Deep Learning Based Resource Allocation for Ultra-Reliable Communications in Wireless Control Systems

by

Amirhassan Babazadeh Darabi

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

August, 2024

Deep Learning Based Resource Allocation for Ultra-Reliable Communications in Wireless Control Systems

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Amirhassan Babazadeh Darabi

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Sinem Çöleri

_____

Prof. Özgür Gürbüz

_____

Assoc. Prof. Ertuğrul Başar

Date: _____

To my beloved mother and father for their unwavering support, endless patience, and boundless love. And to my brother, the most loving and smartest person I know, and to my Nazgol, who has my pure love till infinity and beyond.

ABSTRACT

Deep Learning Based Resource Allocation for Ultra-Reliable Communications in Wireless Control Systems
Amirhassan Babazadeh Darabi
Master of Science in Electrical and Electronics Engineering
August, 2024

Wireless Networked Control Systems (WNCSs) play an important role in fifth-generation (5G) and sixth-generation (6G) networks to support mission-critical applications, such as Internet of Things (IoT), Remote Driving, and Collaborative Robots (Cobots). WNCS design demands consideration of both control and communication systems requirements to guarantee the broadcasting of reliable control commands at low latency from the controller to the actuators. In the first part of the thesis, a joint optimization of control and communication systems in the Finite Blocklength (FBL) regime is devised with the objective of minimizing the total power consumption by optimizing the sampling period of the control system, blocklength, and packet error probability of the communication system. Then, the optimization framework is simplified using optimality conditions to only one decision variable of blocklength. Then, the new optimization problem is fed to an online Deep Reinforcement Learning (DRL) algorithm to be trained, and the changing wireless environment is learned, executing optimal results. Second, a diffusion model, specifically the Denoising Diffusion Probabilistic Model (DDPM), is proposed to allocate resources for WNCSs. The optimization framework is utilized to collect a dataset of Channel State Information (CSI) and its corresponding optimal blocklength values. Then, the dataset is used to train the DDPM-based model to learn the complex distribution of the solution and the environmental parameters and generate optimal blocklength values based on the CSI as conditional information. The proposed schemes perform close to the optimization theory-based solution and outperform the previously proposed benchmarks, demonstrating superior performance in total power consumption and avoiding critical constraint violations.

ÖZETÇE

Kablosuz Kontrol Sistemlerinde Ultra Güvenilir İletişim için Deep Learning Tabanlı Kaynak Tahsisi

Amirhassan Babazadeh Darabi
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Ağustos 2024

Kablosuz Ağa Bağlı Kontrol Sistemleri (WNCS'ler), Nesnelerin İnterneti (IoT), Uzaktan Sürüş ve İşbirliğine Dayalı Robotlar (Cobot'lar) gibi kritik görev uygulamalarını desteklemek için beşinci nesil (5G) ve altıncı nesil (6G) ağlarda önemli bir rol oynar. WNCS tasarımı, güvenilir kontrol komutlarının kontrolörden aktüatörlere düşük gecikmeyle yayınlanmasını garanti etmek için hem kontrol hem de iletişim sistemi gereksinimlerinin dikkate alınmasını gerektirir. Tezin ilk bölümünde, kontrol sisteminin örnekleme periyodunu, blok uzunluğunu ve paket hatasını optimize ederek toplam güç tüketimini en aza indirmek amacıyla Sonlu Blok Uzunluğu (FBL) rejiminde kontrol ve iletişim sistemlerinin ortak optimizasyonu tasarlanmıştır. iletişim sisteminin olasılığı. Daha sonra optimizasyon çerçevesi, optimallik koşulları kullanılarak blok uzunluğunun yalnızca bir karar değişkenine göre basitleştirilir. Daha sonra yeni optimizasyon problemi, eğitilmek üzere çevrimiçi bir Derin Güçlendirme Öğrenme (DRL) algoritmasına beslenir ve değişen kablosuz ortam öğrenilerek en iyi sonuçların uygulanması sağlanır. İkinci olarak, WNCS'lere kaynak tahsis etmek için bir yayılma modeli, özellikle Gürültü Giderici Difüzyon Olasılık Modeli (DDPM) önerilmektedir. Optimizasyon çerçevesi, Kanal Durumu Bilgilerinin (CSI) ve buna karşılık gelen optimal blok uzunluğu değerlerinin bir veri kümesini toplamak için kullanılır. Daha sonra veri seti, çözümün karmaşık dağılımını ve çevresel parametreleri öğrenmek ve koşullu bilgi olarak CSI'ye dayalı en uygun blok uzunluğu değerlerini oluşturmak üzere DDPM tabanlı modeli eğitmek için kullanılır. Önerilen şemalar, optimizasyon teorisine dayalı çözüme yakın bir performans sergiliyor ve daha önce önerilen kıyaslamalardan daha iyi performans göstererek toplam güç tüketiminde üstün performans sergiliyor ve kritik kısıtlama ihlallerinden kaçınıyor.

ACKNOWLEDGMENTS

I wish to convey my heartfelt gratitude to my esteemed advisor, Dr. Sinem Çöleri, whose unwavering inspiration has been a guiding light throughout my journey. Her boundless motivation, infectious enthusiasm, and profound expertise have consistently impressed and inspired me. This dissertation would not have been possible without her continuous support.

I would also like to express my deepest appreciation to my dissertation committee, Dr. Özgür Gürbüz and Dr. Ertuğrul Başar for their invaluable time, insightful feedback, and constructive criticism throughout the evaluation of this thesis. Your expertise, guidance, and encouragement have been crucial in shaping this work, and I am sincerely grateful for your support and commitment.

To dear Dr. Alper T. Erdoğan, whose courses have inspired and motivated me in the field of communication. Thank you for your valuable comments and insights. It has been an honor to sit in your class.

Last but not least, I deeply appreciate my family, dear Nazgol, and friends for their unconditional love and support throughout every stage of my life. Thank you for always being there for me and motivating me throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

5G	fifth-generation
6G	sixth-generation
AI	Artificial Intelligence
AoI	Age of Information
AoL	Age of Loop
BDQ	Branching Deep Q-Network
CSCG	Circularly Symmetric Complex Gaussian
CSI	Channel State Information
D3QN	Dueling Double Deep Q-Network
DDPM	Denoising Diffusion Probabilistic Model
DDQN	Double Deep Q-Network
DL	Deep Learning
DNN	Deep Neural Network
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
EDF	Earliest Deadline First
ECDF	Empirical Cumulative Distribution Function
EMA	Exponential Moving Average
FBL	Finite Blocklength
FIFO	First-In First-Out
GAN	Generative Adversarial Network
GenAI	Generative AI
i.i.d.	Independent and Identically Distributed
IoT	Internet of Things
IP	Integer Programming

MAC	Media Access Control
MAD	Maximum Allowed Packet Delay
MATI	Maximum Allowable Transfer Interval
MDP	Markov Decision Process
MIMO	Multi-Input Multi-Output
NLP	Natural Language Processing
PER	Prioritized Experience Replay
PSD	Power Spectral Density
Q-Q	Quantile-Quantile
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
SNR	Signal-to-Noise Ratio
TDMA	Time Division Multiple Access
URLLC	Ultra-Reliable Low-Latency Communication
VoI	Value of Information
WNCS	Wireless Networked Control Systems

Chapter 1

INTRODUCTION

Wireless Networked Control Systems (WNCSs) rely on wireless communication networks to operate as closed-loop control systems. They are crucial for cutting-edge 6G network applications, such as remote driving, cooperative robots (cobots), and remote robot control. The primary challenge lies in the joint optimization of communication and control within these systems, given the stringent demands for ultra-reliability in control systems and low latency in communication systems. Moreover, the rapidly changing wireless environment further complicates the issue. Consequently, developing an optimization framework that integrates both systems is vital for critical control applications.

