for the m-th link	g_m	small-scale gain for the m-th link
T_m^{p}	prediction horizon for the m-th task	P_m	maximum transmit power for the m-th link
T_{th}	prediction horizon threshold	N_0	single sided noise spectral density
$D_m^{\max}$	delay requirement for the m-th task	l_m	blocklength for the m-th link
$\varepsilon_m^{\mathrm{q}}$	queuing delay bound violation probability for the m-th task	A	number of critical tasks
$\varepsilon_m^{\mathrm{d}}$	decoding error probability for the m-th task	B	number of non-critical tasks
$\varepsilon_m^{\mathrm{p}}$	prediction error probability for the m-th task	$Q(.)$	Q-function
$\varepsilon_m^{\mathrm{o}}$	overall error probability for the m-th task	$Q^{-1}(.)$	inverse of the Q-function
$\varepsilon_m^{\max}$	maximum tolerable error probability for the m-th task	$\mathrm{W}_{-1}(.)$	-1 branch of Lambert W-function
δ_m	JND threshold for the m-th task	$f_{\varepsilon_m^{\mathrm{p}}}(.)$	function of prediction error probability for the m-th task
λ_m	average packet arrival rate for the m-th link	$f_{\varepsilon_m^{\mathrm{q}}}(.)$	function of queuing delay bound violation probability for the m-th task

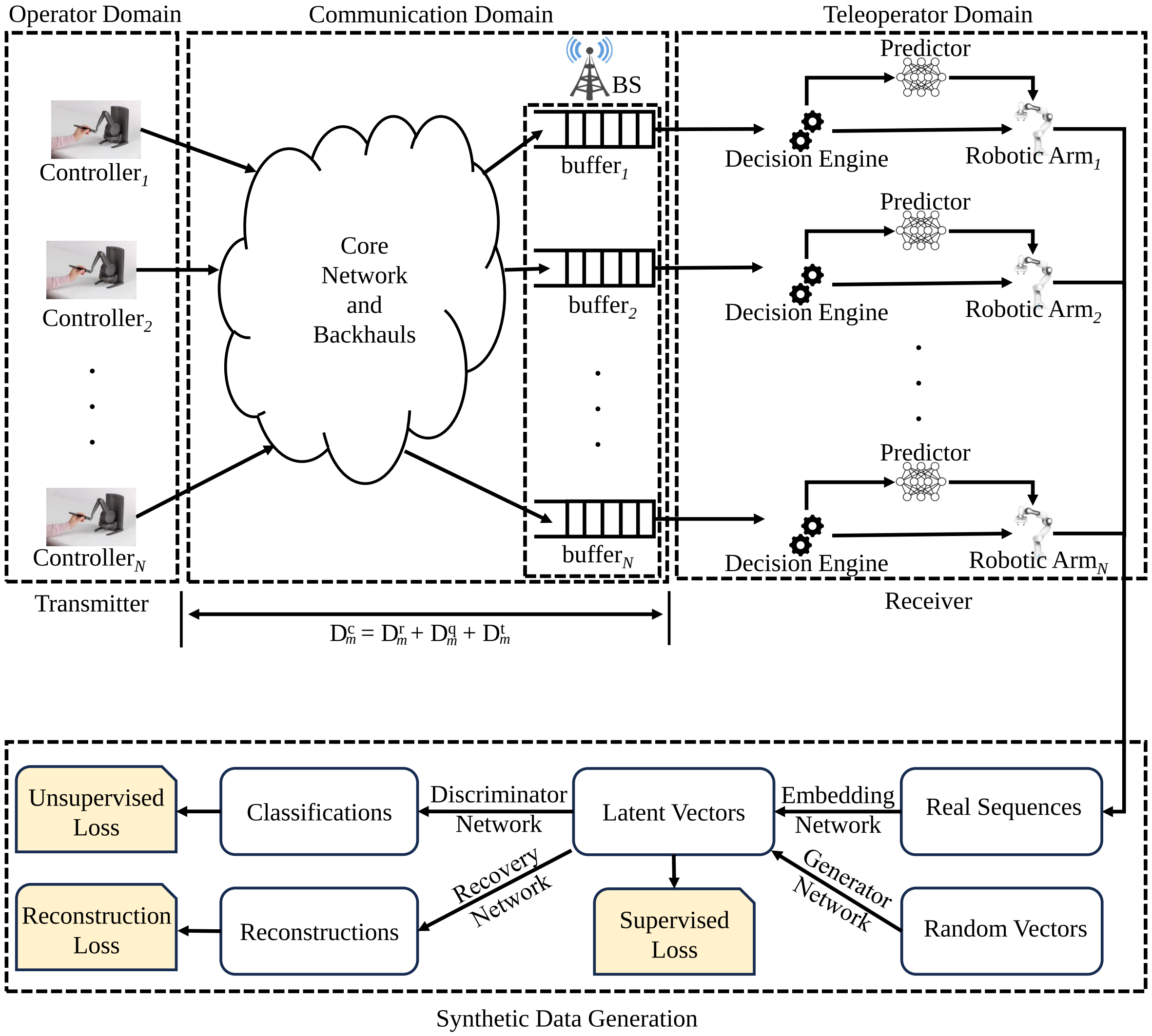


Figure 4.1: System model

4.2 A General Design Framework

We consider a haptic communication system shown in Fig. 4.1, where multiple human users can remotely control robots via a communication system. N pairs of transmitters and receivers are considered over a shared wireless channel, where orthogonal subchannels are assigned to different transceiver pairs to avoid interference. The predictors are deployed at receivers to reduce user experienced delays.

4.2.1 User Experienced Delay and Delay Requirement

Time is discretized into slots. At the t-th time slot, the m-th transmitter sends the control command $K_m(t) = [k_m^1(t), k_m^2(t), \ldots, k_m^n(t)]^T$ to the receiver, where n is the number of features in the command (e.g., joint angles, angular velocity, forces, and torque). Then, $K_m(t)$ arrives at the

m-th receiver at $(t+D_m^{\mathrm{c}})$-th time slot, where D_m^{c} is E2E communication delay. If the communication system has no prediction capability, the time delay experienced by the user, denoted as D_m^{e}, is the same as the communication delay, i.e., $D_m^{\mathrm{e}} = D_m^{\mathrm{c}}$[1]. Here, the communication delay D_m^{c} consists of the delay in core network and backhauls D_m^{r}, the queuing delay in the buffer of the base station (BS) D_m^{q}, and the transmission delay in the radio access network D_m^{t}. Then, we have $D_m^{\mathrm{c}} = D_m^{\mathrm{r}} + D_m^{\mathrm{q}} + D_m^{\mathrm{t}}$.

In this work, D_m^{r} and D_m^{t} are assumed to be bounded by a constant, but D_m^{q} is a random variable. As a result, the communication delay D_m^{c} is a random variable that could be longer than the maximum tolerable delay bound $D_m^{\max}$. To reduce the user experienced delay D_m^{e}, a predictor is equipped at each receiver to predict the delayed or lost trajectories. We denote the prediction horizon of the m-th user by T_m^{p}, which is smaller than the maximum prediction horizon T_{th}. This means that the predictor cannot predict the trajectory beyond the maximum prediction horizon T_{th} because the temporal correlation of data becomes very weak.

Since the user experience delay is determined by the relationship among the communication delay D_m^{c} and the prediction horizon T_{th}, we need to consider the following three cases:

- Case 1: $D_m^{\mathrm{c}} \in (0, D_m^{\max}]$

  The receiver only needs the predicted trajectory when some packets are lost. The prediction horizon depends on the number of consecutive packet losses. We consider a block fading channel. The channel gain remains constant within duration D_m^{ch}, and varies independently from one duration to another. D_m^{ch} is referred to as the channel coherence time. When the wireless channel is in deep fading, most of the packets cannot be decoded by the receiver. Given this fact, it is reasonable to assume that D_m^{ch} is upper bound of the time horizon with consecutive packet losses.[2] With the block fading channel, the prediction horizon is bounded by $T_m^{\mathrm{p}} \leq D_m^{\mathrm{ch}}$. In this case, the user experienced delay is equal to the communication delay, i.e.,
  $$D_m^{\mathrm{e}} = D_m^{\mathrm{c}} \leq D_m^{\max}.$$

- Case 2: $D_m^{\mathrm{c}} \in (D_m^{\max}, T_{\mathrm{th}} + D_m^{\max}]$

  To satisfy the delay requirement, the prediction horizon is $T_m^{\mathrm{p}} = D_m^{\mathrm{c}} - D_m^{\max}$. With the help of the predictor, the user experienced delay becomes
  $$D_m^{\mathrm{e}} = D_m^{\mathrm{c}} - T_m^{\mathrm{p}} = D_m^{\max}.$$

- Case 3: $D_m^{\mathrm{c}} \in (T_{\mathrm{th}} + D_m^{\max}, \infty)$

[1] There are different delay components such as computing delay and control delay. However, we focus on communication system design in this work and ignore other delay components.

[2] The probability that the channel stays in deep fading in multiple consecutive blocks is extremely small, i.e., much smaller than 10^{-5}, and hence is not considered.

In this case, it is not possible to meet the delay requirement even with the maximum prediction horizon. Then, the user experienced delay becomes

$$D_m^{\mathrm{e}} = D_m^{\mathrm{c}} - T_m^{\mathrm{p}} > D_m^{\max}.$$

For haptic communications, the delay experienced by a user should not exceed a delay bound. The maximum tolerable delay bound of the m-th user is denoted by $D_m^{\max}$. Then, the user experienced delay, D_m^{e}, should satisfy the following requirement

$$D_m^{\mathrm{e}} \leq D_m^{\max}. \tag{4.1}$$

It is worth noting that the above constraint cannot be satisfied with probability one since the queuing delay is stochastic in most of communication systems. In the next subsection, we will analyze the delay bound violation probability.

4.2.2 Reliability Components and Reliability Requirement

Let's denote the overall error probability and the maximum tolerable error probability of the m-th task by $\varepsilon_m^{\mathrm{o}}$ and $\varepsilon_m^{\max}$, respectively. The overall error probability consists of decoding error probability, $\varepsilon_m^{\mathrm{d}}$, queuing delay bound violation probability, $\varepsilon_m^{\mathrm{q}}$, and prediction error probability, $\varepsilon_m^{\mathrm{p}}$.

The decoding error probability of the m-th user depends on the channel fading, the resource allocation policy, and the modulation and coding scheme in wireless communications. We need to optimize the communication system to obtain a satisfactory decoding error probability. In the next section, we will provide the expression of $\varepsilon_m^{\mathrm{d}}$ in a specific communication system.

The prediction error probability is defined as the probability that the prediction error is larger than the required JND threshold of a task. By considering JND violation probability, we take position control accuracy into account which is the discrepancy between operator's position and robot's position. In other words, it is the probability of tracking error being larger than a JND threshold that can noticeable by the operator. For the m-th user, the JND threshold is denoted by δ_m. For a given prediction algorithm, the relationship between prediction error probability, $\varepsilon_m^{\mathrm{p}}$, the prediction horizon, T_m^{p}, and the JND threshold is characterized by the following function,

$$\begin{aligned} f_{\varepsilon_m^{\mathrm{p}}}(T_m^{\mathrm{p}}, \delta_m) = & \Pr\{|\hat{K}_m(t+T_m^{\mathrm{p}}) - K_m(t+T_m^{\mathrm{p}})| > \delta_m\}, \\ & t = 1,2,3,... \end{aligned} \tag{4.2}$$

where $\hat{K}_m(t+T_m^{\mathrm{p}})$ is the predicted trajectory for the $(t+T_m^{\mathrm{p}})$-th slot and $K_m(t+T_m^{\mathrm{p}})$ is the actual trajectory in this slot. As shown in [4], $f_{\varepsilon_m^{\mathrm{p}}}(T_m^{\mathrm{p}}, \delta_m)$ increases with T_m^{p} and decreases with δ_m.

Similarly, we denote the relationship between the queuing delay bound and the delay bound violation probability by a function $f_{\varepsilon_m^{\mathrm{q}}}(\kappa)$. It is the probability that the queuing delay, D_m^{q}, is greater than a queuing delay bound, κ, i.e.,

$$f_{\varepsilon_m^{\mathrm{q}}}(\kappa) = \Pr\{D_m^{\mathrm{q}} > \kappa\}. \tag{4.3}$$

Since the delay bound violation probability decreases with the required delay bound, $f_{\varepsilon_m^{\mathrm{q}}}(\kappa)$ is a monotonic decreasing function. To meet the maximum delay bound, a threshold of queuing delay is given by $D_m^{\mathrm{th}} = D_m^{\max} - D_m^{\mathrm{t}} - D_m^{\mathrm{r}}$.

In the sequel, we analyze the overall error probability in the three cases discussed in the previous subsection.

- Case 1: $D_m^{\mathrm{c}} \in (0, D_m^{\max}]$

  In this case, we have $D_m^{\mathrm{q}} \leq D_m^{\mathrm{th}}$. From the definition of $f_{\varepsilon_m^{\mathrm{q}}}(\kappa)$, the probability that the queuing delay does not exceed D_m^{th} can be expressed as

  $$\Pr\{D_m^{\mathrm{q}} \leq D_m^{\mathrm{th}}\} = 1 - f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}}). \tag{4.4}$$

  Since the maximum prediction horizon does not exceed D_m^{ch}, the prediction error probability is bounded by $f_{\varepsilon_m^{\mathrm{p}}}(D_m^{\mathrm{ch}}, \delta_m)$. Given the decoding error probability, $\varepsilon_m^{\mathrm{d}}$, the error probability in case 1 can be expressed as $\varepsilon_m^{\mathrm{c},1} = f_{\varepsilon_m^{\mathrm{p}}}(D_m^{\mathrm{ch}}, \delta_m)\varepsilon_m^{\mathrm{d}}$.

- Case 2: $D_m^{\mathrm{c}} \in (D_m^{\max}, T_{\mathrm{th}} + D_m^{\max}]$

  In this case, $D_m^{\mathrm{q}} \in (D_m^{\mathrm{th}}, D_m^{\mathrm{th}} + T_{\mathrm{th}}]$, which happens with a probability of

  $$\begin{aligned} &\Pr\{D_m^{\mathrm{q}} \in (D_m^{\mathrm{th}}, D_m^{\mathrm{th}} + T_{\mathrm{th}}]\} = \\ &f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}}) - f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}} + T_{\mathrm{th}}). \end{aligned} \tag{4.5}$$

  The prediction horizon is bounded by T_{th}, and thus the error probability is bounded by $\varepsilon_m^{\mathrm{c},2} \leq f_{\varepsilon_m^{\mathrm{p}}}(T_{\mathrm{th}}, \delta_m)$.

- Case 3: $D_m^{\mathrm{c}} \in (T_{\mathrm{th}} + D_m^{\max}, \infty)$

  In this case, $D_m^{\mathrm{q}} \in (T_{\mathrm{th}} + D_m^{\mathrm{th}}, \infty)$. Case 3 happens with a probability of

  $$\Pr\{D_m^{\mathrm{q}} \in (T_{\mathrm{th}} + D_m^{\mathrm{th}}, \infty)\} = f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}} + T_{\mathrm{th}}). \tag{4.6}$$

  Since the delay requirement is not satisfied, all the packets are lost. The error probability in this case is $\varepsilon_m^{\mathrm{c},3} = 1$.

The overall error probability is a combination of error probabilities in the above three cases. It can be expressed as follows,

$$\begin{aligned}\varepsilon_m^{\mathrm{o}} =&\varepsilon_m^{\mathrm{c},1}\Pr\{D_m^{\mathrm{q}}\le D_m^{\mathrm{th}}\}+\varepsilon_m^{\mathrm{c},2}\Pr\{D_m^{\mathrm{q}}\in(D_m^{\mathrm{th}},D_m^{\mathrm{th}}+T_{\mathrm{th}}]\}\\ &+\varepsilon_m^{\mathrm{c},3}\Pr\{D_m^{\mathrm{q}}\in(T_{\mathrm{th}}+D_m^{\mathrm{th}},\infty)\}\\ \le& f_{\varepsilon_m^{\mathrm{p}}}(D_m^{\mathrm{ch}},\delta_m)\varepsilon_m^{\mathrm{d}}(1-f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}}))\\ &+f_{\varepsilon_m^{\mathrm{p}}}(T_{\mathrm{th}},\delta_m)(f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}})-f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}}+T_{\mathrm{th}}))\\ &+f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}}+T_{\mathrm{th}}).\end{aligned} \tag{4.7}$$

It is worthy to note that the overall error probability is formulated considering both communication and prediction errors to take implications of wrong predictions into account in the system design. The reliability requirement $\varepsilon_m^{\mathrm{o}}\le\varepsilon_m^{\max}$ can be satisfied if the upper bound in (4.7) meets the following constraint,

$$\begin{aligned}&f_{\varepsilon_m^{\mathrm{p}}}(D_m^{\mathrm{ch}},\delta_m)\varepsilon_m^{\mathrm{d}}(1-f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}}))+f_{\varepsilon_m^{\mathrm{p}}}(T_{\mathrm{th}},\delta_m)(f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}})\\ &-f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}}+T_{\mathrm{th}}))+f_{\varepsilon_m^{\mathrm{q}}}(D_m^{\mathrm{th}}+T_{\mathrm{th}})\le\varepsilon_m^{\max}.\end{aligned} \tag{4.8}$$

We denote the general utilization efficiency of a communication system by $U(\mathbf{x})$, where $\mathbf{x}=[x_1,...,x_N]$ is the optimization variables of the N tasks. A general task-oriented prediction and communication co-design framework can be formulated as follows,

$$\begin{aligned}&\max_{\mathbf{x}}\, U(\mathbf{x}) \qquad (4.9)\\ &\text{s.t. (4.1) and (4.8).}\end{aligned}$$

With this framework, we can jointly optimize prediction and communication systems to achieve better resource utilization efficiency.

4.3 5G New Radio: An Example of Communication System

To illustrate how to obtain the upper bound in (4.7), we consider 5G New Radio as an example in the rest part of this work and derived the decoding error probability, $\varepsilon_m^{\mathrm{d}}$, and the queuing delay bound violation probability, $f_{\varepsilon_m^{\mathrm{q}}}(.)$, in this section.

4.3.1 Decoding Error Probability

To achieve low transmission delay, the blocklength of channel codes is short. In the finite blocklength regime, the maximal achievable rate can be accurately approximated as [130]

$$R_m \approx C_m - \sqrt{\frac{V_m}{l_m}} Q^{-1}(\varepsilon_m^{\mathrm{d}}) \text{ (bits/s/Hz)}, \tag{4.10}$$

where $C_m = \log(1+\gamma_m)$ is the Shannon capacity, $\gamma_m = \frac{\alpha_m g_m P_m}{N_0 W_m}$ is the received signal to noise ratio (SNR) at BS, α_m is the large-scale channel gain, g_m denotes the small-scale channel gain, P_m denotes the transmit power, N_0 is the single sided noise spectral density, $V_m = \log(e)^2 \left[1 - \frac{1}{(1+\gamma_m)^2}\right]$ is the channel dispersion, $l_m = D_m^{\mathrm{t}} W_m$ is the blocklength, D_m^{t} is the transmission duration, W_m is the bandwidth, and $Q^{-1}(.)$ is the inverse of the Q-function. Then, decoding error probability can be expressed as

$$\varepsilon_m^{\mathrm{d}} \approx Q\left(\frac{D_m^{\mathrm{t}} W_m C_m - b_m + \log(D_m^{\mathrm{t}} W_m)/2}{\sqrt{D_m^{\mathrm{t}} W_m V_m}}\right), \tag{4.11}$$

where $b_m = D_m^{\mathrm{t}} W_m R_m$ is the number of information bits. If SNR is higher than 5dB, channel dispersion V_m becomes $\log(e)^2$ [131]. Then, $\varepsilon_m^{\mathrm{d}}$ becomes

$$\varepsilon_m^{\mathrm{d}} \approx Q\left(\frac{D_m^{\mathrm{t}} W_m C_m - b_m + \log(D_m^{\mathrm{t}} W_m)/2}{\log(e)\sqrt{D_m^{\mathrm{t}} W_m}}\right). \tag{4.12}$$

From (4.12), we can obtain the following Lemma.

Lemma 1: Given W_m, $\varepsilon_m^{\mathrm{d}}$ increases with b_m.

Proof: Given W_m, the input of the Q-function in (4.12) decreases with b_m. Since Q-function is a decreasing function, $\varepsilon_m^{\mathrm{d}}$ increases with increasing b_m.