Earlier research typically focused on separately designing and optimizing control and communication systems, with the communication system tailored to meet the predetermined requirements of the control system. Conversely, the joint optimization approach introduces unique abstractions of control systems, such as Age-of-Information (AoI), Age-of-Loop (AoL), and Value-of-Information (VoI), to mathematically model the joint problem, addressing the needs of both the communication and control systems simultaneously. This optimization problem is often solved using heuristic or iterative algorithms by deriving the problem's optimality conditions. However, model-based approaches are not fast or practical for real-time scenarios and struggle to meet the low-latency requirements of WNCSs under URLLC conditions. Additionally, sensors and other devices transmit small, critical command packets, necessitating the use of the finite blocklength information-theoretic approach within the optimization framework.

Deep Learning (DL) is a prevalent technique for tackling complex problems by

learning solutions through training on extensive datasets. It has been applied across various domains, including image processing and natural language processing (NLP), to determine problem parameters and perform optimal actions to solve complex tasks. Recently, DL has found applications in communication engineering, such as channel estimation and resource allocation. A Deep Reinforcement Learning (DRL) agent aims to act optimally by interacting with its environment and receiving rewards. While DRL-based methods have been explored in the literature, it is the first time that optimization theory-based DRL approaches combine an optimization framework with DRL to develop a resource allocation strategy for ultra-reliable Wireless Networked Control Systems (WNCS).

Generative AI (GenAI) has been envisaged as one of the key points of next-generation wireless communications, enabling many applications. The advancement of diffusion models, a leading class of generative models, is recognized as a crucial factor in recent GenAI breakthroughs, including notable solutions like OpenAI's DALL.E 3. Alongside rapid advancements in AI algorithms, the 6G networks are expected to incorporate "native intelligence" as a fundamental element of system design. This highlights the need for state-of-the-art AI solutions to create "AI-native" wireless systems. Diffusion models can capture complex data distribution and generate high-quality samples from a Gaussian distribution because of their "denoise-and-sample" feature.

In this thesis, we propose a novel optimization theory-based DRL resource allocation algorithm for WNCSs in the FBL regime. First, the optimization framework is formulated based on the sampling period of the control system and the blocklength and packet error probability of the communication system. Then, by utilizing the optimality conditions, the problem is reduced to consider blocklength only. The simplified problem is passed through a DRL model to learn the wireless environment and execute optimal actions. In addition, a novel diffusion model-based strategy for resource allocation is proposed to mimic the distribution of the optimization theory-based solutions and generate optimal blocklength with channel state information (CSI) as conditional information.

1.1 Organization

The rest of the thesis is organized as follows:

- **Chapter 2** presents a novel joint control and communication optimization framework for WNCSs in the URRLC domain. This framework is based on the control systems' sampling period, blocklength, and packet error probability of the communication system in the finite blocklength (FBL) regime, aiming to minimize total power consumption. It also considers the schedulability and rate constraints of the communication system in the FBL regime, as well as the stability constraint of the control system. To reduce complexity, the optimization problem is simplified to focus on blocklength using optimality conditions, allowing for the decomposition of the problem into multiple blocks. This simplified problem is then used to develop a DRL framework, trained online to learn the environment. The DRL-based approach outperforms benchmark models and achieves near-optimal performance.[1]

- **Chapter 3** introduces a novel diffusion model algorithm for tackling the resource allocation problem. Diffusion models, known for their ability to capture complex data distributions, are widely used in generative AI. This chapter proposes a DDPM-based framework for allocating blocklength values in a WNCS system, aiming to minimize total power consumption through the joint optimization of control and communication systems. The proposed scheme utilizes an optimization theory-based solution to gather a dataset of CSI and corresponding blocklength values. This data is then used to train the resource allocation algorithm, which generates optimal blocklength values based on the CSI. The method surpasses previously proposed DRL-based algorithms and closely approximates the performance of optimization theory-based solutions. Additionally, the DDPM-based framework shows up to an eighteen-fold im-

[1]H. Q. Ali, A. Bababzadeh Darabi and S. Coleri, "Optimization Theory Based Deep Reinforcement Learning for Resource Allocation in Ultra-Reliable Wireless Networked Control Systems," in IEEE Transactions on Communications, doi: 10.1109/TCOMM.2024.3381712.

provement in reducing the frequency of critical constraint violations, underscoring the solution's effectiveness.[2]

- **Chapter 4** concludes the study and predicts the possible future works.

[2]Babazadeh Darabi, A., Coleri, S. (2024). Diffusion Model Based Resource Allocation Strategy in Ultra-Reliable Wireless Networked Control Systems. arXiv preprint arXiv:2407.15784.

Chapter 2

OPTIMIZATION THEORY BASED DEEP REINFORCEMENT LEARNING FOR RESOURCE ALLOCATION IN ULTRA-RELIABLE WIRELESS NETWORKED CONTROL SYSTEMS

The content of this chapter has been published in IEEE Transactions on Communications (TCOM).

2.1 Overview

The design of Wireless Networked Control System (WNCS) requires addressing critical interactions between control and communication systems with minimal complexity and communication overhead while providing ultra-high reliability. This chapter introduces a novel optimization theory based deep reinforcement learning (DRL) framework for the joint design of controller and communication systems. The objective of minimum power consumption is targeted while satisfying the schedulability and rate constraints of the communication system in the finite blocklength regime and stability constraint of the control system. Decision variables include the sampling period in the control system, and blocklength and packet error probability in the communication system. The proposed framework contains two stages: optimization theory and DRL. In the optimization theory stage, following the formulation of the joint optimization problem, optimality conditions are derived to find the mathematical relations between the optimal values of the decision variables. These relations allow the decomposition of the problem into multiple building blocks. In the DRL stage, the blocks that are simplified but not tractable are replaced by DRL. Via extensive simulations, the proposed optimization theory based DRL approach is demonstrated to outperform the optimization theory and pure DRL based

approaches, with close to optimal performance and much lower complexity.