4.3.2 Queuing Delay Violation Probability

In URLLC, the transmission delay (transmission time interval could be 0.125 ms in 5G) is much shorter than the channel coherence time. Thus, the service rate of the queuing system is a constant. We use effective bandwidth to characterize the minimum service rate that is required to achieve the delay bound and delay bound violation probability [132]. To derive the closed-form expression of queuing delay violation probability, we further assume that the packet arrival processes are Poisson processes. Denote the average packet arrival rate of the m-th user by λ_m with the unit of (packets/s). As discussed in [4, 133], the queuing delay violation probability decreases exponentially as the delay bound increases, i.e.,

$$f_{\varepsilon_m^{\mathrm{q}}}(\kappa) = e^{\kappa\xi(E_m^{\mathrm{B}},\lambda_m)}, \tag{4.13}$$

$$\xi(E_m^{\mathrm{B}},\lambda_m) = E_m^{\mathrm{B}} \mathrm{W}_{-1}\left(-\frac{\lambda_m}{E_m^{\mathrm{B}}} e^{-\frac{\lambda_m}{E_m^{\mathrm{B}}}}\right) + \lambda_m, \tag{4.14}$$

$$\begin{aligned}\frac{\partial f_{\varepsilon_m^{\mathrm{q}}}}{\partial b_m} =& \kappa \exp\left\{ \kappa \left(\lambda_m + \frac{b_m \mathrm{W}_{-1}\left(-\frac{D_m^{\mathrm{t}} \lambda_m \mathrm{e}^{-\frac{D_m^{\mathrm{t}} \lambda_m}{b_m}}}{b_m} \right)}{D_m^{\mathrm{t}}} \right) \right\} \times \left(\frac{\mathrm{W}_{-1}\left(-\frac{D_m^{\mathrm{t}} \lambda_m \mathrm{e}^{-\frac{D_m^{\mathrm{t}} \lambda_m}{b_m}}}{b_m} \right)}{D_m^{\mathrm{t}}} \right. \\ & \left. - \frac{b_m^2 \mathrm{e}^{\frac{D_m^{\mathrm{t}} \lambda_m}{b_m}} \left(\frac{D_m^{\mathrm{t}} \lambda_m \mathrm{e}^{-\frac{D_m^{\mathrm{t}} \lambda_m}{b_m}}}{b_m^2} - \frac{{D_m^{\mathrm{t}}}^2 \lambda_m^2 \mathrm{e}^{-\frac{D_m^{\mathrm{t}} \lambda_m}{b_m}}}{b_m^3} \right) \mathrm{W}_{-1}\left(-\frac{D_m^{\mathrm{t}} \lambda_m \mathrm{e}^{-\frac{D_m^{\mathrm{t}} \lambda_m}{b_m}}}{b_m} \right)}{{D_m^{\mathrm{t}}}^2 \lambda_m \left(\mathrm{W}_{-1}\left(-\frac{D_m^{\mathrm{t}} \lambda_m \mathrm{e}^{-\frac{D_m^{\mathrm{t}} \lambda_m}{b_m}}}{b_m} \right) + 1 \right)} \right) < 0.\end{aligned} \tag{4.17}$$

where $\mathrm{W}_{-1}(.)$ is the -1 branch of Lambert W-function, which is defined as the inverse function of $f(x) = xe^x$ and E_m^{B} is the effective bandwidth, which is the required service rate. To meet the queuing delay bound κ and the delay bound violation probability in (4.13), the number of bits transmitted in a transmission time interval should satisfy the following expression

$$b_m / D_m^{\mathrm{t}} = E_m^{\mathrm{B}}. \tag{4.15}$$

Upon substituting E_m^{B} into (4.13) and (4.14), the queuing delay bound violation probability can be expressed as

$$f_{\varepsilon_m^{\mathrm{q}}}(\kappa) = \exp\left\{ \kappa \left[\frac{b_m \mathrm{W}_{-1}\left(-\frac{\lambda_m D_m^{\mathrm{t}}}{b_m} e^{-\frac{\lambda_m D_m^{\mathrm{t}}}{b_m}} \right)}{D_m^{\mathrm{t}}} + \lambda_m \right] \right\}. \tag{4.16}$$

With the expression given in (4.16), the following property of queuing delay bound violation probability function can be obtained.

Lemma 2: For given κ, λ_m, and D_m^{t}, $f_{\varepsilon_m^{\mathrm{q}}}(\kappa)$ strictly decreases with b_m.

Proof: To check the monotonicity of $f_{\varepsilon_m^{\mathrm{q}}}(\kappa)$ in terms of b_m, we have the partial derivative in (4.17).

Since $W_{-1}(.)$ is always negative, and exponential function is always positive, the partial derivative in (4.17) is negative. Therefore, $f_{\varepsilon_m^{\mathrm{q}}}(\kappa)$ decreases with b_m when κ, λ_m, and D_m^{t} are given.

4.4 TimeGAN Assisted Prediction: An Example of Prediction Algorithm

To characterize the tradeoff between the prediction error probability, the prediction horizon, and the JND threshold, we require a large number of trajectories for evaluating the prediction error probability below 10^{-5}. [3] However, collecting enough trajectories in a real-world robotic platform may take years. To address this issue, we generate synthetic data using TimeGAN. In this section, we provide details of real-world data collection, synthetic data generation, training and testing of different prediction algorithms. It is also worth noting that the prediction performed in this work is only for robot's trajectory, i.e., it does not include prediction of communication latency or reliability.

4.4.1 Real-world Data Set Collection

The real-world trajectory data samples are collected from our teleoperation testbed[4]. The robotic arm is controlled by a human user to finish three types of tasks as shown in Fig. 4.2.

1. *Pushing a box:* Push a small box from the starting point to the end point along a given routine.

2. *Grouping items with different colors:* Move items with the same color to the same area.

3. *Writing symbols:* Write symbols by controlling the robotic arm.

Each trajectory is a time-series of observations given by $\mathbf{k}_{t:t'} = \{\mathbf{k}_t, \mathbf{k}_{t+1}, ..., \mathbf{k}_{t'}\}$ where each observation, $\mathbf{k}_t = [q_t, \dot{q}_t]$, consists of an angular position, q_t, and angular velocity, $\dot{q}_t$, in the t-th time slot. In experiments, we recorded around 1.7×10^7 observations with timestamps at the frequency of one thousand observations per second.

4.4.2 Synthetic Data Set Generation

To generate synthetic data, we apply TimeGAN [70], which is a framework for time-series data generation. TimeGAN framework consists of four neural networks namely embedding network, recovery network, sequence generator, and sequence discriminator. The parameters of the four neural networks are denoted by θ_{em}, θ_{re}, θ_{g}, and θ_{d}, respectively.

Embedding and recovery networks are mappings from feature space to latent space and vice versa. Let's denote the latent variable and the latent space by $\mathbf{v}_{i:i'}$ and $\mathscr{V}_{\mathscr{T}}$, respectively. Then, the embedding network outputs a sequence of latent variables from a given sequence of features, i.e.,

[3] The reliability requirement is defined by the 5G standard in [24].

[4] The demonstration video: https://youtu.be/c3onK5Vh6QE

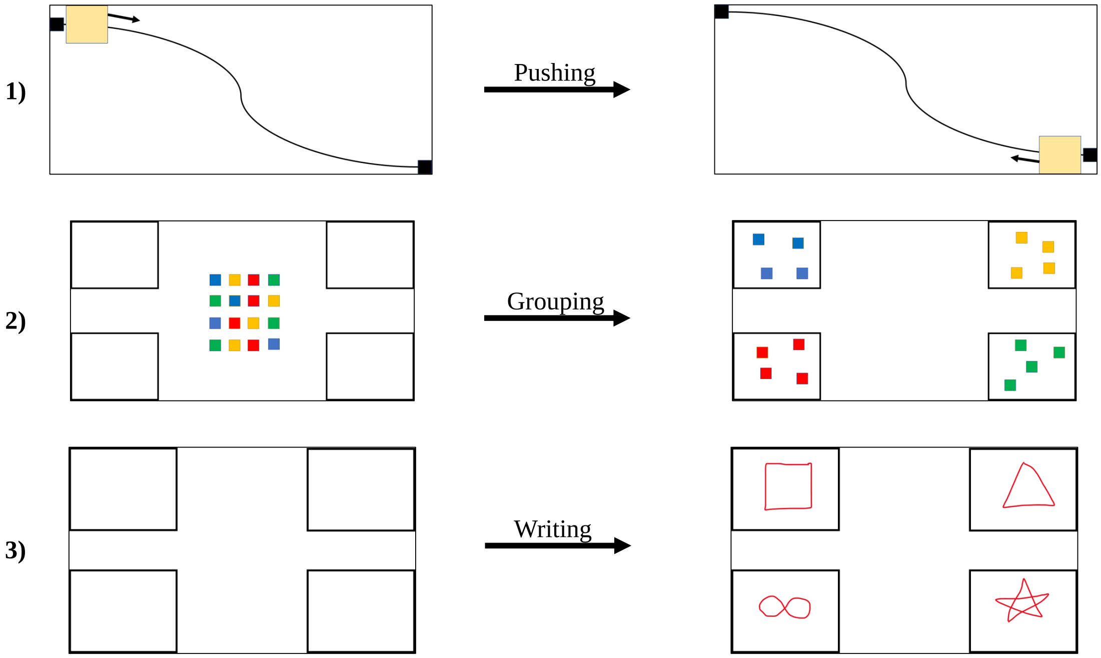


Figure 4.2: Tasks for data collection.

$\mathbf{v}_{i:i'} = \vartheta_{\text{em}}(\mathbf{k}_{i:i'}|\theta_{\text{em}})$. The recovery network reconstructs features from latent variables according to $\hat{\mathbf{k}}_{i:i'} = \vartheta_{\text{re}}(\mathbf{v}_{i:i'}|\theta_{\text{re}})$.

The generator network is denoted by $\hat{\mathbf{v}}_{i:i'} = \vartheta_{\text{g}}(\mathbf{z}_{i:i'}|\theta_{\text{g}})$. It takes a sequence of random variables $\mathbf{z}_{i:i'}$ from a known distribution as its input and generates synthetic sequences in the latent space, $\hat{\mathbf{v}}_{i:i'}$. The discriminator is a classification network in the latent space. Given a sequence of latent variables, a synthetic sequence or a real sequence, the output of the discriminator is an indicator $\tilde{y} = \vartheta_{\text{d}}(\mathbf{v}_{i:i'}|\theta_{\text{d}})$. If $\tilde{y} = 1$ the sequence is classified as a real one. Otherwise, the sequence is classified as a synthetic one.

In TimeGAN, the four components are trained jointly with three loss functions: reconstruction loss, $\mathscr{L}_{\text{R}}$, unsupervised loss, $\mathscr{L}_{\text{U}}$, and supervised loss, $\mathscr{L}_{\text{S}}$.

Reconstruction loss is used to measure the difference between the reconstructed features, $\hat{\mathbf{k}}_{i:i'}$, and the original features, $\mathbf{k}_{i:i'}$ i.e.,

$$\mathscr{L}_{\text{R}} = \mathbb{E}_{\mathbf{k}_{i:i'}}\left[\sum_t ||\mathbf{k}_t - \hat{\mathbf{k}}_t||_2\right]. \tag{4.18}$$

Unsupervised loss comes from the zero-sum game as in conventional GAN [71]. It maximizes the likelihood of correct classifications for the discriminator. Supervised loss, on the other hand, is introduced to check the discrepancy between real and synthetic data distributions in latent space. The definitions of $\mathscr{L}_{\text{U}}$ and $\mathscr{L}_{\text{S}}$ can be found in [70, 134].

The generator and discriminator networks are trained iteratively by solving the following

Table 4.2: TimeGAN model and Hyper-parameters.

Hyper-parameters	**Values**
Sequence length	600
Number of features	2
Hidden units for generator	24
Gamma (used for discriminator loss)	1
Noise dimension (used by generator as a starter dimension)	32
Number of layers	128
Batch size	128
Learning rate	5x10^{-4}

problem,

$$\min_{\theta_{\text{em}},\theta_{\text{re}}} \quad \alpha\mathscr{L}_{\text{R}} + \mathscr{L}_{\text{S}}, \tag{4.19}$$

where $\alpha \geq 0$ is a hyperparameter.

The embedding and recovery networks are trained to minimize $\mathscr{L}_{\text{R}}$ and $\mathscr{L}_{\text{S}}$ which yields to following optimization

$$\min_{\theta_g} \quad \eta\mathscr{L}_{\text{S}} + \max_{\theta_d} \mathscr{L}_{\text{U}}, \tag{4.20}$$

where $\eta \geq 0$ is a hyperparameter.

In our training, we have implemented the TimeGAN in Tensorflow 2.0 [135] using original implementation in [136], and *ydata-synthetic* package in [137]. The hyper-parameters are listed in Table 4.2.

4.4.3 Prediction Algorithms

We design trajectory prediction algorithms with three different types of neural networks(NNs): Recurrent Neural Networks (RNN) [63], Long Short Term Memory (LSTM) networks [64], and Convolutional Neural Networks (CNN) [62].

RNN for Prediction

In the t-th time slot, the input of a RNN cell includes the feature observed in the current slot $\mathbf{k}_t$, and the hidden state generated by the previous RNN cell, $\mathbf{h}_{t-1}$. Then, RNN model updates the output and hidden states from the following steps,

$$\mathbf{o}_t = \sigma(\mathbf{W}_{\text{o}}[\mathbf{h}_{t-1}, \mathbf{k}_t] + \mathbf{b}_{\text{o}}) \text{ and } \mathbf{h}_t = \sigma(\mathbf{W}_{\text{h}}\mathbf{o}_t + \mathbf{b}_{\text{h}}),$$

where $\mathbf{o}_t$ is the output of the RNN cell, $\mathbf{h}_t$ is the new hidden state, $\mathbf{W}_\mathrm{o}$ and $\mathbf{W}_\mathrm{h}$ are weight matrices, $\mathbf{b}_\mathrm{o}$ and $\mathbf{b}_\mathrm{h}$ are bias terms, and $\sigma(\cdot)$ is the activation function.

LSTM for Prediction

Each LSTM cell takes three inputs at each time slot: the feature observed in the current slot $\mathbf{k}_t$, the previous LSTM cell state (i.e., long-term memory unit) $\mathbf{L}_{t-1}$, and the previous hidden state (i.e., the short-term memory unit) $\mathbf{h}_{t-1}$. Then, LSTM model updates the output, hidden state, and cell state as follows.

$$\mathbf{f}_t = \sigma(\mathbf{W}_\mathrm{f}[\mathbf{h}_{t-1}, \mathbf{k}_t]) + \mathbf{b}_\mathrm{f}, \tag{4.21}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_\mathrm{i}[\mathbf{h}_{t-1}, \mathbf{k}_t] + \mathbf{b}_\mathrm{i}), \tag{4.22}$$

$$\tilde{\mathbf{L}}_t = \tanh(\mathbf{W}_\mathrm{k}[\mathbf{h}_{t-1}, \mathbf{k}_t] + \mathbf{b}_\mathrm{k}), \tag{4.23}$$

$$\mathbf{L}_t = \mathbf{f}_t \mathbf{L}_{t-1} + \mathbf{i}_t \tilde{\mathbf{L}}_t, \tag{4.24}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_\mathrm{o}[\mathbf{h}_{t-1}, \mathbf{k}_t] + \mathbf{b}_\mathrm{o}), \tag{4.25}$$

$$\mathbf{h}_t = \mathbf{o}_t \tanh(\mathbf{L}_t), \tag{4.26}$$

where $\mathbf{f}_t$ is the forget gate which decides what information will be kept from the previous cell state, $\mathbf{i}_t$ is the input gate which decides what information will be added to cell state of the network, $\mathbf{o}_t$ is the output gate of the LSTM unit, $\mathbf{h}_t$ is the new hidden state of the network, $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_k$, $\mathbf{W}_o$ are the coefficient matrices, $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_k, \mathbf{b}_o$ are the bias terms, and $\sigma(\cdot)$ is the activation function.

CNN for Prediction

CNN consists of convolution layer, pooling layer and fully connected layer. In convolution layer the feature representations of inputs (i.e., *feature maps*) are computed by applying element-wise convolution to the input with the kernel and then applying non-linear activation function to obtain the output of the layer. For input $\mathbf{k}_t$ and kernel ψ_t, the resulting feature at location (i, j) can be computed from the following steps,

$$\mathbf{Z}_t^{i,j} = \mathbf{W}_{\psi_t} * \mathbf{k}_t^{i,j} + \mathbf{b}_{\psi_t}, \tag{4.27}$$

$$\mathbf{Y}_t^{i,j} = \Phi(\mathbf{Z}_t^{i,j}), \tag{4.28}$$

where $\mathbf{W}_{\psi_t}$ and $\mathbf{b}_{\psi_t}$ are the weights and bias of the filter ψ_t, $\mathbf{k}_t^{i,j}$ is the subsection of the input centered at (i, j), $\Phi(\cdot)$ is the non-linear activation function, and '$*$' is the convolution operator. Then, the pooling layer is employed to decrease the number of features or resolution of the feature map after the convolution layer. The most used pooling operation is max-pooling [78] which computes a new feature map by traversing the output of convolution layer and calculating

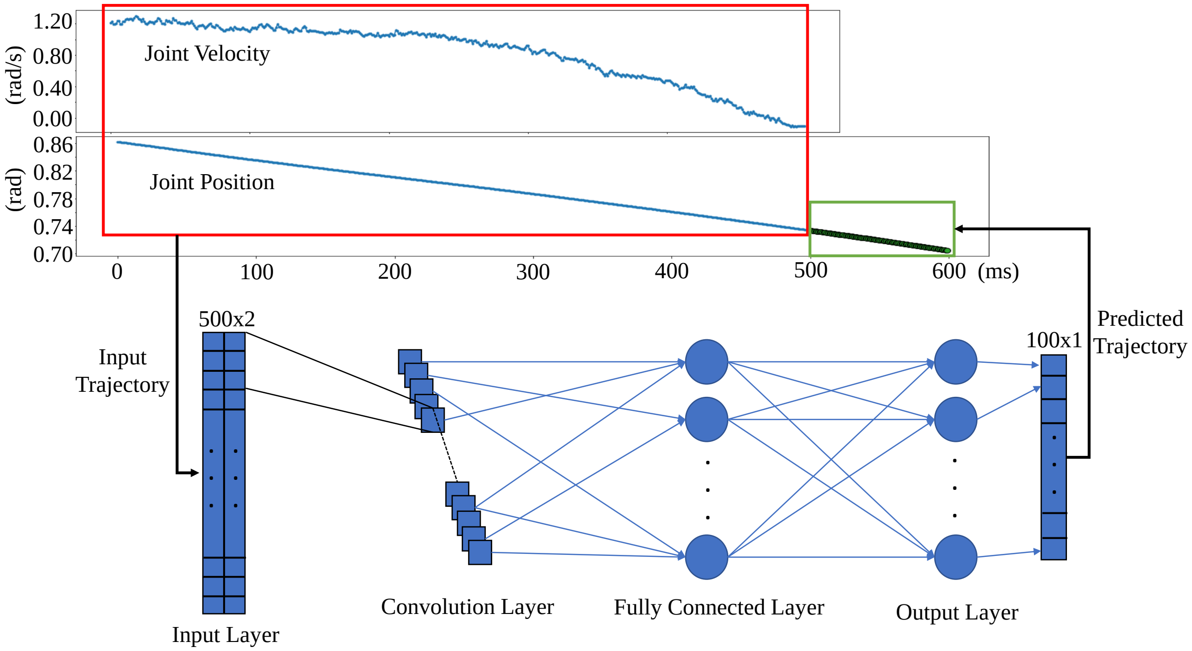


Figure 4.3: Illustration of trajectory prediction via CNN.

the maximum of each patch (i.e., subsection of the convolution output according to filter size). After computing feature maps with several convolution and pooling layers, computed features are flatten to be fed to fully connected layer (i.e., dense layer) as seen in the Fig. 4.3 which is trained to forecast the future trajectories.