2.2 Introduction

The recent advancements in the field of computing, sensing, control, and wireless technologies have drawn enormous interest in Wireless Networked Control Systems (WNCSs) that support emerging applications such as cyber-physical systems [Castro et al., 2022], Internet of Things [Bello and Zeadally, 2014], and Tactile Internet [Promwongsa et al., 2020]. WNCSs are spatially distributed control systems in which a physical plant is attached to sensor nodes that measure and transmit the plant state to the controller over a wireless channel. The controllers process the received information and forward their feedback to the actuators in order to control the physical plant [Hespanha et al., 2007], [Park et al., 2018b], [Al-Dabbagh and Chen, 2016]. WNCS plays a key role in Industry 4.0 [Kagermann et al., 2013] and several international organizations such as the Wireless Avionics Intra-Communications Alliance [Win,], Zigbee Alliance [Hingmire, 2015], International Society of Automation [of Automation, 2009], Highway Addressable Remote Transducer communication foundation [WHa,], and Industry IoT Consortium [IIO,]. The main challenge in the design of wireless control systems is to optimize the performance of control and communication systems together while considering the critical interaction between the parameters of the two systems that directly influence the efficiency of the control systems and the lifetime, throughput and reliability of the wireless communication systems.

Earlier research on the design and optimization of WNCS mostly focuses on model based methods, where mathematical models are used to model the performance of control and communication systems, and the solution methodologies employ either interactive design-based approach or joint design-based approach. The interactive design focuses on tuning the parameters of the wireless networks to meet the prespecified requirements of the control systems [Montalban et al., 2020], [Dobslaw et al., 2014]. The joint design of WNCS further considers the crucial interaction between the sampling period of the control system and the parameters of the wire-

less network such as the transmission power and rate at the physical layer, channel access mechanism at the MAC layer, and routing paths at the network layer in the formulation of the optimization problem. Due to the extensive complexity of the joint approach, earlier research was mostly focused on the usage of numerical techniques [Wu et al., 2010], [Pereira et al., 2007]. Later research has focused on providing the abstractions of control and communication systems to formulate the joint optimization problem [17]-[21]. In the joint optimization frameworks with the unique abstraction of the parameters of control and communication systems, the control system performance is formulated in terms of stochastic maximum allowed transfer interval (MATI) and the maximum allowed packet delay (MAD) constraint [Sadi et al., 2014], [Sadi and Coleri Ergen, 2017]. Stochastic MATI is defined as guaranteeing the time interval between the reception time of subsequent state vector reports above MATI value with a prespecified probability whereas MAD ensures that the packet transmission delay does not exceed a predefined maximum limit. Other abstractions of control systems include Age-of-Information (AoI) [Zhao et al., 2023], Age of Loop (AoL) [de Sant Ana et al., 2021] and Value of Information (VoI) [Farjam and Charalambous, 2021]. None of these studies incorporate ultra-reliability requirement into the joint design of control and communication systems. However, in many emerging WNCS applications, sensors and other devices generate small packets carrying critical information that is expected to be received with low latency and ultra-high reliability, which requires incorporating finite blocklength information theory into the optimization framework. Moreover, the runtime complexity of the proposed algorithms may not be tolerated within the strict latency limitation of the control systems due to the usage of optimization theory based iterative or heuristic solution methodologies.

Recent model based techniques developed for the resource allocation in WNCS incorporates finite blocklength information theory into their optimization framework [Ren et al., 2019], [Ren et al., 2020], [Ren et al., 2022], [Subchan et al., 2021]. [Ren et al., 2019] jointly optimizes the blocklength and power allocation for a factory automation scenario where a central controller transmits small packets to a robot and an actuator with the objective of minimizing the decoding error prob-

ability of the actuator subject to the reliability requirement of the robot as well as total energy and latency constraints. [Ren et al., 2020] studies the resource allocation for uplink massive MIMO systems by jointly optimizing the pilot and payload transmission power with the objective of maximizing the achievable data rate in the finite blocklength regime. [Ren et al., 2022] analyzes the performance of intelligent reflecting surfaces in terms of data rate and decoding error probability for URLLC in a factory automation scenario. [Subchan et al., 2021] proposes an optimal estimator at the controller based on Kalman Filter and Linear Quadratic Regulator, and an optimal power scheduler to minimize the energy per symbol. These works do not consider the sampling period of the control system in the formulation of the joint optimization. Furthermore, the high complexity of the proposed model based methods may prevent their usage in low latency WNCS applications.

Due to the high complexity of model-based methods and the high overhead of collecting environmental parameters, machine learning-based approaches have recently been proposed for the joint design of control and communication systems [Lima et al., 2022], [Zhao et al., 2023], [Baumann et al., 2018]. [Lima et al., 2022] proposes a RL-based algorithm for Markovian control and resource allocation policies with the objective to minimize the deviations of the plant states from the equilibrium point. [Zhao et al., 2023] proposes a DRL-based algorithm for controller and scheduler optimization. A scheduler at the controller schedules only necessary transmissions from the sensor in order to reduce the uplink transmission rate while guaranteeing a required level of control quality. [Baumann et al., 2018] proposes two approaches for learning resource-aware control policy by using a DRL-based algorithm; the first approach is based on an end-to-end learning of the structure of the communication and controller while the underlying policy outputs the communication decision and the control input. The second approach separately learns the communication strategy, and the stabilizing controller. These studies are completely data based without using any model of communication and control system, therefore, may require a large amount of training data in a dynamic environment.

In this chapter, we propose an optimization theory based deep reinforcement learning algorithm for the resource allocation in wireless networked control systems,

for the first time in literature. The resource allocation aims to determine the optimal values of all the parameters of the control and communication systems, including sampling period, blocklength, and packet error probability, with the goal of minimizing total power consumption. The constraints include the stochastic MATI and MAD constraints of the control systems, and the maximum transmit power and schedulability constraints of the communication system. The proposed framework utilizes the optimality conditions of mathematical system models in DRL to decrease the required amount of training data in DRL and enhance robustness against the non-idealities of the mathematical models. The novel contributions of this chapter are presented below:

- We propose an optimization theory based DRL framework for the joint optimization of controller and communication systems by considering the ultra-reliable transmission in the finite blocklength regime, for the first time in literature. The framework includes the formulation of the optimization problem based on the abstractions of the communication and control systems while considering the ultra-reliable transmission in the finite blocklength regime, the derivation of the optimality conditions based on this formulation, and the replacement of the specific parts of the optimization problem that are not tractable by DRL.

- We propose an optimization theory based DRL for the optimization of the sampling period, blocklength and packet error probability in the co-design of control and communication systems, for the first time in the literature. We perform this task in two stages, optimization theory stage and DRL stage. In the optimization theory stage, we first derive the optimality conditions among the sampling period, blocklength and packet error probability by using the model based approach. This allows us to reduce the decision variables to the blocklength only. In the DRL stage, we use DNN to formulate the remaining simplified problem.

- We analyze the performance of the proposed optimization theory based DRL

approach in comparison to the optimization theory model based approach and pure DRL approach via extensive simulations, for different network sizes, environments, stochastic MATI and MAD constraints.

This chapter is structured as follows: Section 2.3 presents control and communication system models and the underlying assumptions used in this chapter. Section 2.4 presents the formulation of the joint optimization of controller and communication systems. Section 2.5 introduces the proposed optimization theory based DRL approach. Section 2.6 presents the performance evaluation of the proposed approach in comparison to the optimization theory and pure DRL based benchmark algorithms. Section 2.7 concludes the chapter.

2.3 System Model and Assumptions

We assume a WNCS architecture comprising of multiple plants that are constantly monitored by a central controller over an unreliable wireless network. Multiple sensors attached to each plant are responsible for periodically measuring the plant state. The transmitter and receiver of the sensor nodes are assumed to be equipped with single antenna. We assume that the size of the packets carrying plant state information is extremely low, usually smaller than 20 bytes, to support the low-latency transmission [Yilmaz et al., 2015]. Moreover, the transmitted data may not be received by the controller due to the unreliable wireless communication channel. Based on the most recent received state update information, the controller computes a new control command and transmits it to the actuator. We assume that the actuator is collocated with the controller due to the criticality of the control command [Arampatzis et al., 2005] and therefore, the control command is always received successfully.

The sensors monitoring a plant send periodic state updates to the controller. The sampling period, the blocklength, the packet transmission delay and the packet error probability of node i are denoted by h_i, m_i, d_i and p_i, respectively, for $i \in \{1, 2, ..., N\}$. We assume that $d_i \leq h_i$ and the packets arriving later than the sampling period are considered as outdated and discarded in order to avoid out

of order arrival problem at the controller. The outdated packets are not retransmitted because outdated state information is useless or even harmful for time-critical control applications. The packet error is modeled as a Bernoulli random process to simplify the problem.

We assume Time Division Multiple Access (TDMA) as a channel access mechanism since TDMA ensures a deterministic access delay and is widely preferred for many industrial control applications [ANS,]. We assume that the channel time is divided into frames. Each frame is further divided into time slots with which the first slot is reserved for the beacon frame. The controller transmits the beacon periodically to broadcast the synchronization and scheduling updates among the nodes within the WNCS. In the scheduling update, nodes are assigned time slots for their corresponding data transmissions along with other parameters including the optimal transmission power, blocklength, and sampling period. We assume that nodes within the network do not transmit concurrently and the packet error rate is continually monitored by the network manager. We assume that the radio of each node can operate in three different modes; active mode in which it can sense the channel, transmit and receive a packet; sleep mode in which significant parts of the transceiver are switched off and the transient mode in which the node transitions from sleep to active state or vice versa. When the nodes are not scheduled to transmit or receive a packet, they switch to the sleep mode. We only consider the energy consumption in the transmission of packets in the active mode because the sleep and transient modes consume very small energy compared to active mode [Cui et al., 2005] and the beacon packets consume fixed energy.

Next, we discuss the system models for the control and communications systems based on the requirements and constraints related to the stability of the control system and the performance of the wireless communication system.

2.3.1 Control System Model

We formulate the stability and performance conditions of the control system in terms of stochastic $MATI$ and MAD as explained in detail in [Sadi and Coleri Ergen,

2017], and widely used in many control applications, such as wireless industrial automation [ANS,], air transportation systems [Park et al., 2014], and autonomous vehicular systems [Karagiannis et al., 2011].

Stochastic MATI Constraint

The stochastic MATI constraint is defined as the maximum allowed time interval between the reception of subsequent state vector reports above MATI value with a predefined probability. This constraint is formulated as

$$P[\mu_i(h_i, d_i(m_i), p_i) \leq \Omega] \geq \delta. \tag{2.1}$$

where μ_i denotes the time interval between two subsequent state vector updates of sensor node i, Ω denotes the stochastic MATI and δ indicates the minimum probability with which MATI should be achieved. μ_i is a function of the sampling period h_i, packet transmission delay d_i and packet error probability p_i. The delay d_i is a function of the blocklength m_i which denotes the number symbols used to transmit packet of length L_i bits. In order to satisfy $\mu_i \leq \Omega$, there must be at least one successful transmission within Ω duration. For given values of h_i and Ω, there are $\lfloor \frac{\Omega}{h_i} \rfloor$ opportunities of the state vector update reception. We model the packet error p_i as a Bernoulli random process and rewrite the stochastic MATI constraint given in Equation (2.1) as

$$1 - p_i^{\lfloor \frac{\Omega}{h_i} \rfloor} \geq \delta. \tag{2.2}$$

The values of Ω and δ are determined by the underlying applications.

MAD Constraint

The MAD constraint is defined as the maximum packet transmission delay not exceeding a predefined maximum limit. This constraint is included in order to

stabilize the control system and ensure its efficiency. MAD is formulated as

$$d_i(m_i) \leq \Delta, \tag{2.3}$$

where Δ denotes MAD. The underlying control system determines the value of MAD.

2.3.2 Communication System Model

We formulate the blocklength, power consumption, transmit power, and packet error probability constraints of the underlying communication system in the finite blocklength regime.

Blocklength Constraint

The plant state information measured by node i is transmitted in the form of small packets of length L_i by using blocklength m_i such that

$$m_i \leq M_{th}, \tag{2.4}$$

where M_{th} is the maximum allowed number of channel uses or maximum delay, within which the packet transmission has to finish. The coding rate is defined as the ratio of the number of information bits to the number of symbols used to transmit them. The Shannon capacity is defined as the highest coding rate of a communication system, provided that an encoder/decoder pair exists whose decoding error probability becomes negligible when the blocklength approaches infinity [She et al., 2018]. However, in URLLC, the blocklength for each frame is small and the decoding error probability cannot be ignored. The coding rate R_i (in bits/sec/Hz) of node i under finite blocklength regime can be approximated as [Polyanskiy et al., 2010]

$$R_i \approx \log_2(1+\gamma_i) - \sqrt{\frac{V_i}{m_i}}\frac{Q^{-1}(\epsilon_i)}{ln(2)}, \tag{2.5}$$

where $\gamma_i = \frac{W_{tx,i}|g_i|}{\sigma^2}$ is the signal-to-noise ratio (SNR) at the controller; $W_{tx,i}$ is the transmit power of node i; g_i is the channel gain from the node i to the controller; σ^2 is the noise power; $V_i = 1-(1+\gamma_i)^{-2}$ is the channel dispersion; ϵ_i is the decoding error probability and $Q^{-1}(\cdot)$ denotes the inverse of the Gaussian Q function. In short frame structure, the frame duration is smaller than the channel coherence time, called a quasi-static channel, which means that the symbols in a packet experience similar fading states. Decoding an information bit wrongly is, therefore, considered as the only factor making a packet erroneous. For this reason, the decoding error probability has been interchangeably used for packet error probability [Durisi et al., 2016], [Ren et al., 2019]. We can safely replace ϵ_i by p_i in our equations in the rest of this chapter.