4.4.4 Training and Testing of Prediction Algorithms

For all the three types of NNs, we use mean squared error, $MSE = \sum_t (\hat{\mathbf{k}}_t - \mathbf{k}_t)^2/s$, as the loss function since it is a differentiable function and eases the mathematical operations in optimizations throughout training process, where s is the number of samples. In addition, we use relative root mean squared error, $RRMSE(\%) = \frac{\sqrt{\sum_t (\hat{\mathbf{k}}_t - \mathbf{k}_t)^2/s)}}{\bar{\mathbf{k}_t}} \times 100$, to compare the performance of different prediction algorithms, where $\bar{\mathbf{k}_t}$ is the mean value of the observations. We set the maximum number of training epochs as 1000. To avoid over fitting, an early stopping criteria is adopted. Specifically, the training process is terminated and the latest model is saved if the model does not improve for 10 consecutive epochs. The hyper-parameters of prediction algorithms are listed in Table 4.3. To train the predictors, each input sample consists joint position and velocity in the past N_{in} time slots. Given the *history window* with N_{in} number of steps, $\mathbf{k}_{t-N_{\text{in}}:t} = \{\mathbf{k}_{t-N_{\text{in}}}, \mathbf{k}_{t-N_{\text{in}}+1}, \ldots, \mathbf{k}_t\}$, predictor is trained to predict the joint positions in the next N_{out} number of steps (i.e., *prediction window*) as shown in the example in Fig. 4.3. In this study, we consider multi-step prediction in which predictor predicts N_{out} steps at once since it is more accurate than single-step prediction [56, 138].

The testing results are provided in Table 4.4. Here, the dimension of the input is $N_{\text{in}} =$

Table 4.3: Hyper-Parameters of Prediction Algorithms

	RNN	**LSTM**	**CNN**
Number of layers	1	1	1
Number of cells in each layer	128 RNN cells	128 LSTM cells	128 Convolutional cells
Batch size	64	64	64
Optimizer	Adam	Adam	Adam
Loss function	MSE	MSE	MSE
Accuracy metric	RRMSE	RRMSE	RRMSE
Max training epochs	1000	1000	1000
Early stopping criteria	min validation loss, patience=10	min validation loss, patience=10	min validation loss, patience=10
Activation function	tanh	tanh	ReLu

Table 4.4: Performance of Different Predictors

Errors(%)	**LSTM**	**RNN**	**CNN**
Training RRMSE	0.8%	0.5%	0.07%
Validation RRMSE	0.8%	0.5%	0.08%
Test RRMSE	0.9%	0.6%	0.2%

500 ms and the length of prediction window is $N_{\text{out}} = 100$ ms. According to our results, CNN outperforms both LSTM and RNN. Therefore, we use CNN in our system and will be referring to CNN as the predictor in the rest part of the chapter. It is worth noting that the prediction accuracies of RNN, LSTM, and CNN depend on the datasets [65–68]. In general, RNN and LSTM outperform CNN in time-series data. But for some datasets, where the time-series data change suddenly, CNN can be better than RNN and LSTM [69].

4.4.5 Tradeoff between Prediction Error Probability and Prediction Horizon

In this subsection, we illustrate how to obtain the tradeoff between the prediction error probability and the prediction horizon. The goal is the estimate the prediction error probability $f_{\varepsilon_m^{\text{p}}}(T_m^{\text{p}}, \delta_m)$, in (4.2), i.e., the probability that the tracking error is greater than a required JND threshold δ_m, when the prediction horizon is T_m^{p}.

When the probability is extremely small, e.g., $f_{\varepsilon_m^{\text{p}}}(T_m^{\text{p}}, \delta_m) = 10^{-5}$, the real-world data set is not enough to obtain an accurate estimation. To overcome this difficulty, the synthetic trajectories generated by TimeGAN are used to estimate the prediction error probability. For every

Table 4.5: Hyper-Parameters and Training Results of FNN

Number of layers	4
Number of cells in each layer	64
Batch size	64
Activation	ReLu
Max training epochs	1000
Loss function	MSE
Accuracy metric	RRMSE
Training RRMSE	0.6%
Validation RRMSE	0.9%
Test RRMSE	0.9%

input trajectory $\mathbf{k}_{t-N_{\text{in}}:t} = \{k_{t-N_{\text{in}}}, k_{t-N_{\text{in}}+1}, ..., k_t\}$, there is a corresponding predicted trajectory $\hat{\mathbf{k}}_{t+1:t+N_{\text{out}}} = \{\hat{k}_{t+1}, \hat{k}_{t+2}, ..., \hat{k}_{t+N_{\text{out}}}\}$ and ground truth $\mathbf{k}_{t+1:t+N_{\text{out}}} = \{k_{t+1}, k_{t+2}, ..., k_{t+N_{\text{out}}}\}$. The prediction errors with different prediction horizon (from 1 time slot to N_{out} time slots) are given by $\mathbf{e}_{t+1:t+N_{\text{out}}} = \{e_{t+1}, e_{t+2}, ..., e_{t+N_{\text{out}}}\}$, where $e_t = \hat{k}_t - k_t$. With both experimental data and synthetic data, we can model the prediction error probability with different prediction horizons and JND thresholds using feedforward neural network (FNN) that takes the prediction horizon, T_m^{p}, and the JND threshold, δ_m, as its inputs and outputs the prediction error probability, i.e.,

$$f_{\varepsilon_m^{\text{p}}}(T_m^{\text{p}}, \delta_m) = FNN(T_m^{\text{p}}, \delta_m | \phi_m), \tag{4.29}$$

where ϕ_m is the training parameters of the FNN. Hyper-parameters and training results are provided in Table 4.5.

4.5 Efficient Resource Allocation with Task-oriented Prediction

Based on the communication system in Section 4.3 and the predictor in Section 4.4, we illustrate how to optimize resource allocation, W_m, and data rate, b_m, in the task-oriented prediction and communication co-design framework. To maximize the number of users, N, served by a BS, the optimization problem in (4.9), can be re-formulated as follows,

$$\max_{W_m, b_m} \quad N \tag{4.30}$$

$$\text{s.t.} \quad \sum_{m=1}^{N} W_m \leq W_{\max} \tag{4.30a}$$

$$D_m^{\text{e}} \leq D_m^{\max} \tag{4.30b}$$

$$\varepsilon_m^{\text{o}} \leq \varepsilon_{\max} \tag{4.30c}$$

$$(4.7), (4.12), (4.16), \text{ and } (4.29),$$

where (4.30a) is the constraint on the maximum available bandwidth, (4.30b) and (4.30c) are the constraints on the QoS requirement. As shown in (4.7), the overall error probability is bounded by

$$\begin{aligned}&\varepsilon_m^{\text{ub}}(W_m,b_m) \triangleq f_{\varepsilon_m^{\text{p}}}(D_m^{\text{ch}},\delta_m)\varepsilon_m^{\text{d}}(1-f_{\varepsilon_m^{\text{q}}}(D_m^{\text{th}}))\\&+f_{\varepsilon_m^{\text{p}}}(T_{\text{th}},\delta_m)(f_{\varepsilon_m^{\text{q}}}(D_m^{\text{th}})-f_{\varepsilon_m^{\text{q}}}(D_m^{\text{th}}+T_{\text{th}}))\\&+f_{\varepsilon_m^{\text{q}}}(D_m^{\text{th}}+T_{\text{th}}).\end{aligned}$$

In the 5G New Radio system, the expression of the decoding error probability, ε_m^{d}, is given by (4.12). Given the queuing system in Section 4.3, the relationship between the queuing delay bound and the queuing delay violation probability, $f_{\varepsilon_m^{\text{q}}}(\cdot)$, is derived in (4.16). With our predictor in Section 4.4, the tradeoff between the prediction error probability and the prediction horizon $f_{\varepsilon_m^{\text{p}}}(\cdot,\delta_m)$ is obtained in (4.29). From (4.12), (4.16), and (4.29), we can see that the upper bound is determined by the bandwidth allocation and the data rate, and is denoted by $\varepsilon_m^{\text{ub}}(W_m,b_m)$.

The number of constraints and the number of optimization variables are not deterministic in (4.30), i.e., they depend on the number of users, N. Thus, it is hard to derive closed-form solution of this problem. To overcome this difficulty, we decompose the problem into multiple single-user subproblems.

4.5.1 Single User Subproblem

To maximize the number of users that can be served with a given bandwidth, we turn to minimize the bandwidth that is required to guarantee the QoS of each user. As such, the optimization problem can be decomposed into independent single-user subproblems, i.e.,

$$\begin{aligned}&\min_{W_m,b_m} W_m \qquad (4.31)\\&\text{s.t. } \varepsilon_m^{\text{ub}}(W_m,b_m) \le \varepsilon_{\max}, (4.1), (4.7), (4.12), (4.16), \text{ and } (4.29).\end{aligned}$$

Please note that the objective function in (4.31) is special linear function: $1 \times W_m + 0$. Such a formation follows the standard form of optimization problem in [139]. We develop a two dimensional binary search algorithm to find the optimal solution of problem (4.31). We first fix the value of W_m and find the optimal data rate that minimizes the upper bound of the overall reliability in (4.7), i.e.,

$$\begin{aligned}&\min_{b_m} \varepsilon_m^{\text{ub}}(W_m,b_m) \qquad (4.32)\\&\text{s.t. } (4.1), (4.12), (4.16), \text{ and } (4.29),\end{aligned}$$

The optimal solution of the problem (4.32) is denoted by $b_m^*(W_m)$. Then, we use binary search to find the value of W_m that satisfies the following equation,

$$\varepsilon_m^{\text{ub}}(W_m, b_m^*(W_m)) = \varepsilon_{\max}. \tag{4.33}$$

We denote the solution of (4.33) by W_m^*.

In the rest part of this section, we prove that W_m^* and $b_m^*(W_m^*)$ are the optimal solution of problem (4.31). To prove this, we need two lemmas.

Lemma 3: For given W_m, $\varepsilon_m^{\text{ub}}(W_m, b_m)$ first decreases and then increases with b_m.

We first validate that $\varepsilon_m^{\text{ub}}(W_m, b_m)$ first decreases and then increases with b_m in two asymptotic scenarios: 1) b_m is sufficiently small such that the decoding error probability is close to zero; 2) b_m is sufficiently large such that the queuing delay violation probability is close to zero.

In the first scenario that b_m is very small, the queuing delay violation probability $f_{\varepsilon_m^{\text{q}}}(.)$ is large and, the decoding error probability ε_m^{d} is close to zero. Then, $\varepsilon_m^{\text{ub}}(W_m, b_m)$ can be simplified as $f_{\varepsilon_m^{\text{p}}}(T_{\text{th}}, \delta_m)(f_{\varepsilon_m^{\text{q}}}(D_m^{\text{th}}) - f_{\varepsilon_m^{\text{q}}}(D_m^{\text{th}} + T_{\text{th}})) + f_{\varepsilon_m^{\text{q}}}(D_m^{\text{th}} + T_{\text{th}})$. Since $f_{\varepsilon_m^{\text{q}}}(.)$ is a decreasing function of b_m according to Lemma 1, $\varepsilon_m^{\text{ub}}(W_m, b_m)$ decreases with b_m.

When b_m is sufficiently large, the queuing delay violation probability $f_{\varepsilon_m^{\text{q}}}(.)$ is close to zero, but the decoding error probability ε_m^{d} is large. Then, $\varepsilon_m^{\text{ub}}(W_m, b_m)$ is dominated by ε_m^{d}, which increases with b_m according to Lemma 2. In this case, $\varepsilon_m^{\text{ub}}(W_m, b_m)$ increases with b_m.

In non-asymptotic scenarios, it is difficult to prove Lemma 3. We will validate Lemma 3 with numerical results.

Lemma 4: $\varepsilon_m^{\text{ub}}(W_m, b_m^*(W_m))$ decreases with W_m.

Proof: From (4.7), (4.16), and (4.12), we can see that given the value of b_m, only the decoding error probability decreases with the value of W_m, and all the other terms remain constant. If $W_m < \tilde{W}_m$, then, $\varepsilon_m^{\text{ub}}(W_m, b_m^*(W_m)) > \varepsilon_m^{\text{ub}}(\tilde{W}_m, b_m^*(W_m))$. According to the definition of $b_m^*(\tilde{W}_m)$, it is the solution of (4.31). Given bandwidth $\tilde{W}_m$, the optimal value of b_m that minimize the upper bound of the decoding error probability is $b_m^*(\tilde{W}_m)$. In other words, $\varepsilon_m^{\text{ub}}(\tilde{W}_m, b_m^*(W_m)) > \varepsilon_m^{\text{ub}}(\tilde{W}_m, b_m^*(\tilde{W}_m))$. Then, we have $\varepsilon_m^{\text{ub}}(W_m, b_m^*(W_m)) > \varepsilon_m^{\text{ub}}(\tilde{W}_m, b_m^*(\tilde{W}_m))$.

Lemma 4 indicates that the optimal solution of problem (4.31) should satisfy (4.33) and the optimal bandwidth can be obtained by binary search. To find the optimal b_m for a given W_m, we can also use binary search as indicated by Lemma 3. Therefore, the optimal solution of problem (4.31) can be obtained via the two dimensional binary search given in Algorithm 2.

4.6 Evaluation of the Proposed Framework

In this section, we evaluate the proposed task-oriented prediction and communication co-design framework with numerical results where Table 4.6 provides the parameter settings. We consider the path loss model $10\log_{10}(\alpha_m) = -128.1 - 36.7\log_{10}(d_m)$, where $d_m = 200$ m is the distance

Algorithm 2 Algorithm for solving (4.31)

```
Input: Delay requirement $D_m^{\max}$, reliability requirement $\varepsilon_m^{\max}$, threshold for queuing delay $D_m^{\rm th}$, prediction horizon threshold $T_{\rm th}$, channel coherence time $D_m^{\rm ch}$, average packet arrival rate $\lambda_m$, JND threshold $\delta_m$, initial bandwidth $W_0$, maximum bandwidth $W_{\max}$, large-scale channel gain $\alpha_m$, small-scale channel gain $g_m$, transmit power $P_m$, single sided noise spectral density $N_0$, initial number of bits $b_0$, maximum number of bits $b_{\max}$.
Output: Optimal $W_m^*$, $b_m^*(W_m^*)$ to ensure URLLC QoS requirements for $m^{th}$ task.
    Initialisation : $W_{\rm L} = W_0$, $W_{\rm R} = W_{\max}$, $W_{\rm mid} = (W_{\rm L} + W_{\rm R})/2$.
Binary search $b_m$ in $[b_0, b_{\max}]$ and obtain $\varepsilon_m^{\rm ub}(W_{\rm mid}, b_m^*(W_{\rm mid}))$
while $|\varepsilon_m^{\rm ub}(W_{\rm mid}, b_m^*(W_{\rm mid})) - \varepsilon_m^{\max}| > (\varepsilon_m^{\max})^2$ and $W_{\rm L} < W_{\rm R}$ do
    if $\varepsilon_m^{\rm ub}(W_{\rm mid}, b_m^*(W_{\rm mid})) \le \varepsilon_m^{\max}$ then
        $W_m^* = W_{\rm mid}$
        $W_{\rm R} = W_{\rm mid}$
        $W_{\rm mid} = (W_{\rm L} + W_{\rm R})/2$
    else
        $W_m^* = W_{\rm mid}$
        $W_{\rm L} = W_{\rm mid}$
        $W_{\rm mid} = (W_{\rm L} + W_{\rm R})/2$
    end if
    Binary search $b_m$ in $[b_0, b_{\max}]$ and obtain $\varepsilon_m^{\rm ub}(W_{\rm mid}, b_m^*(W_{\rm mid}))$
end while
return $W_m^*$, $b_m^*(W_m^*)$
```

between the BS and the receiver [4]. The overall error probability, ε_m^o, is obtained from (4.7).

In the following, we first illustrate the tradeoff between prediction error probability and prediction horizon for different JND thresholds. Then, we consider the single-user scenario to illustrate the optimality of the Algorithm 2 as well as we compare the two frameworks, where the predictors are either deployed at transmitter sides or at the receiver sides. Finally, we consider the multi-user scenario to evaluate the performance gain of the proposed framework in terms of the required bandwidth for each user or the maximum number of users that can be served.

Table 4.6: Numerical Values of Parameters for Overall Error [4, 5].

Parameter	**Value**
λ_m, average packet arrival rate	100 packets/s
P_m, maximum transmit power	23 dBm
N_0, single sided noise spectral density	-144 dBm/Hz
$D_m^{\rm r}$, core network and backhaul delay	10 ms
$D_m^{\rm t}$, transmission delay	0.5 ms
$D_m^{\rm ch}$, channel coherence time	10 ms
$T_{\rm th}$, prediction horizon threshold	50 ms
$D_m^{\rm th}$, queuing delay threshold	$\max\{D_m^{\max} - D_m^{\rm t} - D_m^{\rm r}, 0\}$

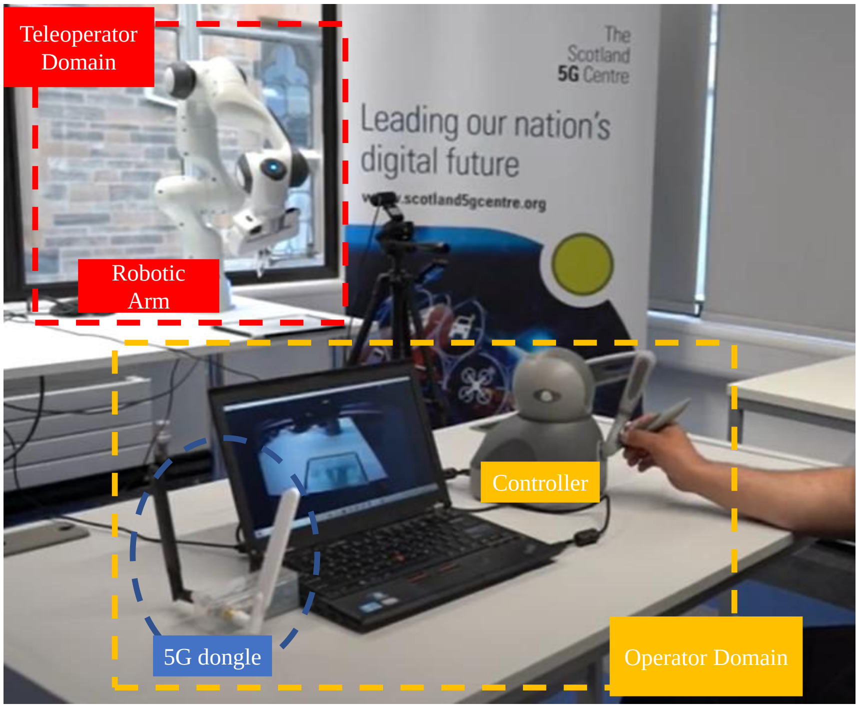


Figure 4.4: Dataset collection from 5G-enabled Teleoperation Prototype

4.6.1 Prediction Error Probability and Prediction Horizon Tradeoff

In Fig. 4.5, we compare a real-world trajectory segment with a synthetic trajectory segment. The results show that real and synthetic trajectories follow similar trend but with different values. This is reasonable since the motivation of generating synthetic data is for increasing the diversity of the dataset rather than generating the same data. Therefore, it is expected to have some differences between synthetic data and real data. We use both synthetic and real trajectories to obtain the tradeoff between the prediction error probability and the prediction horizon for different JND thresholds. In Fig. 4.6, we provide the tradeoff between the prediction error probability and the prediction horizon. The results validate our assumption that the prediction error probability increases with the prediction horizon T_m^p and decreases with the required JND threshold δ_m.

4.6.2 Single-user Scenarios

In single-user scenario, we consider two categories of tasks namely critical and non-critical tasks (see Fig. 4.2 where task 1 and task 2 are non-critical, and task 3 is critical), whose JND thresholds are $\delta_m = 0.1\%$, and $\delta_m = 1\%$, respectively. For fair comparison, we consider both tasks have the same value of delay requirement $D_m^{\max}$ as well as reliability requirement $\varepsilon_m^{\max}$.

Fig. 4.7 illustrates the trade-off between the required delay bound and the overall error probability with different JND thresholds. Here, the allocated bandwidth W_m is 140 kHz and the packet size b_m is 256 bits. The results show that the overall error probability is constant

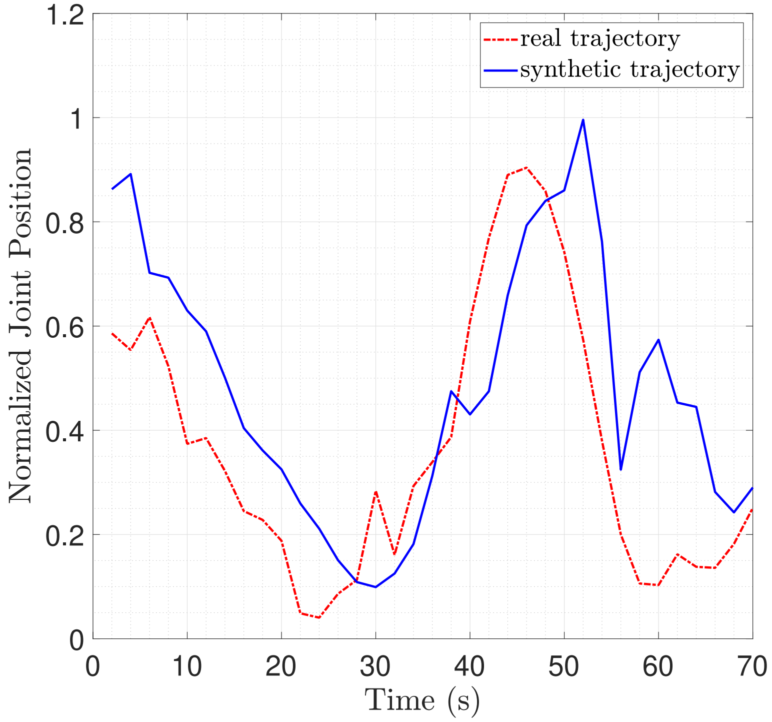


Figure 4.5: Real and synthetic trajectory comparison to illustrate the quality of generated data.

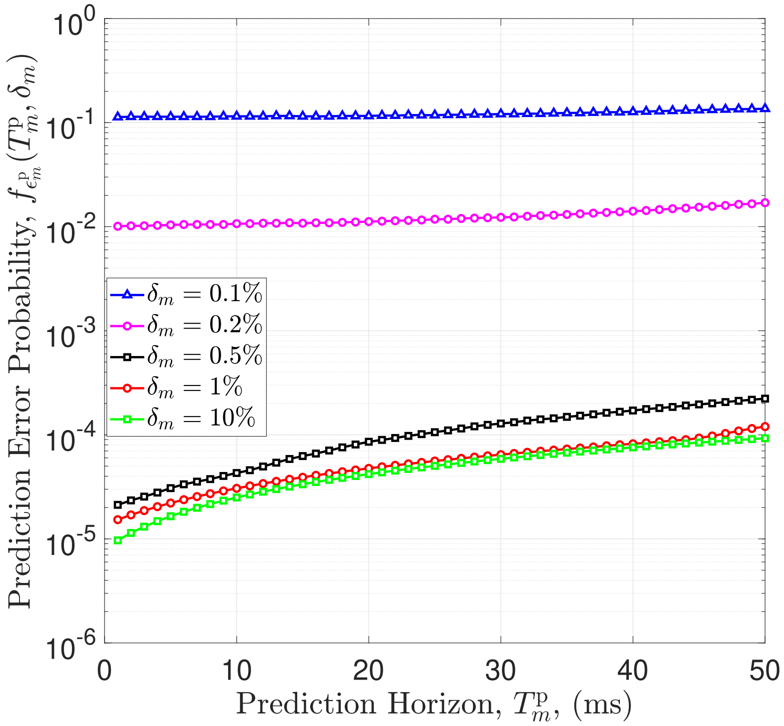


Figure 4.6: Prediction error probability versus prediction horizon for different JND thresholds. Prediction error probability curves for different JND thresholds are obtained from 1.7×10^7 real-world data samples and 2×10^{11} synthetic data samples which correspond to 6.7×10^8 history and prediction window pairs.

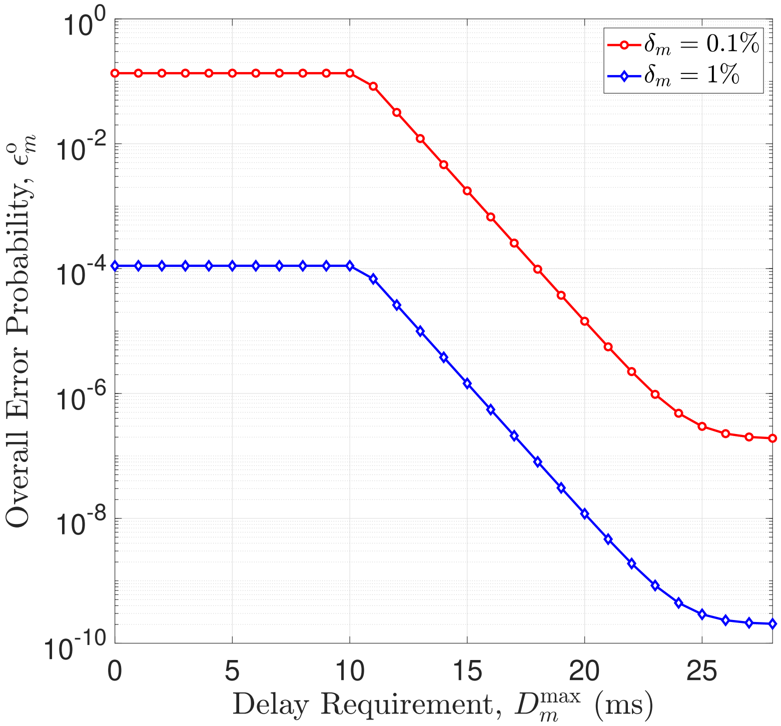
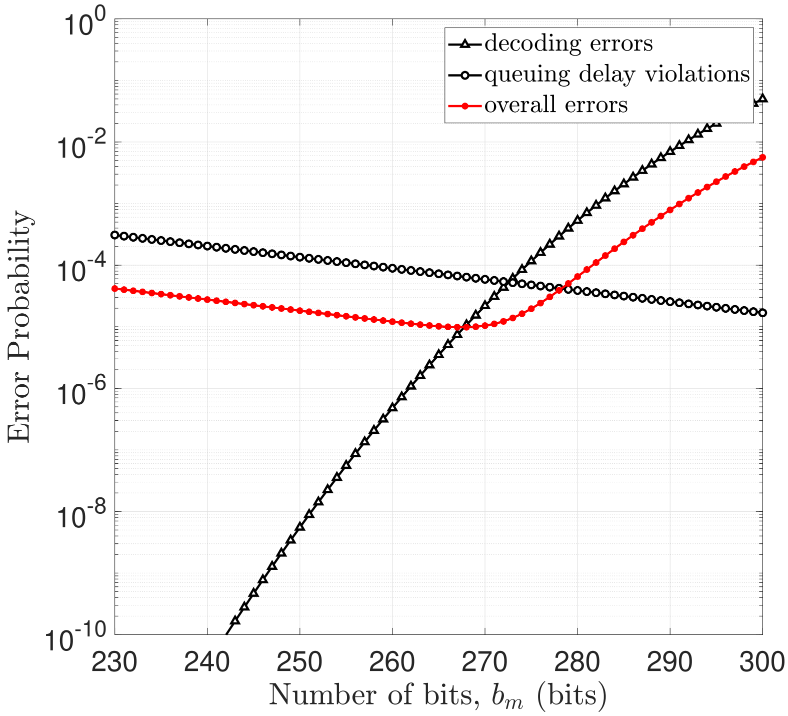


Figure 4.7: Overall error probability versus delay requirement, where $W_m = 140$ kHz and $b_m = 256$ bits.

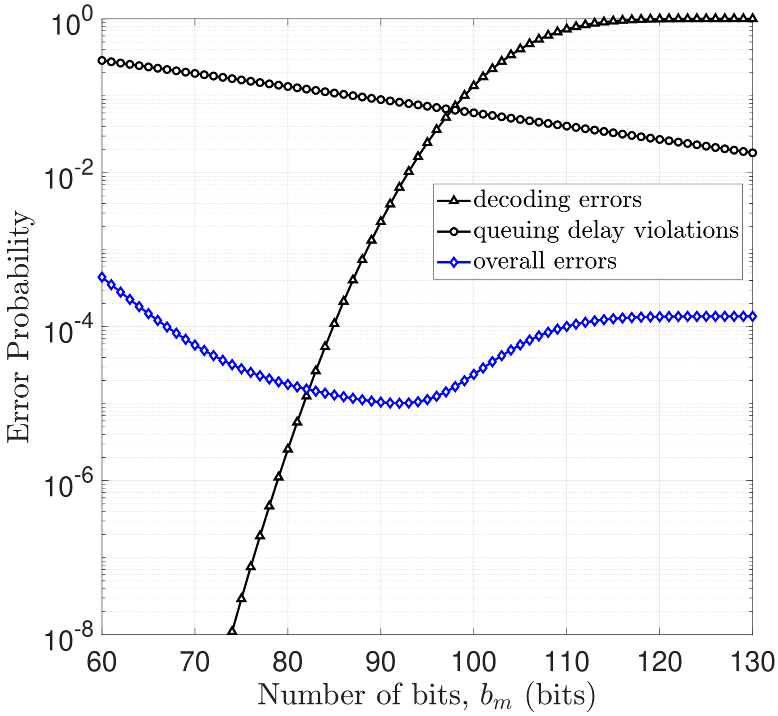
and equal to prediction error probability when delay requirement smaller than or equal to the communication delay, i.e., $D_m^{\rm max} \le D_m^{\rm c}$. Then, it decreases rapidly as the delay requirement $D_m^{\rm max}$ grows from 10 to about 25 ms. This is reasonable because the queuing delay violation probability $f_{\varepsilon_m^{\rm q}}(D_m^{\rm th})$ is the dominant factor in the region $10 < D_m^{\rm max} < 25$ ms. When the delay requirement is larger than 25 ms, both curves are nearly constant with $D_m^{\rm max}$. This is because $f_{\varepsilon_m^{\rm p}}(D_m^{\rm ch}, \delta_m)\varepsilon_m^{\rm d}$ becomes the dominant factor in (4.7) and does not change with the required delay bound. Furthermore, the error probability of the critical task is much higher than that of the non-critical task. This implies that the bandwidth consumption of different task will be significantly different if they require the same error probability.

To validate Lemmas 1-3, we provide the relationship between different error probabilities in Fig. 4.8. The results show that the queuing delay bound violation probability, $f_{\varepsilon_m^{\rm q}}(D_m^{\rm th})$, decreases with number of bits and the decoding error probability, $\varepsilon_m^{\rm d}$, increases with number of bits. These two curves validate Lemma 1 and Lemma 2, respectively. For both critical and non-critical tasks, the overall error probability, $\varepsilon_m^{\rm o}$, first decreases and then increases with number of bits. Moreover, when the queuing delay violations are larger than the decoding errors, the overall error is dominated by the queuing delay violation. When the decoding errors are larger than the queuing delay violations, the overall error is dominated by the decoding errors. These observations are consistent with Lemma 3.

In Fig. 4.9, we compare error probabilities of critical and non-critical tasks under different



(a) Error probability versus number of bits, where $\delta_m = 0.1\%$.



(b) Error probability versus number of bits, where $\delta_m = 1\%$.

Figure 4.8: Error probabilities versus number of bits, where $D_m^{\max} = 20$ ms and $\varepsilon_m^{\max} = 10^{-5}$.

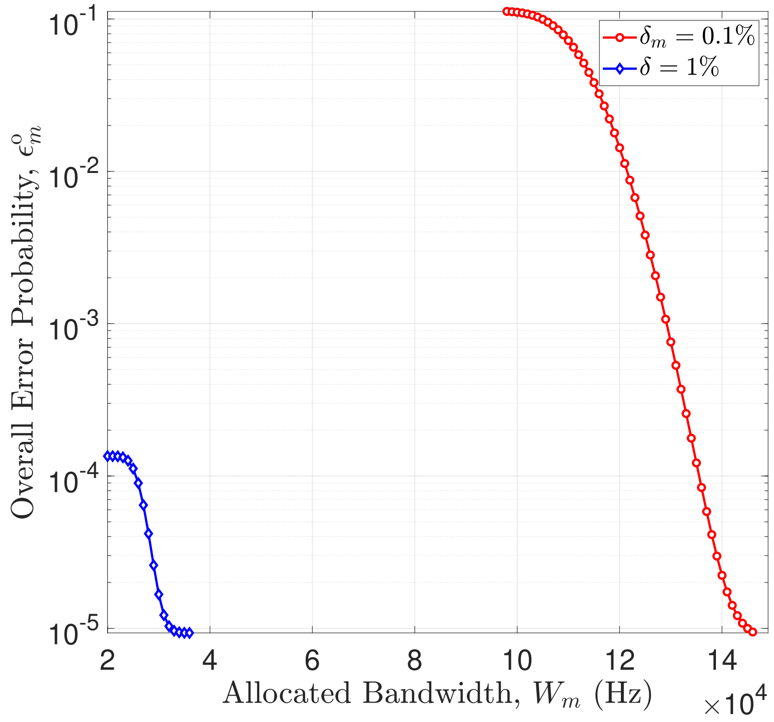
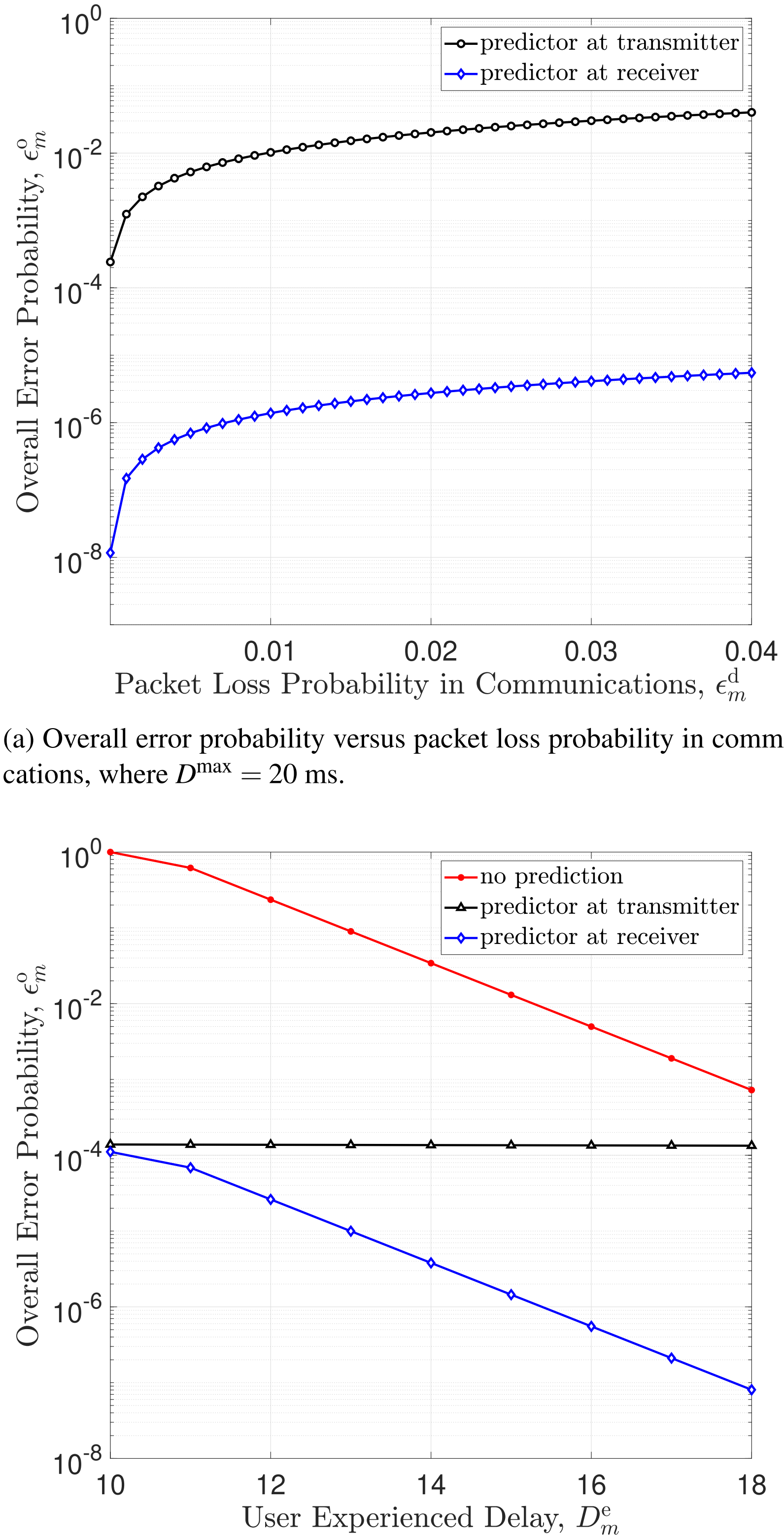


Figure 4.9: Error probabilities versus allocated bandwidth, where $D_m^{\max} = 20$ ms and $\varepsilon_m^{\max} = 10^{-5}$.

values of allocated bandwidth, W_m. Here, the delay requirement is 20 ms and the reliability requirement is 10^{-5}. From the figure, the overall error probability decreases with increasing bandwidth in both critical and non-critical cases which is consistent with Lemma 4. The results in Figs. 4.8 and 4.9 indicate that the overall error probability achieved by the proposed framework for critical case is $\varepsilon_m^{\text{ub}}(W_m^*, b_m^*(W_m^*)) = 1.00 \times 10^{-5}$ with $W_m^* = 145.24$ kHz, $b_m^*(W_m^*) = 268$ bits and for non-critical case is $\varepsilon_m^{\text{ub}}(W_m^*, b_m^*(W_m^*)) = 1.00 \times 10^{-5}$ with $W_m^* = 32.19$ kHz, $b_m^*(W_m^*) = 92$ bits.

In the existing literature, the predictor is either deployed at the transmitter [4, 85, 87, 89] or at the receiver [86, 88]. Both deployment strategies have advantages and disadvantages. If the predictor is deployed at the transmitter, the advantage is that the historical information used in the prediction algorithm is accurate. The disadvantage is that either a prediction error or a packet loss in communication may result in a JND violation. If the predictor is deployed at the receiver, it can adjust the prediction horizon according to the communication delays of different packets. If the communication delay is satisfactory, there is no need to do any prediction. In this way, a JND violation happens when both the communication and the prediction fail. The disadvantage of this framework is that the historical information used in the prediction algorithm may not be accurate, because some packets are lost or severely delayed. Nevertheless, the effects of deployment strategy hasn't been investigated and deserve further analyses.

Here, we provide comparison between two deployment strategies to highlight the differences



(a) Overall error probability versus packet loss probability in communications, where $D^{\max} = 20$ ms.

(b) Overall error probability versus user experienced delay, where $\varepsilon^{\mathrm{d}} = 10^{-5}$.

Figure 4.10: Predictor at transmitter versus predictor at receiver where, $W_m = 140\,\mathrm{kHz}$, $\delta_m = 1\%$.

and deliver some insights for prediction and communication co-design research. Fig. 4.10 compares the overall system reliability of two deployment strategies under different values of packet loss probabilities in communications (Fig. 4.10(a)) and user experienced delay (Fig. 4.10(b)). From Fig. 4.10(a), deploying predictor at receiver achieves better overall reliability with identical communication conditions. The reason behind this result is that if the predictor at the transmitter, then the overall system reliability dominated with least reliable system (i.e. either prediction system or communication system) since predicted future trajectories transmitted over communication system. However, if predictor at receiver, then it becomes compensation mechanism for communication system where lost packets can be predicted with cost of prediction errors. From Fig. 4.10(b), deploying predictor at receiver can achieve better delay and reliability tradeoff. Both strategies can compensate or reduce user experienced delay. However, deploying predictor at receiver can achieve similar user experienced delay with higher reliability. The insight is that if the communication system reliability is high, i.e., packet loss probability less than 10^{-5}, both strategies are suitable to reduce user experienced delay with accurate predictor. However, when the communication system is not reliable, i.e. packet loss probability greater than 10^{-5}, we can achieve the URLLC QoS requirements by only deploying the predictor at the receiver.