Power Consumption

Average power consumption W_i of node i is a function of the sampling period h_i, packet error probability p_i, and delay $d_i(m_i)$ such that

$$W_i(h_i, d_i(m_i), p_i) = \frac{(W_{tx,i} + W_i^c)d_i(m_i)}{h_i}, \tag{2.6}$$

where

$$d_i(m_i) = \frac{m_i}{B}, \tag{2.7}$$

W_i^c is the circuit power consumption when the transmitter is in the active mode and B is the bandwidth of the wireless channel. We derive the packet error probability p_i of node i from Equation (2.5) as

$$p_i = Q\left[\ln(2)\sqrt{\frac{m_i}{V_i}}\left(log_2(1+\frac{W_{tx,i}|g_i|}{\sigma^2}) - \frac{L_i}{m_i}\right)\right]. \tag{2.8}$$

The transmit power $W_{tx,i}$ of node i can be derived from Equation (2.8) as

$$W_{tx,i} = \frac{\sigma^2}{|g_i|}\left[\exp\left(\frac{Q^{-1}(p_i)}{\sqrt{m_i}} + \frac{\ln(2)L_i}{m_i}\right) - 1\right]. \tag{2.9}$$

Now Equation (2.6) can be rewritten as

$$W_i(h_i, d_i(m_i), p_i) = \frac{m_i \sigma^2}{h_i B |g_i|} \left[\exp\left(\frac{Q^{-1}(p_i)}{\sqrt{m_i}} + \frac{\ln(2) L_i}{m_i} \right) - 1 \right] + \frac{W_i^c m_i}{h_i B}, \tag{2.10}$$

where the channel dispersion is approximated by $V_i \approx 1, \forall i \in \mathbb{N}$, which is widely used in ultra-reliable communication for medium-to-high values of SNR to make the problem tractable [Ren et al., 2019].

Maximum Transmit Power Constraint

Each node can use a maximum power level, denoted by $W_{tx,max}$, for its transmissions. We formulate the maximum transmit power constraint as

$$W_{tx,i} \leq W_{tx,max}. \tag{2.11}$$

Schedulability Constraint

The schedulability constraint handles the assignment of transmission times to multiple sensor nodes in a network considering wireless transmission constraints. We assume that no two nodes can transmit concurrently and generate transmission schedule by the Earliest Deadline First (EDF) algorithm which is a dynamic scheduling algorithm prioritizing transmissions closest to their deadlines [Dertouzos, 1974]. When the maximum allowed packet transmission delay and the packet generation period are not equal, i.e. $\exists i \in [1, N], \Delta \neq h_i$, the simulation of the EDF algorithm can generate an exact schedule based on d_i, h_i, Δ and task arrival times a_i for all $i \in [1, N]$ which is feasible if and only if no deadlines are missed [Zhang and Burns, 2009], [Eisenbrand and Rothvoß, 2010]. For an optimization framework, however, we need to explicitly formulate the necessary and sufficient conditions for schedulability. We define the schedulability constraint as proposed by [Sadi et al., 2014]

Table 2.1: Table of frequently used notation

Notation	Stands For
B	Bandwidth
N	Number of nodes
m_i	Blocklength of node i
M_{th}	Maximum blocklength
L_i	Packet length of node i
W_i	Power consumption of node i
W_i^c	Circuit power consumption of node i
$W_{tx,i}$	Transmit power of node i
$W_{tx,max}$	Maximum transmit power constraint
Ω	MATI constraint
δ	Minimum probability with which MATI should be achieved
Δ	MAD
p_i	Packet error probability of node i
R_i	Data Rate in bits/symbol/Hz of node i
h_i	Sampling period in seconds of node i
g_i	Channel gain of node i
σ^2	Noise Power
$\gamma_i^{(t)}$	SNR of node i at time frame t
$\mathbf{m}^{(t)}$	Blocklength vector at time frame t
$\mathbf{h}^{(t)}$	Sampling period vector at time frame t
$\mathbf{p}^{(t)}$	Packet error probability vector at time frame t
$\mathbf{W}^{(t)}$	Transmit power vector at time frame t
$\mathbf{g}^{(t)}$	Channel gain vector at time frame t
$\mathcal{P}^{(t)}$	Total power consumption of nodes at time frame t
$\alpha^{(t)}$	Learning rate at time frame t
η	Discount factor
ϵ	Value for ϵ-greedy algorithm

$$\sum_{i=1}^{N} \frac{d_i(m_i)}{h_i} \leq \beta, \tag{2.12}$$

where β is defined as the utilization bound which should satisfy $0 < \beta \leq 1$. Equation (2.12) is a necessary and sufficient condition for a feasible schedule for $\beta = \beta_{nec} = 1$ and $\beta = \beta_{suf} = \min(1, \min_{i\in[1,N]} \frac{\Delta}{h_i})$, respectively [Zhang and Burns, 2009]. Each node i in the network is allocated a ratio of the total schedule length $\frac{d_i}{h_i}$ for its transmission. We assume that no two nodes can transmit concurrently, so the sum of these terms equals to the ratio of total time assigned to all the nodes $i \in [1, N]$ in the network to the schedule length. β can not exceed one because the total time allocated to all the nodes in the network should be within the schedule length. A table of frequently used notations is listed in Table 2.1.

2.4 Joint Optimization of Control and Communication Systems

In this section, we formulate the problem of the joint optimization of control and communication systems for ultra-reliable communication in the finite blocklength regime with the objective of minimizing the power consumption of the communication system. The model considers the stochastic MATI and MAD constraints to guarantee the stability of the control system, and maximum transmit power, maximum blocklength, power consumption and schedulability constraints of the wireless communication system. The joint optimization of control and communication systems is formulated as follows:

$$\underset{\substack{h_i,m_i,p_i \\ i\in[1,\mathcal{N}]}}{\text{minimize}} \quad \sum_{i=1}^{\mathcal{N}} C_{i1}C_2\frac{m_i}{h_i}\Big[\exp\Big(\frac{Q^{-1}(p_i)}{\sqrt{m_i}}+\frac{\ln 2L_i}{m_i}\Big)-1\Big] + \frac{C_2W_i^c m_i}{h_i} \tag{2.13a}$$

subject to

$$\lfloor\frac{\Omega}{h_i}\rfloor \ln p_i - \ln(1-\delta) \leq 0,\ \forall i \in [1,\mathcal{N}] \tag{2.13b}$$

$$0 < d_i(m_i) \leq \min(\Delta, h_i),\ \forall i \in [1,\mathcal{N}] \tag{2.13c}$$

$$0 < h_i \leq \Omega, \forall i \in [1,\mathcal{N}] \tag{2.13d}$$

$$0 < p_i \leq 1, \forall i \in [1,\mathcal{N}] \tag{2.13e}$$

$$m_i \leq M_{th},\ \forall i \in [1, \mathcal{N}] \tag{2.13f}$$

$$C_{i1}\Big[\exp\Big(\frac{Q^{-1}(p_i)}{\sqrt{m_i}} + \frac{\ln 2L_i}{m_i}\Big) - 1\Big] \leq W_{tx,max} \tag{2.13g}$$

$$\sum_{i=1}^{\mathcal{N}} \frac{d_i(m_i)}{h_i} \leq \beta, \tag{2.13h}$$

where the variables $C_{i1} = \frac{\sigma^2}{|g_i|}$ and $C_2 = \frac{1}{B}$. The decision variables of the optimization problem are the sampling period h_i, packet error probability p_i and blocklength $m_i, i \in [1, N]$. Equation 2.13a gives the objective of the problem for the minimization of the total power consumption, as derived in Equation (2.10). Equation 2.13b represents the stochastic MATI constraint. Equation 2.13c combines the MAD constraint with the requirement that the packets arriving later than the sampling period are considered as outdated, i.e. $d_i \leq h_i$. Equation 2.13d states that the sampling period $h_i, i \in [1, N]$ must be less than MATI. Equations 2.13e, 2.13f, 2.13g, and 2.13h represent the threshold constraint of the lower and upper bounds for the packet error probability, blocklength limit, the maximum transmit power constraint and the schedulability constraint, respectively.