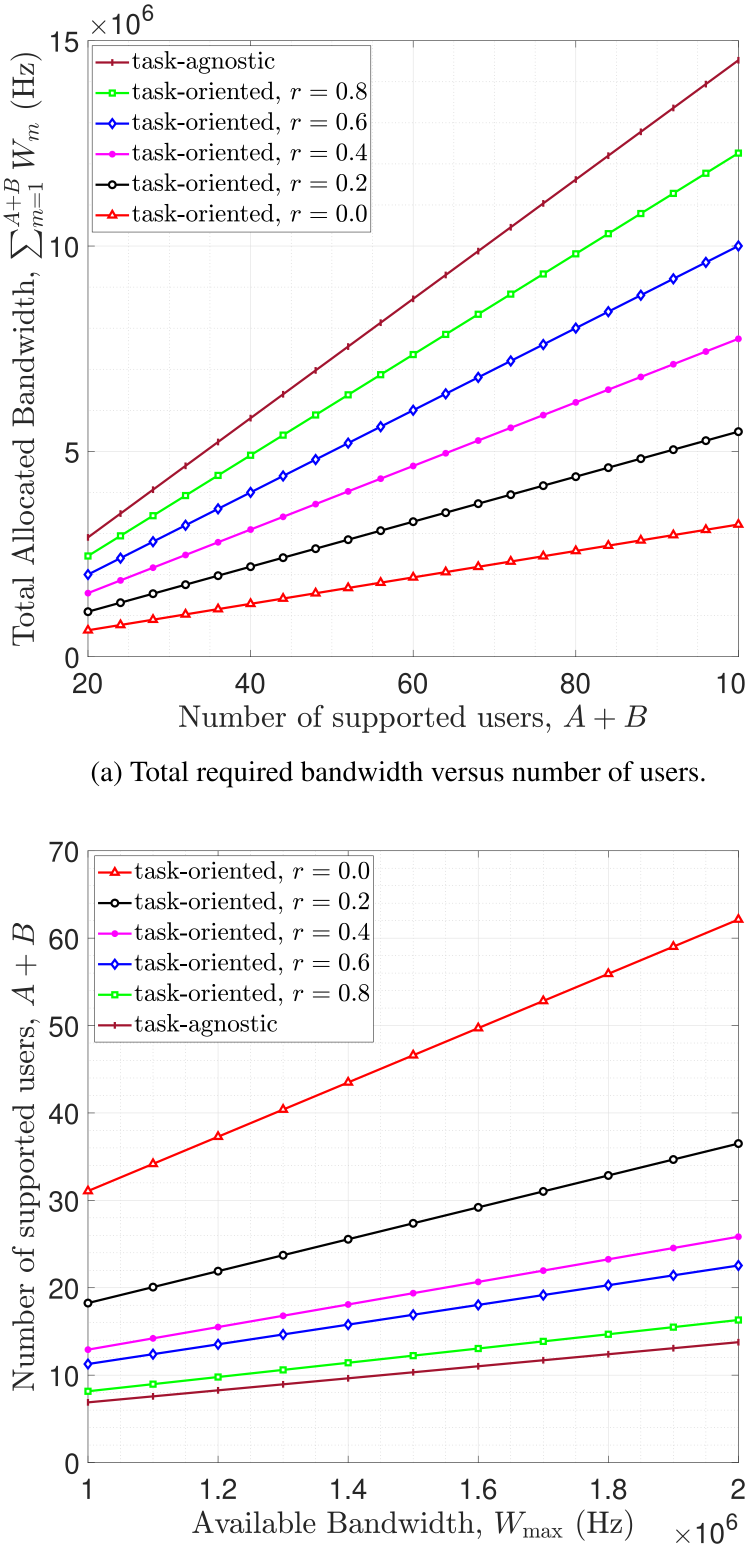
4.6.3 Multi-users Scenarios

In multi-user scenarios, we compare the proposed task-oriented prediction and communication framework with the task-agnostic prediction and communication benchmark. Similar to single-user scenario, we assume two types of tasks and used the results from single-user scenario for the minimum required bandwidth and optimal packet rate. Furthermore, we denote critical task ratio to total number of tasks as $r = \frac{A}{A+B}, A \geq 0, B \geq 0$ where A is the number of critical tasks and B is the number of non-critical tasks.

Fig. 4.11 compares the resource utilization of the proposed task-oriented prediction and communication framework and the benchmark that is task-agnostic under different values of available resources and critical task ratios. From the figures, the proposed approach achieves more efficient use of resources with up to 77.80% resource saving. This is because the proposed approach allocates resources according to JND thresholds of different tasks.

4.7 Conclusions

In this chapter, we proposed the task-oriented prediction and communication co-design framework to increase the wireless resource utilization efficiency for haptic communications, where low latency and high reliability performance are required. The basic tradeoff between resource utilization efficiency and overall reliability are provided. For predictions, we see that real-world data are not enough to achieve URLLC level reliability. To address this issue, we de-



(a) Total required bandwidth versus number of users.

(b) Number of supported users versus maximum available bandwidth.

Figure 4.11: The proposed task-oriented design versus task-agnostic design in terms of resource utilization.

ploy TimeGAN to generate realistic synthetic data which we use to reveal experimental tradeoff between prediction reliability and prediction horizon for different JND thresholds. We analysed prediction and communication systems and studied their relationship to reveal the tradeoff between wireless resources and reliability in proposed task-oriented communication and prediction co-design. We considered the teleoperation scenario via 5G New Radio and demonstrated a design example using the proposed framework. We formulated a joint optimization problem to maximize the number of users in a communication system by jointly optimizing communication data rate and task-dependent JND threshold. Numerical results show that the proposed approach can reduce the wireless resource consumption by 77.80% compared with the benchmark that is task-agnostic.

Chapter 5

Intelligent Mode-Switching Framework

5.1 Introduction

The typical teleoperation[1], in which a human operator manually controls a remote teleoperator, is difficult and requires an excessive amount of time to reach the level of expertise due to the limited perception, high communication latency, and limited DoF at the operator side [104]. In addition, the operator's mental and physical demands are relatively high in conventional teleoperation [105] which makes the process cumbersome for the operator since s/he needs to concentrate on every single detail in the control process instead of focusing on the task at hand.

To address these issues, autonomous teleoperation systems are proposed and extensively studied in the existing literature [99, 100, 102–106, 108–111]. The basic idea of autonomous teleoperation is to predict user intention and execute some parts of the task autonomously to decrease the demand on the operator and increase the task completion rate. Existing studies can be categorized based on user intention recognition techniques, task performance metrics, and decision-making algorithms. User intention recognition is performed either by model based methods [99, 103, 111], data-driven methods [104–106, 108–110] or combinations of both types of methods [100]. It is worth noting that user intention recognition accuracy varies between 20% and 95%, which was not considered in some references. Performance metrics highly depend on specific tasks. To better measure the performance, both objective metrics (e.g., task success rate and task completion time) and subjective metrics (e.g., operators' mental and physical demands) are considered.

In some existing studies [104, 105], decision-making for mode-switching is assumed to be done by the operator. This approach brings an extra DoF to be controlled by the operator and generally introduces extra mental demand [100]. Hence, developing a mode-switching policy that works autonomously and seamlessly is in urgent need. Furthermore, communication imperfections and resource limitations are the main bottlenecks for long-distance teleopera-

[1] Teleoperation covers any remote operation done by a human operator by controlling a remote robot (which can be a manipulator robot, mobile robot, UGV, UAV, etc.) over a communication network.

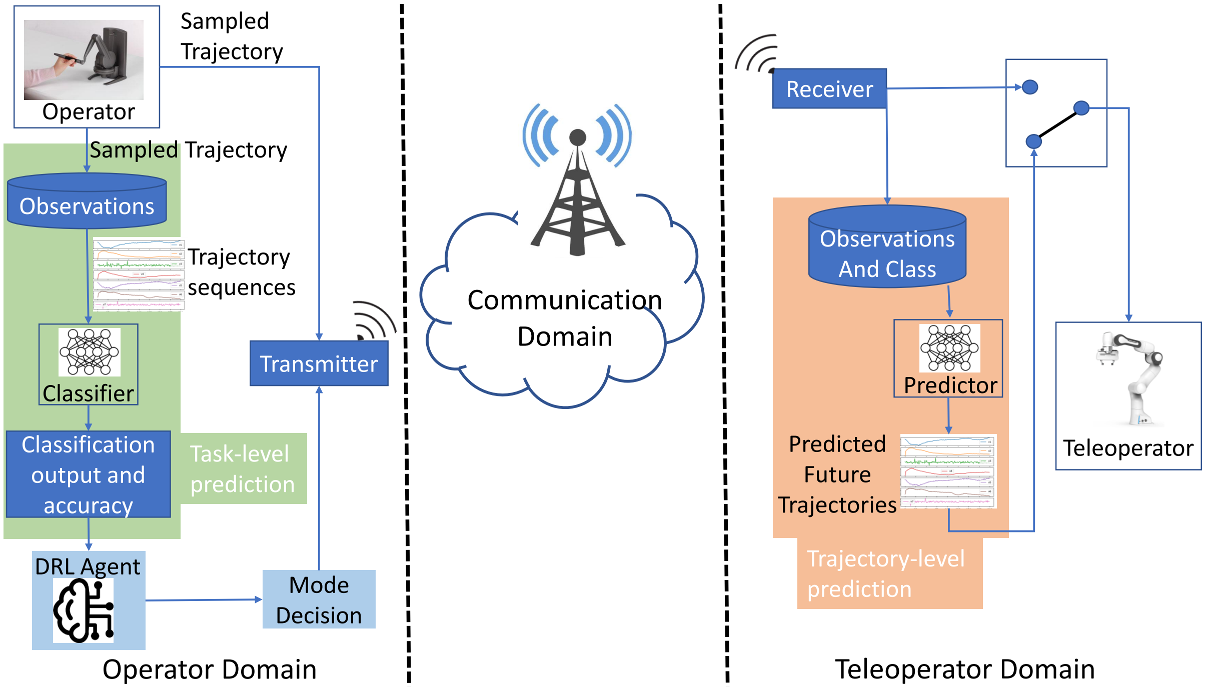


Figure 5.1: Intelligent Mode Switching Framework for Long Distance Teleoperation

tion. Nevertheless, how a communication system can be designed to accommodate autonomous teleoperation efficiently is an open question. Therefore, we need a new design methodology to jointly optimize autonomous teleoperation systems and communication systems to achieve seamless operation with low physical and mental operator demand.

With the aforementioned considerations, we propose an intelligent mode-switching framework for long-distance teleoperation. We design a general framework by jointly considering the communication system and the decision-making system. User intention recognition is done at the operator side with a CNN-based classification model considering that the CNN-based models perform better in the existing literature compared to other model-based and data-driven classification algorithms for user intention recognition [99–111]. Based on user intentions, a DRL agent is trained and deployed on the operator side to switch between the autonomous and teleoperation modes. A real-world data set is collected from our teleoperation testbed to train both user intention recognition and DRL algorithms. Our results show that the proposed framework can improve task completion probability tradeoffs with up to 50% communication load reduction compared to conventional teleoperation.

5.2 System Model

The proposed framework consists of three main domains, namely operator domain, communication domain, and teleoperator domain, as shown in Fig. 5.1, and time is discretized into slots. A human operator uses a controller with a haptic interface to control a remote teleoperator. At time slot t, m-th operator's control commands, $K_m(t)$, are sampled and transmitted. Sampled control commands are also saved in the observation memory to be used by a local task-level prediction algorithm, which attempts to recognize the user's intention in every time slot using the history of observations. Prediction output, $L_m(t) = \{l_m^1(t), ..., l_m^N(t)\}^{\mathrm{T}}$, is a probability vector obtained from task-level prediction algorithm, where $l_m^n(t) \in [0,1]$ is the probability that the user intends to execute task n in the t-th time slot. $L_m(t)$ is used by a DRL agent to make a mode-switching decision, and we will introduce the intention recognition algorithm and the DRL algorithms in the next section.

In teleoperation mode, the system works as a conventional teleoperation system in which the controller and the teleoperator exchange packets over the communication domain. In autonomous mode, there is no real-time packet exchange between the operator and the teleoperator. Task-level predictor works as an error detection mechanism in autonomous mode, which enables recovery from wrong user intention recognition. In case of prediction error, the system switches back to the teleoperation mode, i.e., gives control to the operator, and continues to monitor prediction output to recognize user intention for possible switching to autonomous mode again. Teleoperator is also equipped with a trajectory-level predictor that is used in autonomous mode. In other words, the teleoperator predicts future trajectories and finishes the task autonomously. It is worth noting that the computation time for user intention recognition and error detection is less than 1 ms considering execution time of a CNN-based classification model [140]. Therefore, we assume that it is negligible compared to long-distance communication delay.

5.2.1 Communication Load

We define the communication load as the average data rate required for a task, D_m, which can be given as

$$D_m = \frac{d_m}{Z_m} b_m \; (bits/slot), \tag{5.1}$$

where d_m is the number of time slots the controller is in the teleoperation mode, Z_m is the number of time slots required to finish the task and b_m is the number of bits transmitted in each time slot. We consider 5G NR as an example communication system. For teleoperation applications, the packet size is small and the transmission duration of each packet is much smaller than the channel coherence time. In other words, the channel fading coefficient remains constant over the transmission duration [40]. In such a scenario, the maximal achievable rate can be accurately approximated as [130]

$$B_m \approx C_m - \sqrt{\frac{V_m}{\tau_m W_m}} Q^{-1}(\varepsilon_m^{\mathrm{d}}) \text{ (bits/s/Hz)}, \tag{5.2}$$

where $C_m = \log(1+\gamma_m)$ is the Shannon capacity, $\gamma_m = \frac{\alpha_m g_m P_m}{N_0 W_m}$ is the received SNR at the base station, α_m is the large-scale channel gain, g_m denotes the small-scale channel gain, P_m denotes the transmit power, N_0 is the single sided noise spectral density, $V_m = \log(e)^2 \left[1 - \frac{1}{(1+\gamma_m)^2}\right]$ is the channel dispersion, $\tau_m W_m$ is the blocklength, τ_m is the transmission duration, W_m is the bandwidth, $Q^{-1}(.)$ is the inverse of the Gaussian Q-function, $Q(x) \triangleq \int_x^\infty \frac{1}{2\pi} e^{-t^2/2} dt$, and $\varepsilon_m^{\mathrm{d}}$ is the decoding error probability. Given that $b_m = \tau_m W_m B_m$, communication load, D_m becomes,

$$D_m \approx \frac{d_m \tau_m W_m}{Z_m} \left[C_m - \sqrt{\frac{V_m}{\tau_m W_m}} Q^{-1}(\varepsilon_m^{\mathrm{d}}) \right] \; (bits/slot). \tag{5.3}$$

5.2.2 Task Completion Probability

The task completion probability can be analyzed in teleoperation mode and autonomous mode. We denote the probability that the system stays in the teleoperation mode and the autonomous mode by P^{t} and P^{a}, respectively, where $P^{\mathrm{t}} + P^{\mathrm{a}} = 1$.

- **Case 1:** *Teleoperation Mode*

  In the teleoperation mode, the task completion probability highly depends on the communication reliability and the operator's experience since real-time packet exchange takes place. The reliability of a communication system can be measured by the decoding error probability, $\varepsilon_m^{\mathrm{d}}$, and the queuing delay violation probability, $\varepsilon_m^{\mathrm{q}}$. To take operator experience into account, we denote the operator experience coefficient by, $\rho_m \in [0,1]$. In this case, task completion probability becomes the multiplication of the communication reliability and user experience coefficient. In other words, any communication imperfections such as decoding error or queuing delay violation, and operator errors can cause task failure. Hence, the task completion probability in the teleoperation mode can be expressed with the following relationship between the decoding error probability, $\varepsilon_m^{\mathrm{d}}$, the queuing delay violation probability, $\varepsilon_m^{\mathrm{q}}$ and the operator experience coefficient ρ_m.

$$\begin{aligned} P_m^{\mu} &= f_m^{\mu}(\varepsilon_m^{\mathrm{q}}, \varepsilon_m^{\mathrm{d}}, \rho_m) \\ &= (1-\varepsilon_m^{\mathrm{q}})(1-\varepsilon_m^{\mathrm{d}})\rho_m \end{aligned} \tag{5.4}$$

- **Case 2:** *Autonomous Mode*

  In the autonomous mode, task completion probability depends on the intention recognition algorithm, the error detection algorithm, and the trajectory prediction algorithm. To take these into account, we define the task-level prediction error probability, $\varepsilon_m^{\mathrm{c}}$, the error detection system failure probability, $\varepsilon_m^{\mathrm{f}}$, and the trajectory-level prediction error probability, $\varepsilon_m^{\mathrm{t}}$. In this case, task completion probability becomes the multiplication of the task-level prediction reliability, error detection system reliability and trajectory-level prediction reliability. In other words, prediction error on trajectory-level prediction can cause task failure. Similarly, prediction error in task-level can cause task failure if it cannot be captured by error detection system. Hence, task completion probability in the autonomous mode can be expressed with the following relationship between the task-level prediction error probability, $\varepsilon_m^{\mathrm{c}}$, the error detection system failure probability, $\varepsilon_m^{\mathrm{f}}$, and the trajectory-level prediction error probability, $\varepsilon_m^{\mathrm{t}}$.

$$\begin{aligned} P_m^{\sigma} &= f_m^{\sigma}(\varepsilon_m^{\mathrm{c}}, \varepsilon_m^{\mathrm{f}}, \varepsilon_m^{\mathrm{t}}) \\ &= ((1-\varepsilon_m^{\mathrm{c}}) + (\varepsilon_m^{\mathrm{c}}(1-\varepsilon_m^{\mathrm{f}})))(1-\varepsilon_m^{\mathrm{t}}) \end{aligned} \tag{5.5}$$

Then, the overall task completion probability can be given as,

$$\begin{aligned} P_m^{\mathrm{o}} &= P^{\mathrm{t}} P_m^{\mu} + P^{\mathrm{a}} P_m^{\sigma}, \\ &= P^{\mathrm{t}} f_m^{\mu}(\varepsilon_m^{\mathrm{q}}, \varepsilon_m^{\mathrm{d}}, \rho_m) + P^{\mathrm{a}} f_m^{\sigma}(\varepsilon_m^{\mathrm{c}}, \varepsilon_m^{\mathrm{f}}, \varepsilon_m^{\mathrm{t}}). \end{aligned} \tag{5.6}$$

5.2.3 Problem Formulation

To efficiently use available wireless resources, we minimize communication load, D_m, by jointly optimizing communication and mode-switching systems subject to joint task completion probability, P_m^o, which can be formulated as

$$\begin{aligned} \min_{b_m, d_m} \; & D_m \approx \frac{d_m \tau_m W_m}{Z_m} \left[C_m - \sqrt{\frac{V_m}{\tau_m W_m}} Q^{-1}(\varepsilon_m^{\mathrm{d}}) \right] \\ \text{s.t.} \; & P_m^o > \psi_m. \end{aligned} \tag{5.7}$$

where ψ_m is a task-dependent task completion requirement.

5.3 Intelligent Mode-switching Framework

5.3.1 Task-level Prediction: User Intention Recognition

User intention recognition problem can be modeled as time series classification where CNNs [141], LSTM [142], RNNs, and distance based classification methods [143] such as k-Nearest Neighbours (kNNs) and Dynamic Time Warping (DTW) are used in the existing literature [144]. In this study, we design CNN based time series classification model as shown in Fig. 5.2. In input layer, multivariate time series data, $\mathbf{k}_{t:t'}$, are received at time slot t' with observation length $t' - t$ time slots. In convolution layers, the element-wise convolution operation is applied to compute the feature representations of inputs (i.e., *feature maps*). For input $\mathbf{k}_{t:t'}$ and kernel $\hat{\rho}_t$,

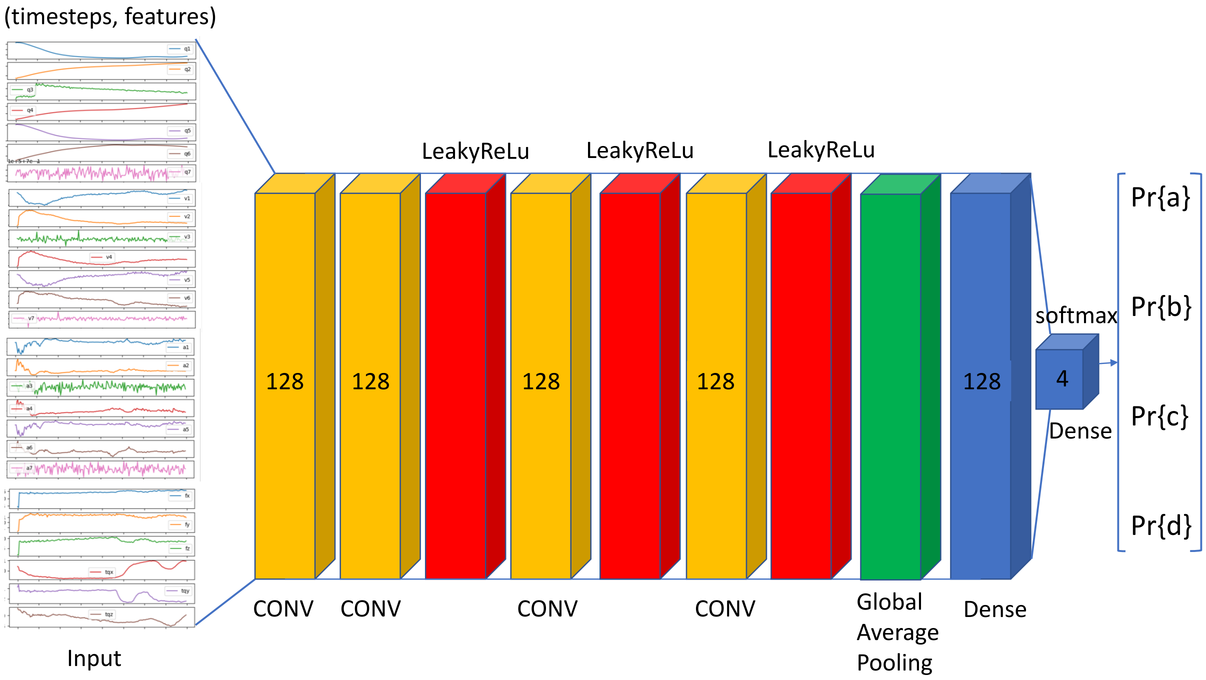


Figure 5.2: Task-level Prediction: User Intention Recognition Model

the resulting feature at location (i, j) can be computed as,

$$\mathbf{X}^{i,j}_{t:t'} = \mathbf{W}_{\hat{\rho}_{t:t'}} * \mathbf{k}^{i,j}_{t:t'} + \mathbf{b}_{\hat{\rho}_{t:t'}}, \tag{5.8}$$

$$\mathbf{Y}^{i,j}_{t:t'} = \Phi(\mathbf{X}^{i,j}_{t:t'}), \tag{5.9}$$

where $\mathbf{W}_{\rho_{t:t'}}$ and $\mathbf{b}_{\rho_{t:t'}}$ are the weights and bias of the filter $\rho_{t:t'}$, $\mathbf{k}^{i,j}_{t:t'}$ is the subsection of the input centered at (i, j), $\Phi(\cdot)$ is the non-linear activation function, and '$*$' is the convolution operator. Then, activation layers are employed after convolution layers using Leaky Rectified Linear Unit, *LeakyReLu*, activation function. After feature map computations with several convolution and activation layers, the pooling layer is employed to decrease the dimension of feature maps in which global average pooling is used. Then, two dense, i.e., fully connected, layers are employed to train the classification model with *softmax* activation function. For a given observation window, $\mathbf{k}_{t:t'}$, the classifier is trained to predict the class of the task. The categorical cross-entropy function is used as a loss function with one-hot encoded labels. Each convolution layer has 128 CNN cells and the batch size is 64. Early stopping criteria is adopted to avoid over-fitting, i.e., the best model is saved if there is no improvement for 10 consecutive epochs in the training process.

5.3.2 Trajectory-level Prediction

The teleoperator predicts the future trajectory from the observed trajectory to be able to finish the task autonomously, in the autonomous mode. Let's assume the teleoperator has observation of

Table 5.1: Hyper-Parameters of Trajectory-level Prediction Models

	LSTM	**CNN**
Number of layers	1	1
Number of cells in each layer	128 LSTM cells	128 Convolutional cells
Batch size	128	128
Optimizer	Adam	Adam
Loss function	MSE	MSE
Accuracy metric	RRMSE	RRMSE
Max training epochs	1000	1000
Activation function	tanh	ReLu

trajectories, $\mathbf{k}_{t:t'}$, at $t^{'}$-th time slot with observation length $t^{'}-t$. Given the observation window and the task label, the trajectory-level predictor predicts future trajectory, $\mathbf{k}_{t':t''}$, with prediction horizon $t^{''}-t^{'}$. We use two types of predictors, namely LSTM and CNN for trajectory-level prediction. Observation and prediction window lengths depend on mode-switching timings. For example, if the mode is switched from teleoperation to autonomous in the middle of the task, then the observation and the prediction windows are in equal length. Hyper-parameters of trajectory-level prediction algorithms are given in Table 5.1.

5.3.3 DRL Framework

In this section, we formulate the problem as discrete-time Markov Decision Process.

States

Let's denote the state of the M devices in the t-th slot as $\mathbf{S}(t) = (S_1(t), S_2(t), ..., S_M(t))$. The state of the m-th device consists of the predicted user intention, $L_m(t)$, the observation length, $T_m(t)$, and the current mode, $M_m(t)$, i.e.,

$$S_m(t) = \{L_m(t), T_m(t), M_m(t)\} \tag{5.10}$$

Actions

The action to be taken by the controller is to either switch the mode or keep the system in the same mode. We denote the action by $A_m(t)$, which is a discrete binary value: (0) staying in the current mode (1) switching to a different mode.

Reward

DRL agent errors can have significant consequences, impacting both the communication load and the successful completion of tasks. Firstly, when a DRL agent remains in teleoperation mode instead of switching to autonomous mode, it can result in unnecessary communication load. This inefficiency can hinder the overall system performance. Secondly, if the DRL agent mistakenly switches to autonomous mode when it should remain in teleoperation mode, it may lead to task failure. To address these challenges effectively, we have defined a reward function that considers these potential pitfalls. By providing a small reward in autonomous mode, we encourage the DRL agent to minimize communication load. Additionally, we incorporate an ultimate reward, contingent upon task completion outcomes, to encourage successful task execution. This reward structure ensures that the DRL agent is motivated to balance both communication efficiency and task accomplishment, thus improving overall system performance and reducing the likelihood of errors. Therefore, the reward depends on the current state and the final outcome of the task, which is defined as follows.

$$R_m(t) = \begin{cases} 1, & t \leq Z_m \text{ and } M_m(t) = auto \\ 0, & t \leq Z_m \text{ and } M_m(t) = tele \\ \frac{Z_m - d_m}{Z_m} \times 100, & t = Z_m \text{ and } P_m^o > \psi_m \end{cases} \tag{5.11}$$

5.3.4 DRL Training

We apply DQN [145] to find the optimal mode-switching policy. The details are given in Algorithm 3. The DQN algorithm has a main network, Q, and a target network, $\tilde{Q}$. The initial parameters of both neural networks are denoted by θ and $\tilde{\theta}$, respectively. We apply the ε-greedy policy to achieve a good trade-off between exploration and exploitation, where the agent chooses a random action with probability ε and chooses the action, A_t, with probability $1-\varepsilon$ according to,

$$A_t = \arg\max_{A_t} Q(S_t, A_t|\theta). \tag{5.12}$$

After taking the action, A_t, agent observes the reward, R_t, and the next state, S_{t+1}. The observed transition $< S_t, A_t, R_t, S_{t+1,}, \mathbb{1}_t >$ is stored in replay buffer, Γ, with capacity κ. We denote $\mathbb{1}_t$ as an indicator showing whether the task is terminated or not in the t-th slot. After collecting enough transitions in the replay buffer, the training starts. Specifically, the agent samples a batch of transitions, $< S_j, A_j, R_j, S_{j+1}, \mathbb{1}_j >$, $j \in \mathcal{N}$, and estimates the long-term reward from the target network,

$$Y_j = \begin{cases} R_j, & \mathbb{1}_j = 1 \\ R_j + \gamma \max_{A_{j+1}} \tilde{Q}(S_{j+1}, A_{j+1}), & \mathbb{1}_j = 0 \end{cases} \tag{5.13}$$

Then the Stochastic Gradient Descent (SGD) algorithm is applied to minimize the mean square Bellman error, i.e., the difference between the main network and Y_j,

$$\mathscr{L} = \frac{1}{N} \sum_{j=1}^{N} (Q(S_j, A_j) - Y_j). \tag{5.14}$$

Finally, the target network is updated in every C steps.

Algorithm 3 DQN training

```
Input: Initialize main network, $Q$ and target network, $\tilde{Q}$ with random weights, $\theta$ and $\tilde{\theta}$, respectively. Initialize replay buffer, $\Gamma$. Observe the initial state, $S_t = \{L_t, T_t, M_t\}$, with predicted user intend, $L_t$, observation length, $T_t$, and current mode, $M_t$.
for episode = 1,... do
    Choose an action, $A_t$, randomly with probability $\varepsilon$, or obtain the action $A_t = \arg\max_{A_t} Q(S_t, A_t|\theta)$ with probability $1 - \varepsilon$
    Apply action $A_t$, observe reward, $R_t$, and next state, $S_{t+1}$
    Store transition $< S_t, A_t, R_t, S_{t+1}, \mathbb{1}_t >$ in replay buffer, $\Gamma$
    Sample $N$ batch of transitions randomly from replay buffer, $\Gamma$
    for each transition $< S_j, A_j, R_j, S_{j+1}, \mathbb{1}_j >$ in $N$ batches do
        if $done_j$ then
            $Y_j = R_j$
        else
            $Y_j = R_j + \gamma \max_{A_{j+1}} \tilde{Q}(S_{j+1}, A_{j+1})$
        end if
    end for
    Update $\theta$ by minimising loss $\mathscr{L} = \frac{1}{N} \sum_{j=1}^{N} (Q(S_j, A_j) - Y_j)$
    Copy $\theta$ into $\tilde{\theta}$ in every $\mathscr{C}$ steps
end for
```

5.4 Evaluation of the Proposed Framework

In this section, we evaluate task-level prediction, trajectory-level prediction, and the proposed intelligent mode-switching framework. In the simulations, we use the relationships derived in (5.6). In trajectory-level prediction, we utilize the LSTM model. The operator coefficient, ρ_m, is assumed to be 0.85 and each packet has $b_m = 256$ bits of information. Communication system reliability is set to $1 - 10^{-5}$.

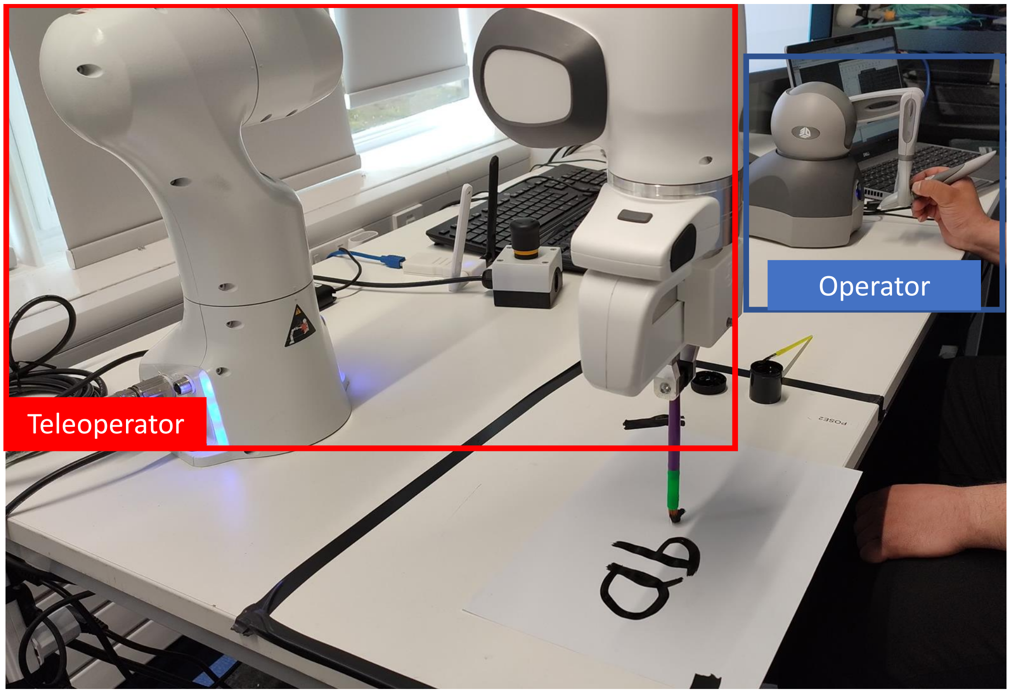


Figure 5.3: Dataset collection from teleoperation testbed

5.4.1 Dataset Collection from Teleoperation Prototype

To train task-level predictor, trajectory-level predictor, and DRL agent, we collect real-world trajectory samples from our teleoperation testbed. A human operator controls a robotic arm to write four different letters, i.e., a, b, c, d, as four types of tasks as seen in Fig. 5.3. Each observation is a time series trajectory given by $\mathbf{k}_{t:t'} = \{\mathbf{k}_t, \mathbf{k}_{t+1}, ..., \mathbf{k}_{t'}\}$ where each observation, $\mathbf{k}_t = [q_t, v_t, a_t, f_t, tq_t]$, consists of an angular position, q_t, and angular velocity, v_t, angular acceleration, a_t, measured external force and torque on the end-effector, f_t, and tq_t, respectively, in the t-th time slot. Each trajectory is collected with timestamps and letter labels at a frequency of 33 Hz. In total, 600 letter trajectory samples are collected which corresponds to 1.98×10^5 data points.

5.4.2 Task-level Prediction: User Intention Recognition

To evaluate the task-level prediction, we conduct experiments with the collected trajectory data. The classification accuracy of task-level prediction is studied with different observation lengths. The accuracy of the task-level prediction increases with observation length, as shown in Fig. 5.4. This is reasonable since more information about the task leads to a more accurate task classification. After 60% of the task, the classification model achieves more than 90% classification accuracy with smaller deviation as shown in Fig. 5.4.

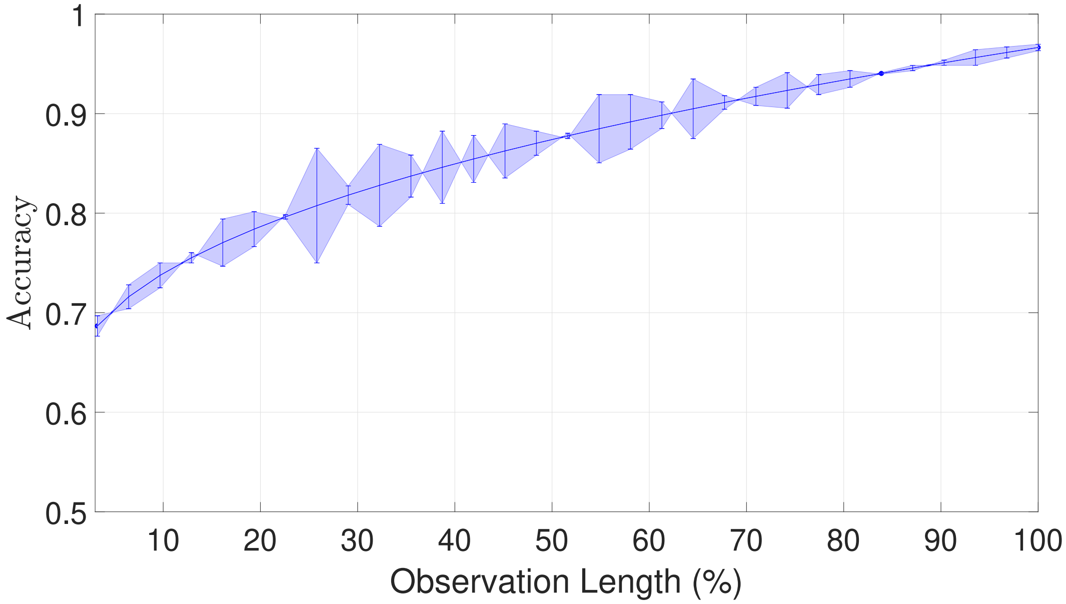


Figure 5.4: User intention recognition (task-level prediction) accuracy vs observation length(% of task).

Table 5.2: Performance of Trajectory-level Predictors

Observation Length (%)	Errors(%)	LSTM	CNN
50%	**Training RRMSE**	10.66%	11.04%
	Testing RRMSE	10.93%	11.08%
60%	**Training RRMSE**	9.37%	9.49%
	Testing RRMSE	9.84%	9.91%
70%	**Training RRMSE**	6.61%	6.94%
	Testing RRMSE	7.48%	7.68%
80%	**Training RRMSE**	4.24%	5.01%
	Testing RRMSE	4.93%	5.30%
90%	**Training RRMSE**	1.99%	1.93%
	Testing RRMSE	2.42%	2.50%

5.4.3 Trajectory-level Prediction

Trajectory-level prediction algorithms are evaluated with different observation lengths between 50% and 90%. The observation length of the trajectory-level prediction algorithm depends on mode-switching timings. For example, if the observation length is 50%, it means that the system is switched to the autonomous mode in the middle of the task and the trajectory-level predictor needs to predict the remaining trajectory to finish the task. Results for both LSTM and CNN models are provided in Table 5.2. According to the results, the accuracy of the trajectory-level prediction increases with increased observation length. This is reasonable since the prediction horizon decreases as the observation length increases. LSTM outperforms CNN, although CNN produces similar accuracy results. This is expected since LSTM performs better on time series (i.e., sequential) data compared to CNN.

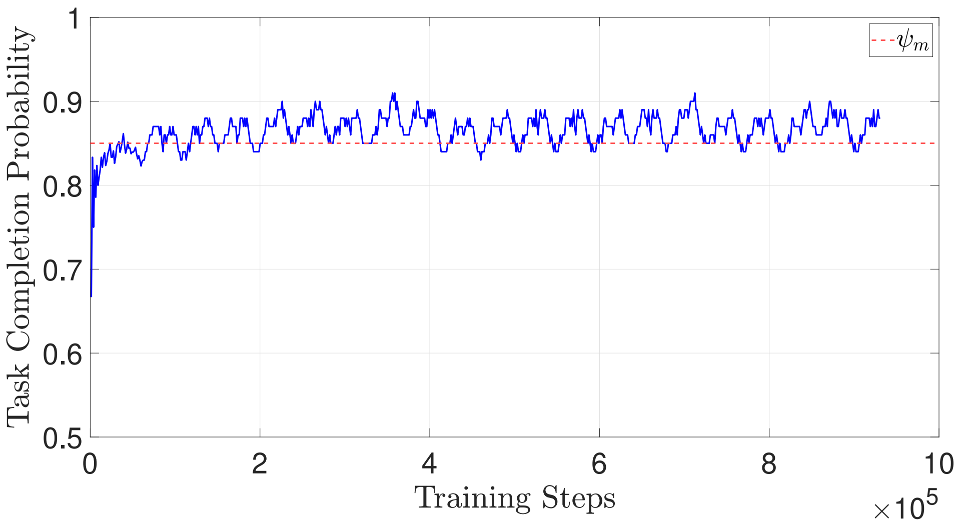


Figure 5.5: DRL training results for task completion probability, where $\psi_m = 0.85$.

5.4.4 DRL Agent

In DRL training, we use the proposed task-level and trajectory-level predictors. We provide training results in Fig. 5.5. From the figure, task completion probability increases at the beginning and converges to a value after training around 100,000 steps. We further evaluate the agent with a single task and multiple consecutive tasks, as shown in Figs. 5.6 and 5.7. System starts with the teleopertion mode, then the agent switches to the autonomous mode around 70% of the task. There are multiple mode switches that are triggered by the error detection mechanism. In other words, the agent switches back to the teleoperation mode if there is an error in user intention recognition. Around 75% of the task, agent switches back to the autonomous mode to finish the task autonomously. As seen from the Figs. 5.6 and 5.7, robot's operating mode switches rapidly and frequently over a period of time before switching to fully autonomous mode. These frequent switches happens in very small amount of time, i.e., $<$ 500 ms, hence, they are not noticeable by the operator. In other words, frequent and rapid switches over a period of time before switching to fully autonomous mode do not lead to unsmooth robot operation. In brief, results show that the agent learns the dynamics of the system and acts as expected.

5.4.5 Overall Results

We evaluate the proposed framework by comparing it with conventional teleoperation. In conventional teleoperation, the operator controls the robotic arm to finish the task, and task completion probability depends on the communication reliability and the operator coefficient as formulated in (5.5). In the proposed framework, intelligent mode-switching is applied and task completion probability depends on the intention recognition algorithm, the error detection system, and the trajectory-level prediction as formulated in (5.6).