This optimization problem is a non-convex Mixed-Integer Programming problem, thus, any solution for global optimum is difficult [Boyd and Vandenberghe, 2004a].

2.5 Optimization Theory Based Deep Reinforcement Learning

The optimization theory based deep reinforcement learning is based on the hybrid usage of the optimization theory based solution methodology using domain knowledge and deep learning techniques that learn from data. The approach consists of two stages: 1) Optimization theory stage: Following the formulation of the optimization

problem based on the domain knowledge from the system models of the control and communication systems, the solution is decomposed into multiple building blocks based on the derivation of the optimality conditions. 2) DRL stage: The building blocks that are complicated or intractable are replaced by deep learning structure, which is then trained based on data. This hybrid usage of model and data based approaches allows us to reduce the amount of training data used in model agnostic DRL. Next, these optimization and DRL stages are provided in Sections 2.5.1 and 2.5.2, respectively.

2.5.1 Optimization Theory Stage

In the optimization theory stage, the general solution is decomposed into multiple building blocks. The optimization theory enables the derivation of the optimality conditions on the sampling period, packet error probability and the blocklength. This allows the reformulation of the problem in terms of only one variable, blocklength, as an Integer Programming (IP) problem. The remaining variables, i.e. sampling period and packet error probability, can then be obtained by using the optimality conditions. Next, we formulate the optimality conditions to find the relation between the optimal sampling period and the optimal packet error probability.

Lemma 1: Let us denote the optimal values of the sampling period, packet error probability and blocklength by h_i^*, p_i^* and m_i^*, respectively. The relation between the optimal values of the sampling period and the packet error probability is given by

$$\frac{\Omega}{h_i^*} = \frac{\ln(1-\delta)}{\ln p_i^*} = k_i, \tag{2.14}$$

where k_i is a positive integer.

Proof. We prove the Lemma by contradiction in a similar way as in [Sadi et al., 2014]. We start by assuming that $\frac{\Omega}{h_i^*}$ is not a positive integer such that $\lfloor \frac{\Omega}{h_i^*} \rfloor < \frac{\Omega}{h_i^*}$. If we increase h_i^* such that $\lfloor \frac{\Omega}{h_i^*} \rfloor = \frac{\Omega}{h_i^*}$ while the upper bound given in Equation (2.13d) is not violated, the constraint in Equation (2.13b) still holds since the value of $\lfloor \frac{\Omega}{h_i^*} \rfloor$ remains unchanged and the constraints (2.13c) and (2.13h) also hold with the

increase in the value of h_i^*. However, the objective cost function given in Equation (2.13a) decreases since it is a monotonically decreasing function of h_i. Similarly, $\frac{\Omega}{h_i^*} = \frac{\ln(1-\delta)}{\ln p_i^*}$ can be proved by contradiction. We assume that $\frac{\Omega}{h_i^*} < \frac{\ln(1-\delta)}{\ln p_i^*}$ and increase $\ln p_i^*$ such that $\frac{\Omega}{h_i^*} = \frac{\ln(1-\delta)}{\ln p_i^*}$, the constraint in Equation (2.13g) still holds since the power consumption of node i is non-increasing function of p_i. However, the power consumption function in Equation (2.13a) decreases since it is a monotonically decreasing function of p_i. □

Next, we express the optimal value of k_i in terms of m_i. This enables us to express the optimization problem (2.13) in terms of only one variable m_i.

Lemma 2: The optimal value of k_i in Lemma 1, expressed by k_i^*, is as a function of m_i and given by

$$k_i^*(m_i) = \max \left[1, \frac{\ln(1-\delta)}{\ln\left[Q\left[\sqrt{m_i}\ln\left(\frac{W_{tx,max}}{m_i C_{i1}}+1\right)-\frac{\ln(2)L}{\sqrt{m_i}}\right]\right]}\right]. \tag{2.15}$$

Proof. The power consumption in Equation 2.13a is a monotonically increasing function of k_i. Therefore, k_i^* is the minimum positive integer that satisfies the constraints 2.13b-2.13e and 2.13g. In Equation (2.15), the second term of the maximum function is the minimum value of k_i that satisfies the constraint 2.13g since constraint 2.13c is satisfied by the minimum value of k_i in case a feasible solution exists. The constraints 2.13b and 2.13d-2.13f follow as the constraints 2.13c and 2.13g are satisfied. □

By using (2.7) and (2.14), Equation (2.12) can be rewritten as

$$\sum_{i=1}^{N} \frac{C_2 m_i k_i^*(m_i)}{\Omega} \leq \beta. \tag{2.16}$$

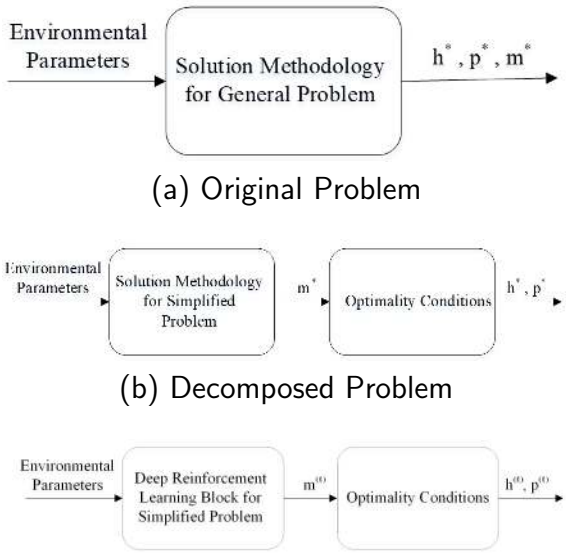


Figure 2.1: Problem decomposition by optimality conditions

The joint optimization problem 2.13 can then be reformulated as

$$\begin{aligned}\underset{m_i, i\in[1,\mathcal{N}]}{\text{minimize}} \quad & \sum_{i=1}^{\mathcal{N}} C_{i1} C_2 \frac{m_i k_i^*(m_i)}{\Omega} \\ & \Big[\exp\Big(\frac{Q^{-1}\left((1-\delta)^{\frac{1}{k_i^*(m_i)}}\right)}{\sqrt{m_i}} + \frac{\ln 2 \cdot L_i}{m_i}\Big) \\ & - 1\Big] + \frac{C_2 W_i^c m_i k_i^*(m_i)}{\Omega}\end{aligned} \tag{2.17a}$$

subject to

$$m_i \leq M_{th},\ \forall i \in [1,\mathcal{N}] \tag{2.17b}$$

$$\sum_{i=1}^{\mathcal{N}} \frac{C_2 m_i k_i^*(m_i)}{\Omega} \leq \beta. \tag{2.17c}$$

Figure 2.1 shows the decomposition of the optimization problem (2.17) into mul-

tiple building blocks based on the derivation of the optimality conditions. Let m^*, h^* and p^* denote the vectors containing the optimal values of the blocklength, sampling period and packet error probability given by m_i^*, h_i^* and p_i^* in the ith element, respectively. Figure 2.1a shows the solution methodology for the general problem, where environmental parameters are the input and the optimal values of blocklength, sampling period and packet error probability are the output. Figure 2.1b shows the decomposed problem, where in the first building block, the solution methodology is used for a simpler problem with environmental variables at the input and only optimal values of the block length at the output, and in the second building block, the optimal values of the blocklength are used to obtain the optimal values of the sampling period and packet error probability by using optimality conditions.