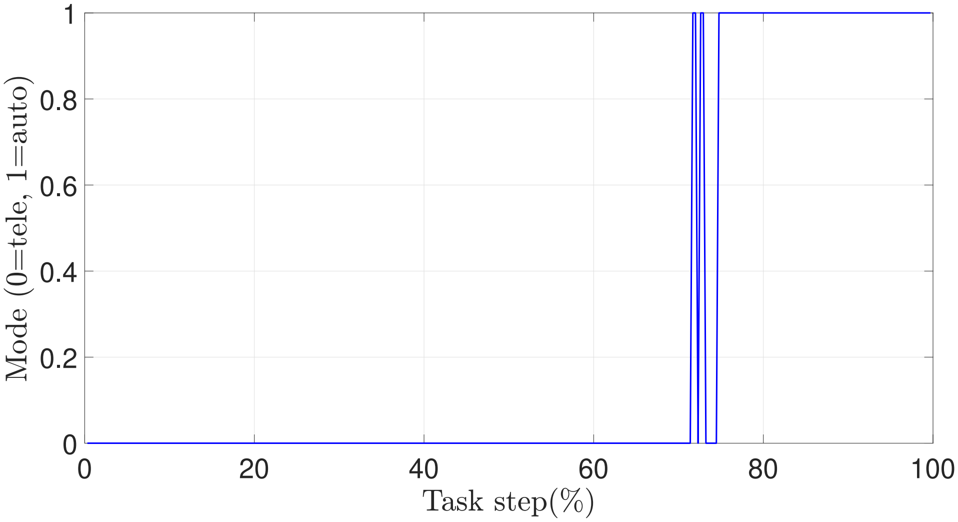


Figure 5.6: Single task example.

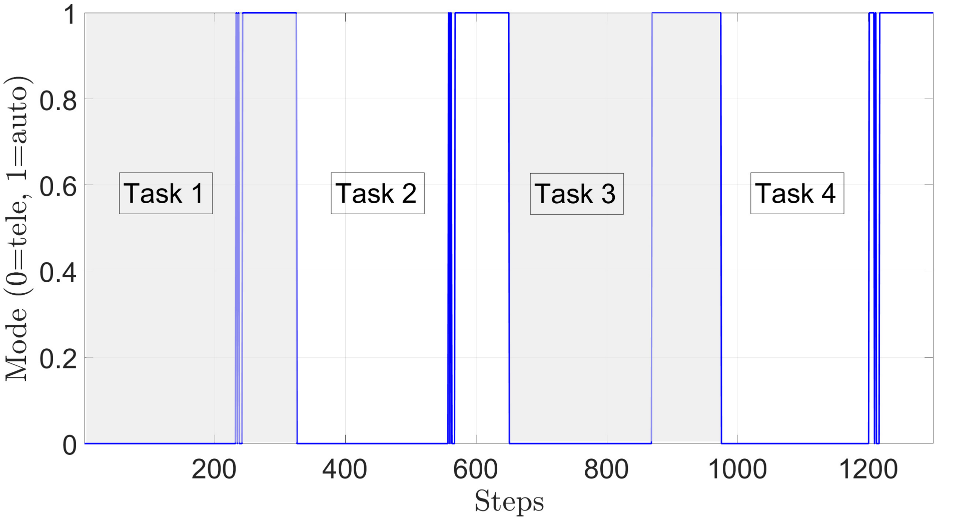


Figure 5.7: Multi task example.

According to our results, the proposed framework demonstrates comparable task completion probabilities while significantly reducing communication load, as illustrated in Fig. 5.8. The figure reveals an interesting trend: task completion probability initially increases and then decreases with the probability that the system is in teleoperation mode, denoted as $P^{\mathbf{t}}$. This behavior can be explained through two asymptotic scenarios. Firstly, when $P^{\mathbf{t}}$ is too small, the system operates solely in autonomous mode, relying on user intention recognition and trajectory prediction algorithms, which are not entirely error-free. Consequently, task completion probability may suffer due to limited operator input. Conversely, when $P^{\mathbf{t}}$ is excessively large, the system remains constantly in teleoperation mode, relying heavily on communication performance and operator experience, both of which are also prone to errors. In such cases, task completion probability may decline due to overreliance on human intervention. To achieve an optimal task

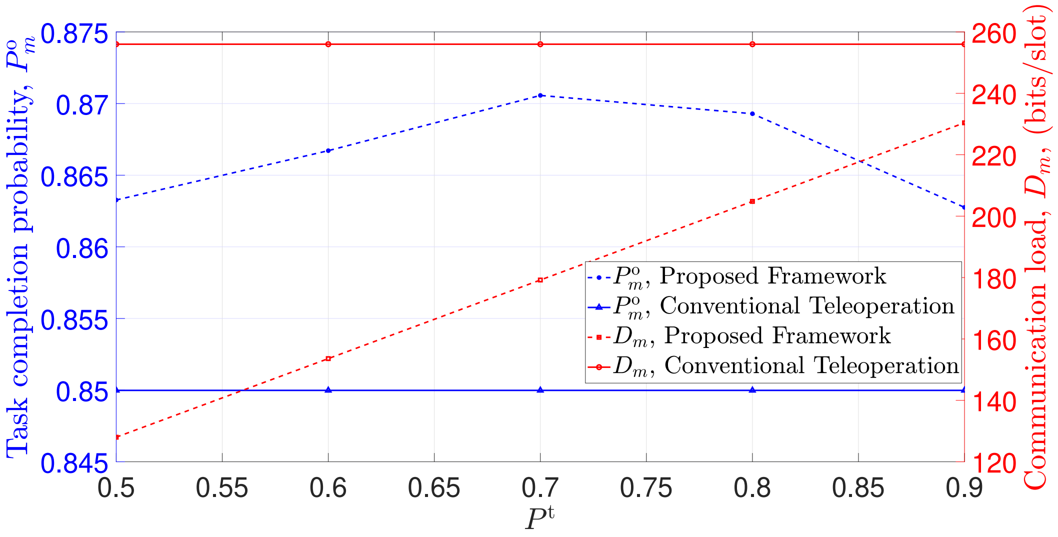


Figure 5.8: Task completion probability and communication load vs probability that system stays in teleoperation mode, where $\rho_m = 0.85$, $b_m = 256$ bits/slot, and $\varepsilon_m^{\mathrm{d}} = \varepsilon_m^{\mathrm{q}} = 10^{-5}$.

completion probability, the system must strike a balance between autonomy and teleoperation, effectively combining the advantages of both approaches. By finding this equilibrium, the proposed framework achieves promising results, maintaining task completion probabilities while significantly reducing communication load.

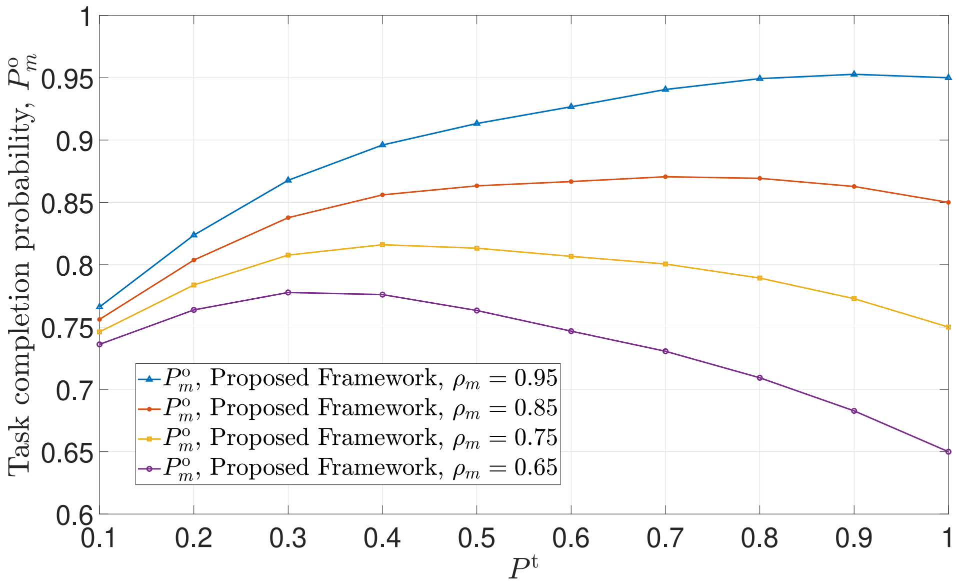


Figure 5.9: Task completion probability vs probability that system stays in teleoperation mode, for different operator experience coefficient ρ_m, where $b_m = 256$ bits/slot, and $\varepsilon_m^{\mathrm{d}} = \varepsilon_m^{\mathrm{q}} = 10^{-5}$.

We further illustrate task completion probability for different operator experience coefficient in Fig. 5.9. As seen from the figure, we observe similar trend of task completion probability for different user experience coefficient values.

In addition, we provide the comparison between conventional teleoperation and proposed approach in terms of task completion probability for different packet loss probabilities in Fig. 5.10. From the figure, both approach perform similar when the packet loss probability is low. On the other hand, proposed framework outperforms the conventional teleoperation when the packet loss probability is higher, showing that proposed framework is more resilient to poor network conditions.

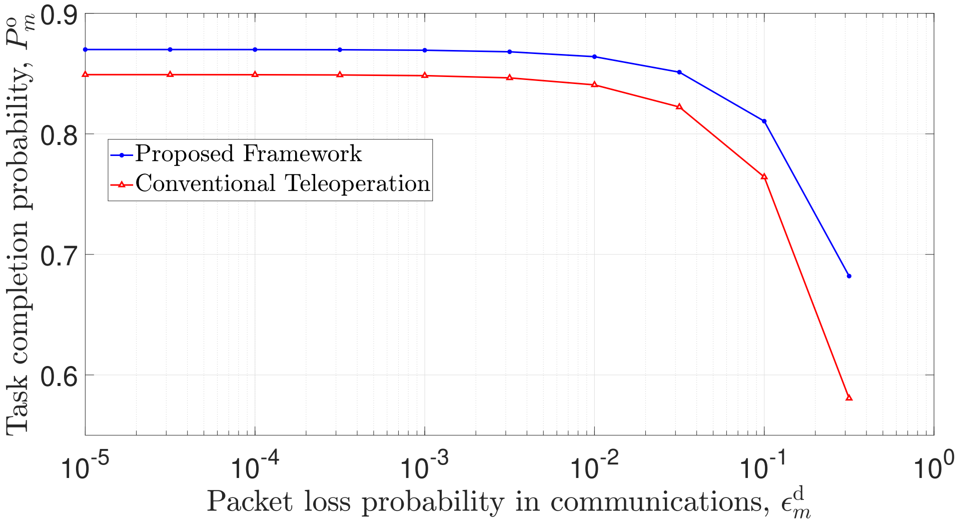


Figure 5.10: Task completion probability vs packet loss probability in communications, where $b_m = 256$ bits/slot, $P^{\mathrm{t}} = 0.7$, and $\rho_m = 0.85$.

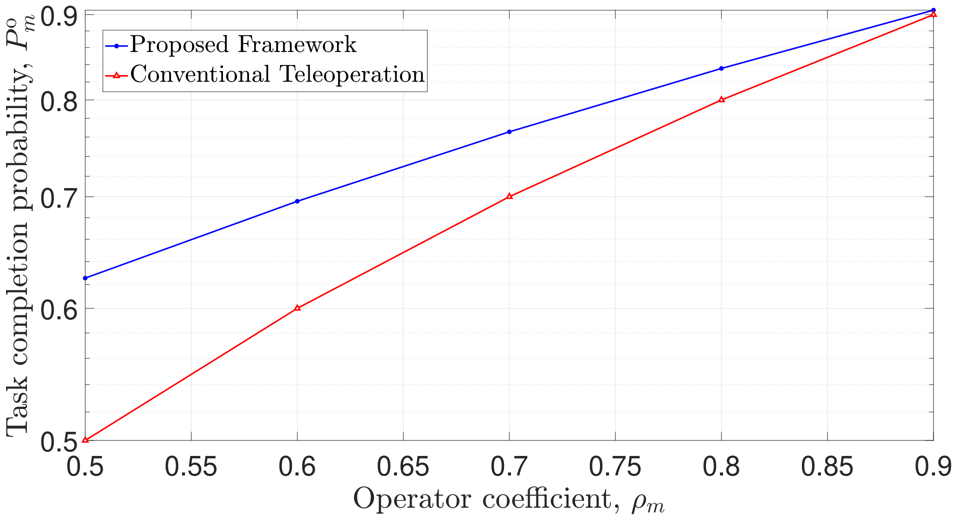


Figure 5.11: Task completion probability vs operator experience coefficient, where $b_m = 256$ bits/slot, $P^{\mathrm{t}} = 0.7$, and $\varepsilon_m^{\mathrm{d}} = \varepsilon_m^{\mathrm{q}} = 10^{-5}$.

Furthermore, we provide the comparison between conventional teleoperation and proposed approach in terms of task completion probability for different operator experience coefficient values in Fig. 5.11. From the figure, both approach perform similar when the operator experience coefficient is high. On the other hand, proposed framework outperforms the conventional teleoperation when the operator experience coefficient is lower, showing that proposed framework is more resilient to novice operator. In brief, the proposed framework improves task completion probability for novice operators and under poor communication conditions.

5.5 Conclusions

In this chapter, we propose an intelligent mode-switching framework to reduce communication load and improve task completion probability. The proposed framework presents an end-to-end joint design of mode-switching and communication systems to provide better overall performance. A real-world data set is used to train and test the system to validate the feasibility of the proposed framework in realistic scenarios. Our results corroborate that the proposed framework achieves better task completion probability for novice operators and under poor communication conditions with up to 50% communication load reduction.

Chapter 6

Conclusions and Future Works

In this thesis, we proposed the task-oriented joint design of communication and computing for Internet of Skills applications to meet three main stringent communication requirements, namely *ultra-low latency, ultra-high reliability*, and *wireless resource utilization efficiency*. We showed that it is challenging to meet these requirements, especially in long-distance communications. To address these challenges, we proposed novel design frameworks jointly considering communication and computing systems.

Specifically, we designed and implemented a 5G-based teleoperation prototype for Internet of Skills applications in Chapter 3. We presented two emerging Internet of Skills use cases in healthcare and education. We performed local and long-distance communication latency and reliability experiments considering both use cases to investigate the current capabilities and limitations. From our local experiments, we observed that communication latency is around 15 ms with a 99.9% packet reception rate (communication reliability). However, communication latency increases up to 2 seconds in long-distance scenarios (between the UK and China), while it is around 50-300 ms within the UK experiments. In addition, we observed that communication reliability and system performance are not directly related; instead, the number of consecutive packet drops is decisive on the overall system performance and user quality of experience. Furthermore, we proposed a two-way timeout approach considering the remote dental inspection use case where we discarded stale packets to mitigate waiting times. However, we observed that the proposed approach reduces the latency in the cost of reliability, which validates that latency and reliability are two conflicting requirements that are very challenging to meet.

As a future work, synchronization issues need to be considered between different feedback. Our implementation lacks a synchronization module that can guarantee synchronized video, audio, and haptic feedback. It is essential to implement this feature since unsynchronised feedback signals may lead to poor quality of experience and unstable performance.

We proposed a task-oriented prediction and communication co-design framework in Chapter 4 to meet conflicting latency and reliability requirements. In the proposed framework, we considered packet losses in communications and prediction errors in prediction algorithms to derive the upper bound for overall system reliability. The tradeoff between overall reliability and resource utilization efficiency was revealed, and we proposed an optimization problem to maximize resource utilization efficiency subject to E2E latency and overall reliability constraints. We took 5G NR as an example communication system. We derived closed-form expressions of queuing delay violation probability and decoding error probability for communication reliability. Similarly, we revealed the tradeoff between prediction error probability and prediction horizon for different JND thresholds. The proposed framework was evaluated with real-data samples and generated synthetic data samples. We showed that TimeGAN can be used to generate synthetic data samples if real data samples are not sufficient. From the results, the proposed framework achieved a 77.80% resource usage reduction compared to a task-agnostic benchmark. In addition, we showed that deploying a predictor at the receiver side achieves better overall reliability compared to a system that predictor at the transmitter.

As a future work, our framework can be extended by deploying predictors at both transmitter and receiver to compensate for reliability and latency imperfections for feedback links. Furthermore, the proposed framework assumes that the BS has knowledge about the task that is performed. Then, it allocates resources according to how critical the task is. However, it would be more interesting to extend our framework by adding a decision engine to categorize tasks according to their criticality level and allocate resources accordingly using our framework.

We proposed an intelligent mode-switching framework in Chapter 5 with a task-oriented co-design approach to address the wireless resource utilization challenge. We jointly considered communication, user intention recognition, and mode-switching parameters to reduce communication load subject to joint task completion probability. To achieve user intention recognition, we utilized a CNN-based classification model that takes the operator's real-time signal and tries to categorize the task. We revealed the tradeoff between task prediction accuracy and task observation length. We showed that higher prediction accuracy can be achieved when the task observation length increases. We achieved more than 90% task prediction accuracy with 60% observation length. Furthermore, we utilized LSTM for trajectory prediction that can achieve around 98% accuracy with 90% observation length. We trained a DRL agent with real-world data collected from our teleoperation prototype for mode-switching between teleoperation and autonomous modes. Our results showed that the proposed framework achieves up to 50% communication load reduction with similar task completion probability compared to conventional teleoperation.

As a future work, the proposed framework can be improved with more advanced classification and DRL algorithms. For DRL, expert knowledge can be used to improve performance. In addition, the trained DRL can only work with the trained tasks, which cannot be generalized. It can be interesting to extend our work by developing generalized DRL algorithms which can be used on any task.

Bibliography

[1] A. Aijaz and M. Sooriyabandara, “The tactile internet for industries: A review,” *Proceedings of the IEEE*, vol. 107, no. 2, pp. 414–435, 2018.

[2] F. Hu, Y. Deng, W. Saad, M. Bennis, and A. H. Aghvami, “Cellular-connected wireless virtual reality: Requirements, challenges, and solutions,” *IEEE Communications Magazine*, vol. 58, no. 5, pp. 105–111, 2020.

[3] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, “5G-enabled tactile internet,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, 2016.

[4] Z. Hou, C. She, Y. Li, L. Zhuo, and B. Vucetic, “Prediction and communication co-design for ultra-reliable and low-latency communications,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1196–1209, 2019.

[5] Z. Hou, C. She, Y. Li, D. Niyato, M. Dohler, and B. Vucetic, “Intelligent communications for tactile internet in 6g: Requirements, technologies, and challenges,” *IEEE Communications Magazine*, vol. 59, no. 12, pp. 82–88, 2021.

[6] M. Series, “Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond,” *Recommendation ITU*, vol. 2083, no. 0, 2015.

[7] M. Series, “Minimum requirements related to technical performance for imt-2020 radio interface (s),” *Report*, pp. 2410–0, 2017.

[8] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, “A brief history of the internet,” *ACM SIGCOMM computer communication review*, vol. 39, no. 5, pp. 22–31, 2009.

[9] G. Fettweis and S. Alamouti, “5g: Personal mobile internet beyond what cellular did to telephony,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 140–145, 2014.

[10] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[11] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information systems frontiers*, vol. 17, pp. 243–259, 2015.

[12] M. Dohler, T. Mahmoodi, M. A. Lema, M. Condoluci, F. Sardis, K. Antonakoglou, and H. Aghvami, "Internet of skills, where robotics meets ai, 5g and the tactile internet," in *2017 European Conference on Networks and Communications (EuCNC)*, pp. 1–5, IEEE, 2017.

[13] M. A. Lema, K. Antonakoglou, F. Sardis, N. Sornkarn, M. Condoluci, T. Mahmoodi, and M. Dohler, "5g case study of internet of skills: Slicing the human senses," in *2017 European Conference on Networks and Communications (EuCNC)*, pp. 1–6, IEEE, 2017.

[14] M. Dohler, "The internet of skills: How 5g-synchronized reality is transforming robotic surgery," *Robotic Surgery*, pp. 207–215, 2021.

[15] M. Dohler, "The future and challenges of communications—toward a world where 5g enables synchronized reality and an internet of skills," *Internet Technology Letters*, vol. 1, no. 2, p. e33, 2018.

[16] S. Latif, J. Qadir, S. Farooq, and M. A. Imran, "How 5G wireless (and concomitant technologies) will revolutionize healthcare?," *Future Internet*, vol. 9, no. 4, pp. 93–117, 2017.

[17] B. Kizilkaya, G. Zhao, Y. A. Sambo, L. Li, and M. A. Imran, "5G-enabled education 4.0: Enabling technologies, challenges, and solutions," *IEEE Access*, vol. 9, pp. 166962–166969, 2021.

[18] T. Zhang, "Toward automated vehicle teleoperation: Vision, opportunities, and challenges," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11347–11354, 2020.

[19] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Tactile internet and its applications in 5G era: A comprehensive review," *International Journal of Communication Systems*, vol. 32, no. 14, p. e3981, 2019.

[20] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.

[21] K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler, "Toward haptic communications over the 5G tactile internet," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3034–3059, 2018.

[22] Y. Miao, Y. Jiang, L. Peng, M. S. Hossain, and G. Muhammad, "Telesurgery robot based on 5g tactile internet," *Mobile Networks and Applications*, vol. 23, no. 6, pp. 1645–1654, 2018.