2.5.2 DRL Stage

In the DRL stage, the building blocks that are complicated or intractable are replaced by the deep learning block. Let $\mathbf{m}^{(t)}$, $\mathbf{h}^{(t)}$ and $\mathbf{p}^{(t)}$ denote the vectors containing the values of the blocklength, sampling period and packet error probability for node i, given by $m_i^{(t)}$, $h_i^{(t)}$ and $p_i^{(t)}$, respectively, in the ith element at time step t. In Figure 2.1c, the deep learning based building block is used to obtain the optimal values of the blocklength in the optimization problem (2.17). The main motivation for using data driven deep learning based solution is handling the high complexity of the problem and providing robustness based on the real data.

Next, we provide an overview of deep reinforcement learning, the proposed DRL-based solution and the proposed DRL-based framework.

Overview of Deep Reinforcement Learning

A Reinforcement Learning (RL) agent aims to act in an optimal manner by having several interactions with its environment and getting rewards [Sutton and Barto, 2018]. Markov Decision Process (MDP) is a mathematical model for representing decision-making problems evolving over discrete time steps, encompassing states, actions, transition probabilities, and rewards. RL employs the MDP framework for

uncertain system dynamics, where agents dynamically engage with the environment to learn and refine optimal sequential decision-making strategies over time. The tuple $s^{(t)} \in S$ is a state consisting of variables that have an impact on the changes in the environment at discrete time step t. The agent takes action $a^{(t)} \in A$ at time step t, following the policy, which can be either stochastic or deterministic, denoted by $a^{(t)} \sim \pi(.|s^{(t)})$ or $a^{(t)} = \mu(s^{(t)})$, respectively. For the stochastic case, the policy function should meet the condition $\sum_{a^{(t)} \in A} \pi(a^{(t)}|s^{(t)}) = 1$, where A is the action space. After taking action $a^{(t)}$, the environment transitions to a new state $s^{(t+1)}$ following an unknown transition matrix P that links state-action pairs to a distribution of new states. The agent moves from state $s^{(t)}$ to $s^{(t+1)}$ while getting a reward $r^{(t)}$. An experience is gathered at time t that is a tuple in the form of $e^{(t)} = (s^{(t)}, a^{(t)}, r^{(t)}, s^{(t+1)})$. Model-free reinforcement learning learns from interacting with the environment without any further information on the transition probabilities.

Q-learning algorithm finds a policy to maximize the cumulative future reward. The Q-function is the expected future reward when an action is taken under the policy π, which is written as

$$Q^{\pi}(s, a) = E_{\pi}[R^{(t)}|s^{(t)} = s, a^{(t)} = a], \tag{2.18}$$

where $R^{(t)}$ is the future discounted reward defined as

$$R^{(t)} = \sum_{\tau=0}^{\infty} \eta^{\tau} r^{(t+\tau)}, \tag{2.19}$$

where $\eta \in (0, 1]$ is the discount factor. The conventional Q-learning algorithm builds a lookup table of Q-values. Once the Q-values are randomly initialized, the agent chooses actions based on ϵ-greedy policy for each time step t. The ϵ-greedy policy implies that the agent takes action a', which maximizes the expected reward, with a probability of $1-\epsilon$, and exploits the environment. On the other hand, it explores the environment by taking a random action with the probability ϵ. A decaying ϵ-greedy algorithm, in which the ϵ value decreases after each iteration, is used to increase the

number of exploitations over time.

The complexity of the conventional Q-learning method increases as the size of the state space increases. A Deep Q-network (DQN) overcomes this problem by employing a deep neural network (DNN) to represent Q-function, which is expressed as $Q(s, a; \theta)$, in which θ is the network parameter. Like conventional Q-learning, a DQN collects experience by taking an action in a specific state. An experience replay buffer $\mathcal{D}$ stores different experiences in the memory. A mini-batch $\mathcal{E}^{(t)} = \{e_1^{(t)}, e_2^{(t)}, ..., e_{\mathcal{B}}^{(t)}\}$ is sampled from the experience replay buffer with length $\mathcal{B}$, where $e_i^{(t)}$ is the i-th experience sampled from the experience replay buffer. Sampling is performed in order to remove correlations in the observations and flatten changes in the data distribution.

In Double Deep Q-Networks (DDQN), two DQNs are defined by using the "quasi-static target network" method: The target network with parameters $\theta_{target}^{(t)}$ and the train network with parameters $\theta_{train}^{(t)}$. $\theta_{target}^{(t)}$ is updated with a soft update method every time frame such that

$$\theta_{target}^{(t)} = \tau\theta_{train}^{(t)} + (1-\tau)\theta_{target}^{(t)}, \tag{2.20}$$

where $\tau \ll 1$. The soft update method offers more stability and excludes extreme updates that differ from past experiences. The least squares loss of the train DDQN for a randomly selected mini-batch $\mathcal{E}^{(t)}$ at time t from the experience replay buffer is

$$\mathcal{L}(\theta_{train}^{(t)}) = \sum_{(s,a,r,s')\in\mathcal{E}^{(t)}} (y_{\mathrm{DDQN}}^{(t)}(r, s') - Q(s, a; \theta_{train}^{(t)}))^2 \tag{2.21}$$

where the target is

$$y_{\mathrm{DDQN}}^{(t)}(r, s') = r + \lambda \max_{a'} Q(s', a'; \theta_{target}^{(t)}). \tag{2.22}$$

The agent updates network parameters $\theta_{train}^{(t)}$ by using an optimization technique to minimize the expected prediction error (cost function) of the sampled mini-batch.

The stochastic gradient descent utilizes the computed gradients and has been proven to converge to a set of suitable parameters fast. The gradient update for the network is performed as follows:

$$\theta_{train}^{(t)} \leftarrow \theta_{train}^{(t)} - \alpha^{(t)} \nabla_{\theta_{train}^{(t)}} \mathcal{L}(\theta_{train}^{(t)}), \tag{2.23}$$

where $\alpha^{(t)} \in (0, 1)$ is the learning rate of the gradient update.