[23] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh, "Realizing the tactile internet: Haptic communications over next generation 5g cellular networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 82–89, 2016.

[24] 3GPP, TR 38.802, *Study on New Radio (NR) Access Technologies*. TSG RAN, Release 14, Mar. 2017.

[25] T. Jacobsen, R. Abreu, G. Berardinelli, K. Pedersen, P. Mogensen, I. Z. Kovács, and T. K. Madsen, "System level analysis of uplink grant-free transmission for urllc," in *2017 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, IEEE, 2017.

[26] A. A. El Kalam, A. Ferreira, and F. Kratz, "Bilateral teleoperation system using qos and secure communication networks for telemedicine applications," *IEEE Systems Journal*, vol. 10, no. 2, pp. 709–720, 2015.

[27] M. Patel and J. Wang, "Applications, challenges, and prospective in emerging body area networking technologies," *IEEE Wireless communications*, vol. 17, no. 1, pp. 80–88, 2010.

[28] M. Perez, S. Xu, S. Chauhan, A. Tanaka, K. Simpson, H. Abdul-Muhsin, and R. Smith, "Impact of delay on telesurgical performance: study on the robotic simulator dv-trainer," *International journal of computer assisted radiology and surgery*, vol. 11, no. 4, pp. 581–587, 2016.

[29] M. Eid, J. Cha, and A. El Saddik, "Admux: An adaptive multiplexer for haptic–audio–visual data communication," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 1, pp. 21–31, 2010.

[30] B. Cizmeci, X. Xu, R. Chaudhari, C. Bachhuber, N. Alt, and E. Steinbach, "A multiplexing scheme for multimodal teleoperation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 2, p. 21, 2017.

[31] T. Hachisu and H. Kajimoto, "Vibration feedback latency affects material perception during rod tapping interactions," *IEEE transactions on haptics*, vol. 10, no. 2, pp. 288–295, 2016.

[32] J. Lee, E. Tejedor, K. Ranta-aho, H. Wang, K.-T. Lee, E. Semaan, E. Mohyeldin, J. Song, C. Bergljung, and S. Jung, "Spectrum for 5g: Global status, challenges, and enabling technologies," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 12–18, 2018.

[33] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, “Digital twin for 5g and beyond,” *IEEE Communications Magazine*, 2020.

[34] J. Park, S. Samarakoon, H. Shiri, M. K. Abdel-Aziz, T. Nishio, A. Elgabli, and M. Bennis, “Extreme urllc: Vision, challenges, and key enablers,” *arXiv preprint arXiv:2001.09683*, 2020.

[35] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile edge computing: A survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[36] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, “Security for 5g and beyond,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3682–3722, 2019.

[37] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, “Wireless network design for control systems: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 978–1013, 2018.

[38] G. Zhao, M. A. Imran, Z. Pang, Z. Chen, and L. Li, “Toward real-time control in future wireless networks: communication-control co-design,” *IEEE Communications Magazine*, vol. 57, no. 2, pp. 138–144, 2018.

[39] S. Feyzabadi, S. Straube, M. Folgheraiter, E. A. Kirchner, S. K. Kim, and J. C. Albiez, “Human force discrimination during active arm motion for force feedback design,” *IEEE Transactions on Haptics*, vol. 6, no. 3, pp. 309–319, 2013.

[40] G. Durisi, T. Koch, and P. Popovski, “Toward massive, ultrareliable, and low-latency wireless communication with short packets,” *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1711–1726, 2016.

[41] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[42] R. A. Costa, M. Langberg, and J. Barros, “One-shot capacity of discrete channels,” in *2010 IEEE International Symposium on Information Theory*, pp. 211–215, IEEE, 2010.

[43] Y. Polyanskiy, H. V. Poor, and S. Verdú, “Channel coding rate in the finite blocklength regime,” *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[44] E. Steinbach, S. Hirche, M. Ernst, F. Brandi, R. Chaudhari, J. Kammerl, and I. Vittorias, “Haptic communications,” *Proceedings of the IEEE*, vol. 100, no. 4, pp. 937–956, 2012.

[45] S. J. Lederman and R. L. Klatzky, “Haptic perception: A tutorial,” *Attention, Perception, & Psychophysics*, vol. 71, no. 7, pp. 1439–1459, 2009.

[46] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 2000.

[47] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and benchmarking of machine learning accelerators," in *2019 IEEE high performance extreme computing conference (HPEC)*, pp. 1–9, IEEE, 2019.

[48] H. A. Helaly, M. Badawy, and A. Y. Haikal, "Deep learning approach for early detection of alzheimer's disease," *Cognitive computation*, pp. 1–17, 2021.

[49] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: Machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[50] H. Luan and C.-C. Tsai, "A review of using machine learning approaches for precision education," *Educational Technology & Society*, vol. 24, no. 1, pp. 250–266, 2021.

[51] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[52] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive mimo csi feedback: Design, simulation, and analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2827–2840, 2020.

[53] Y. Yang, D. B. Smith, and S. Seneviratne, "Deep learning channel prediction for transmit power control in wireless body area networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2019.

[54] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *IEEE INFOCOM 2017-IEEE conference on computer communications*, pp. 1–9, IEEE, 2017.

[55] A. Azari, P. Papapetrou, S. Denic, and G. Peters, "User traffic prediction for proactive resource management: Learning-powered approaches," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.

[56] W. Zhang, Y. Liu, T. Liu, and C. Yang, "Trajectory prediction with recurrent neural networks for predictive resource allocation," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pp. 634–639, IEEE, 2018.

[57] H. Gebrie, H. Farooq, and A. Imran, "What machine learning predictor performs best for mobility prediction in cellular networks?," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, IEEE, 2019.

[58] M. Lopez-Martin, B. Carro, J. Lloret, S. Egea, and A. Sanchez-Esguevillas, "Deep learning model for multimedia quality of experience prediction based on network flow packets," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 110–117, 2018.

[59] G. White, A. Palade, and S. Clarke, "Forecasting qos attributes using lstm networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[60] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[61] W. Lee, M. Kim, and D.-H. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1276–1279, 2018.

[62] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[63] C. L. Giles, G. M. Kuhn, and R. J. Williams, "Dynamic recurrent neural networks: Theory and applications," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 153–156, 1994.

[64] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[65] C.-W. Huang, C.-T. Chiang, and Q. Li, "A study of deep learning networks on mobile traffic forecasting," in *Proc. IEEE PIMRC*, pp. 1–6, 2017.

[66] S. Khan, N. Javaid, A. Chand, A. B. M. Khan, F. Rashid, and I. U. Afridi, "Electricity load forecasting for each day of week using deep cnn," in *Proc. WAINA*, pp. 1107–1119, Springer, 2019.

[67] T. Li, M. Hua, and X. Wu, "A hybrid cnn-lstm model for forecasting particulate matter (pm2.5)," *IEEE Access*, vol. 8, pp. 26933–26940, 2020.

[68] R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, and F. Li, "Multi-hour and multi-site air quality index forecasting in beijing using cnn, lstm, cnn-lstm, and spatiotemporal clustering," *Expert Systems with Applications*, vol. 169, p. 114513, 2021.

[69] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *Proc. IEEE ICACCI*, pp. 1643–1647, 2017.

[70] J. Yoon, D. Jarrett, and M. van der Schaar, “Time-series generative adversarial networks,” in *Proc. NeurIPS*, pp. 5509–5519, 2019.

[71] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[72] X. Liang, H. Zhang, L. Lin, and E. Xing, “Generative semantic manipulation with mask-contrasting gan,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 558–573, 2018.

[73] Y. Chen, Y.-K. Lai, and Y.-J. Liu, “Cartoongan: Generative adversarial networks for photo cartoonization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9465–9474, 2018.

[74] Y. Zhang, Z. Gan, and L. Carin, “Generating text via adversarial training,” in *NIPS workshop on Adversarial Training*, vol. 21, pp. 21–32, 2016.

[75] W. Fedus, I. Goodfellow, and A. M. Dai, “Maskgan: better text generation via filling in the_,” *arXiv preprint arXiv:1801.07736*, 2018.

[76] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, “Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification,” *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[77] Q. Wang, H. Yin, H. Wang, Q. V. H. Nguyen, Z. Huang, and L. Cui, “Enhancing collaborative filtering with generative augmentation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 548–556, 2019.

[78] Y.-L. Boureau, J. Ponce, and Y. LeCun, “A theoretical analysis of feature pooling in visual recognition,” in *Proc. ICML*, pp. 111–118, 2010.

[79] L. R. Medsker and L. Jain, “Recurrent neural networks,” *Design and Applications*, vol. 5, pp. 64–67, 2001.

[80] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[81] M. L. Puterman, “Markov decision processes,” *Handbooks in operations research and management science*, vol. 2, pp. 331–434, 1990.

[82] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[83] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, pp. 279–292, 1992.

[84] J. Jordon, L. Szpruch, F. Houssiau, M. Bottarelli, G. Cherubin, C. Maple, S. N. Cohen, and A. Weller, "Synthetic data–what, why and how?," *arXiv preprint arXiv:2205.03257*, 2022.

[85] X. Tong, G. Zhao, M. A. Imran, Z. Pang, and Z. Chen, "Minimizing wireless resource consumption for packetized predictive control in real-time cyber physical systems," in *Proc. IEEE ICC Workshops*, pp. 1–6, 2018.

[86] F. Boabang, R. Glitho, H. Elbiaze, F. Belqami, and O. Alfandi, "A framework for predicting haptic feedback in needle insertion in 5G remote robotic surgery," in *Proc. IEEE CCNC*, pp. 1–6, 2020.

[87] A. Hosseini, F. Richthammer, and M. Lienkamp, "Predictive haptic feedback for safe lateral control of teleoperated road vehicles in urban areas," in *Proc. IEEE VTC-Spring*, pp. 1–7, 2016.

[88] A. M. Girgis, J. Park, M. Bennis, and M. Debbah, "Predictive control and communication co-design via two-way gaussian process regression and AoI-aware scheduling," *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 7077–7093, 2021.

[89] A. M. Girgis, J. Park, C.-F. Liu, and M. Bennis, "Predictive control and communication co-design: A gaussian process regression approach," in *Proc. IEEE SPAWC*, pp. 1–5, 2020.

[90] Z. Hou, C. She, Y. Li, T. Q. Quek, and B. Vucetic, "Burstiness-aware bandwidth reservation for ultra-reliable and low-latency communications in tactile internet," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2401–2410, 2018.

[91] Z. Meng, C. She, G. Zhao, and D. De Martini, "Sampling, communication, and prediction co-design for synchronizing the real-world device and digital model in metaverse," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 288–300, 2022.

[92] S. Yoon, H. jeong Lim, J. H. Kim, H.-S. Lee, Y.-T. Kim, and S. Sull, "Deep 6-dof head motion prediction for latency in lightweight augmented reality glasses," in *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, IEEE, 2022.

[93] X. Hou and S. Dey, "Motion prediction and pre-rendering at the edge to enable ultra-low latency mobile 6dof experiences," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1674–1690, 2020.

[94] Y. Wang, S. Wu, J. Jiao, P. Yang, and Q. Zhang, "On the prediction policy for timely status updates in space-air-ground integrated transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2716–2726, 2022.

[95] Q. Xiong, X. Zhu, Y. Jiang, J. Cao, X. Xiong, and H. Wang, “Status prediction and data aggregation for aoi-oriented short-packet transmission in industrial iot,” *IEEE Transactions on Communications*, 2022.

[96] Q. Xiong, X. Zhu, Y. Jiang, J. Cao, and Y. Wang, “Status prediction for age of information oriented short-packet transmission in industrial iot,” in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 794–799, IEEE, 2022.

[97] Q. Wang, S. Xie, G. Zhao, L. Zhang, and Z. Chen, “Urllc packet management for packetized predictive control,” in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–5, IEEE, 2019.

[98] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[99] A. K. Tanwani and S. Calinon, “A generative model for intention recognition and manipulation assistance in teleoperation,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 43–50, IEEE, 2017.

[100] A. Akay and Y. S. Akgul, “An end-to-end stochastic action and visual estimation system towards autonomous teleoperation,” *IEEE Access*, vol. 10, pp. 16700–16719, 2022.

[101] A. D. Dragan and S. S. Srinivasa, “Assistive teleoperation for manipulation tasks,” in *Proceedings of the seventh annual ACM/IEEE international conference on human-robot interaction*, pp. 123–124, 2012.

[102] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, “Assistive teleoperation of robot arms via automatic time-optimal mode switching,” in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 35–42, IEEE, 2016.

[103] M. Gao, J. Oberländer, T. Schamm, and J. M. Zöllner, “Contextual task-aware shared autonomy for assistive mobile robot teleoperation,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3311–3318, IEEE, 2014.

[104] M. K. Zein, A. Sidaoui, D. Asmar, and I. H. Elhajj, “Enhanced teleoperation using autocomplete,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9178–9184, IEEE, 2020.

[105] M. K. Zein, M. Al Aawar, D. Asmar, and I. H. Elhajj, “Deep learning and mixed reality to autocomplete teleoperation,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4523–4529, IEEE, 2021.

[106] G. Gonzalez, M. Agarwal, M. V. Balakuntala, M. M. Rahman, U. Kaur, R. M. Voyles, V. Aggarwal, Y. Xue, and J. Wachs, "Deserts: Delay-tolerant semi-autonomous robot teleoperation for surgery," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12693–12700, IEEE, 2021.

[107] M. M. Rahman, N. Sanchez-Tamayo, G. Gonzalez, M. Agarwal, V. Aggarwal, R. M. Voyles, Y. Xue, and J. Wachs, "Transferring dexterous surgical skill knowledge between robots for semi-autonomous teleoperation," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1–6, IEEE, 2019.

[108] M. Agarwal, G. Gonzalez, M. V. Balakuntala, M. M. Rahman, V. Aggarwal, R. M. Voyles, Y. Xue, and J. Wachs, "Dexterous skill transfer between surgical procedures for teleoperated robotic surgery," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pp. 1236–1242, IEEE, 2021.

[109] M. H. Hussein, I. H. Elhajj, and D. Asmar, "Personalized autocomplete teleoperation: Real-time user adaptation using transfer learning with partial feedback," in *2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 175–180, IEEE, 2021.

[110] M. H. Hussein, B. Ibrahim, I. H. Elhajj, and D. Asmar, "Incremental learning for enhanced personalization of autocomplete teleoperation," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 515–521, IEEE, 2022.

[111] S. Jain and B. Argall, "Probabilistic human intent recognition for shared autonomy in assistive robotics," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–23, 2019.

[112] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.

[113] "Franka control interface documentation." `Available:https://frankaemika.github.io/docs/`, 2017. Accessed: 2020-11-25.

[114] P. M. Kebria, H. Abdi, M. M. Dalvand, A. Khosravi, and S. Nahavandi, "Control methods for internet-based teleoperation systems: A review," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 1, pp. 32–46, 2018.

[115] J. Marescaux, J. Leroy, F. Rubino, M. Smith, M. Vix, M. Simone, and D. Mutter, "Transcontinental robot-assisted remote telesurgery: feasibility and potential applications," *Annals of surgery*, vol. 235, no. 4, p. 487, 2002.

[116] ChinaDaily, "China performs first 5g-based remote surgery on human brain." `http://www.chinadaily.com.cn/a/201903/18/WS5c8f0528a3106c65c34ef2b6.html`, 2016.

[117] R. Durairajan, S. K. Mani, J. Sommers, and P. Barford, "Time's forgotten: Using ntp to understand internet latency," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, pp. 1–7, 2015.

[118] J. García-Guzmán, F. H. Villa-López, J. A. Vélez-Enríquez, L. A. García-Mathey, and A. Ramírez-Ramírez, "Remote laboratories for teaching and training in engineering," in *Design, control and applications of mechatronic systems in engineering*, ch. 3, pp. 47–50, Intech Open, 2017.

[119] M. Rampazzo, A. Cervato, and A. Beghi, "Remote refrigeration system experiments for control engineering education," *Computer Applications in Engineering Education*, vol. 25, no. 3, pp. 430–440, 2017.

[120] C. M. Ionescu, E. Fabregas, S. M. Cristescu, S. Dormido, and R. De Keyser, "A remote laboratory as an innovative educational tool for practicing control engineering concepts," *IEEE Transactions on Education*, vol. 56, no. 4, pp. 436–442, 2013.

[121] M. T. Restivo, J. Mendes, A. M. Lopes, C. M. Silva, and F. Chouzal, "A remote laboratory in engineering measurement," *IEEE transactions on industrial electronics*, vol. 56, no. 12, pp. 4836–4843, 2009.

[122] E. Fabregas, G. Farias, S. Dormido-Canto, S. Dormido, and F. Esquembre, "Developing a remote laboratory for engineering education," *Computers & Education*, vol. 57, no. 2, pp. 1686–1697, 2011.

[123] A. Chevalier, C. Copot, C. Ionescu, and R. De Keyser, "A three-year feedback study of a remote laboratory used in control engineering studies," *IEEE Transactions on Education*, vol. 60, no. 2, pp. 127–133, 2016.

[124] F. Luthon and B. Larroque, "Laborem—a remote laboratory for game-like training in electronics," *IEEE Transactions on learning technologies*, vol. 8, no. 3, pp. 311–321, 2014.

[125] N. Wang, X. Chen, G. Song, Q. Lan, and H. R. Parsaei, "Design of a new mobile-optimized remote laboratory application architecture for m-learning," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2382–2391, 2016.

[126] P. Trentsios, M. Wolf, and S. Frerich, "Remote lab meets virtual reality–enabling immersive access to high tech laboratories from afar," *Procedia Manufacturing*, vol. 43, pp. 25–31, 2020.

[127] A. Yayla, H. Korkmaz, A. Buldu, and A. Sarikas, “Development of a remote laboratory for an electronic circuit design and analysis course with increased accessibility by using speech recognition technology,” *Computer Applications in Engineering Education*, vol. 29, no. 4, pp. 897–910, 2021.

[128] J. M. Andujar, A. Mejías, and M. A. Márquez, “Augmented reality for the improvement of remote laboratories: an augmented remote laboratory,” *IEEE transactions on education*, vol. 54, no. 3, pp. 492–500, 2010.

[129] Y. Zhang, Y. Chen, B. Yu, X. Diao, and Y. Cai, “Minimizing age of information based on predictions and short packet communications in uav relay systems,” in *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–5, IEEE, 2021.

[130] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, “Quasi-static multiple-antenna fading channels at finite blocklength,” *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, 2014.

[131] S. Schiessl, J. Gross, and H. Al-Zubaidy, “Delay analysis for wireless fading channels with finite blocklength channel coding,” in *Proc. ACM MSWiM*, pp. 13–22, 2015.

[132] C.-S. Chang and J. A. Thomas, “Effective bandwidth in high-speed digital networks,” *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1091–1100, 1995.

[133] C. She, C. Yang, and T. Q. Quek, “Cross-layer optimization for ultra-reliable and low-latency radio access networks,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 127–141, 2017.

[134] J. Yoon, D. Jarrett, and M. van der Schaar, “Supplementary materials: Time-series generative adversarial networks.” `https://www.vanderschaar-lab.com/papers/NIPS2019_TGAN_Supplementary.pdf`.

[135] M. Abadi, P. Barham, J. Chen, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *Proc. USENIX OSDI*, pp. 265–283, 2016.

[136] J. Yoon, D. Jarrett, and M. van der Schaar, “Time series generative adversarial networks-github repository.” `https://github.com/jsyoon0823/TimeGAN`.

[137] Y. AI, “ydata-synthetic.” `https://github.com/ydataai/ydata-synthetic`.

[138] N. Nikhil and B. T. Morris, “Convolutional neural network for trajectory prediction,” in *Proc. ECCV, Springer*, pp. 186–196, 2018.

[139] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[140] A. Shustanov and P. Yakimov, “Cnn design for real-time traffic sign recognition,” *Procedia engineering*, vol. 201, pp. 718–725, 2017.

[141] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, “Convolutional neural networks for time series classification,” *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[142] F. Karim, S. Majumdar, H. Darabi, and S. Chen, “Lstm fully convolutional networks for time series classification,” *IEEE access*, vol. 6, pp. 1662–1669, 2017.

[143] A. Abanda, U. Mori, and J. A. Lozano, “A review on distance based time series classification,” *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 378–412, 2019.

[144] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Deep learning for time series classification: a review,” *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[145] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.