Dueling Double Deep Q-Network (D3QN) represents a neural network structure specifically created to facilitate value-based reinforcement learning [Hessel et al., 2018]. The fundamental idea driving the new architecture is that, for many states, there is no need to compute the value of each action choice. The dueling network incorporates two separate modules, namely the value and advantage estimator functions. The value function determines the reward from the state, and the advantage function shows how much the chosen action is better than the others. These estimator functions are combined using a unique aggregator mechanism to calculate the Q-function. It is crucial to emphasize that the aggregator is considered and implemented as an integral part of the network rather than a distinct algorithmic step. Since the algorithm consists of two branches in the final layer, it can capture the environmental effects accurately.

Proposed DRL Algorithm

We propose centrally-trained-centrally-executed multi-agent deep reinforcement learning model, as illustrated in Figure 2.2, where each sensor node operates as an agent with its local network in the central server. The method promotes stability through the training of global policy parameters that are shared across the network. This training is facilitated by a centralized trainer, which accumulates experiences from all agents. The changes in the states of the environment in the multi-agent learning system are dependent on the joint actions of the agents. All the agents work in a synchronized way and take their actions simultaneously. Each agent i observes the state $s_i^{(t)}$ and chooses an action $a_i^{(t)}$ using a local neural network and gets the re-

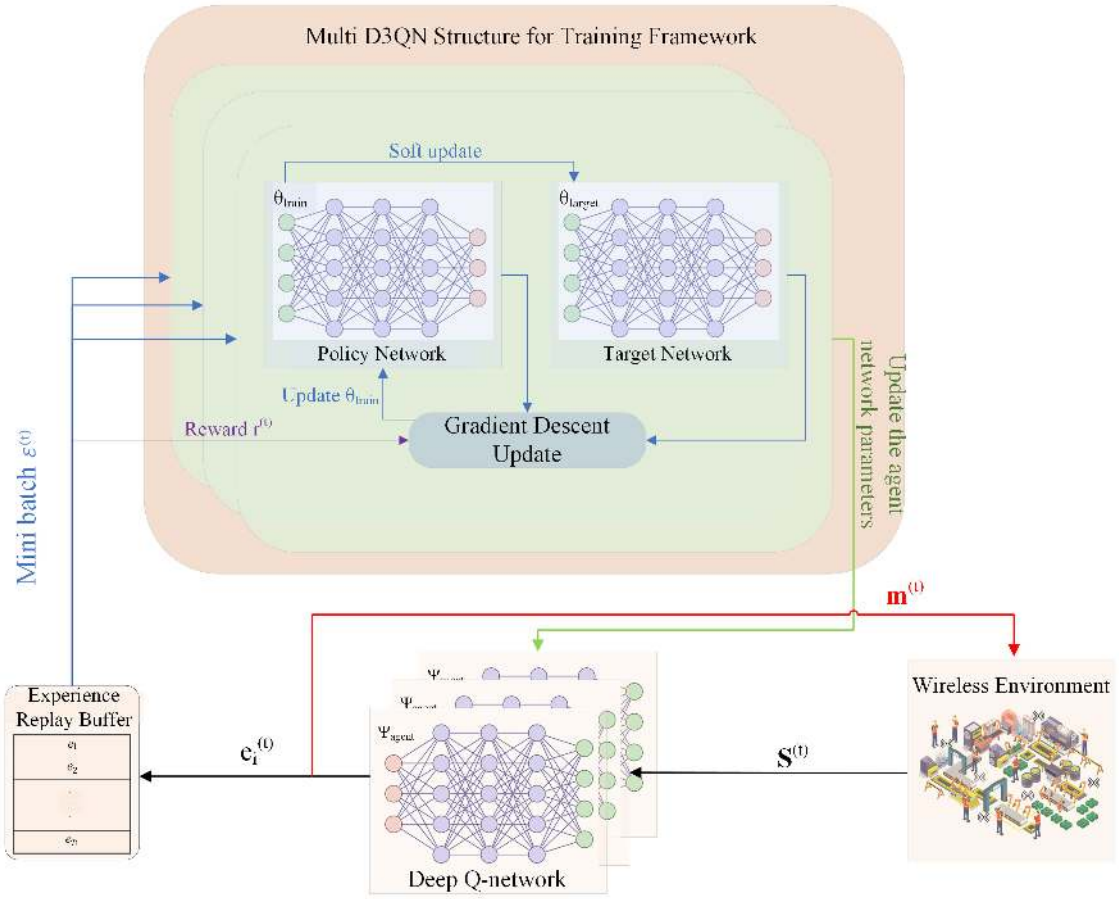


Figure 2.2: Proposed DRL algorithm.

ward $r_i^{(t)}$. Then, the experience of the agents is collected into a replay buffer, which works in a FIFO (first in, first out) fashion. Training is done based on the batches sampled from the buffer to update the network parameters. Since the experience replay buffer is the only memory unit in the structure and has a direct impact on the learning process, its design plays a critical role in detecting and discarding outdated data to adapt to the fast-changing environment. A Prioritized Experience Replay (PER) buffer [Schaul et al., 2015] is used to prioritize the most relevant experience to discard the outdated experience and avoid retraining. D3QN is implemented as the learning scheme for the agents. The elements of the implementation are described below.

- *States:* States are the variables that directly have an impact on the environ-

ment. The next state of each agent is determined by the joint actions of all agents. The state of agent i is given by

$$s_i^{(t)} = (\mathbf{m}^{(t-1)}, R_i^{(t-1)}, \mathbf{W}^{(t-1)}, \mathcal{P}^{(t-1)}, \gamma_i^{(t)}, \mathbf{g}^{(t)}). \tag{2.24}$$

where $\mathbf{m}^{(t-1)}$ is the blocklength vector in the previous time step, where the ith element of the vector corresponds to the chosen blocklength of the ith node, $m_i^{(t-1)}$; $R_i^{(t-1)}$ is the rate of node i in the previous time step; $\mathbf{W}^{(t-1)}$ is the transmitted power vector in the previous time step, where the power of node i is the ith element of the transmitted power vector, $W_{tx,i}^{(t-1)}$; $\mathcal{P}^{(t-1)}$ is the total power consumption of the nodes; $\gamma_i^{(t)}$ is the SNR calculated based on the instantaneous channel gains and the power level in the previous time step, i.e., $\gamma_i^{(t)} = \frac{g_i^{(t)} W_{tx,i}^{(t-1)}}{\sigma^2}$; and $\mathbf{g}^{(t)}$ is the instantaneous channel gain vector, where the ith element corresponds to the channel gain of node i, $g_i^{(t)}$. The parameters from the previous time step are used to form a Markov Decision Process (MDP) and constrain the model further for the blocklength. Rate and SNR are included to get as much information as possible about the channel condition.

- *Action:* The agent i determines the blocklength $m_i^{(t)}$ to send. All agents have the same action space given by

$$A = \{1, 2, ..., M_{th}\}, \tag{2.25}$$

which is determined from the optimization problem constraint related to the maximum blocklength available for an agent.

- *Reward:* The reward function is the central core of the learning process in RL. Each agent takes optimal action to maximize the total reward. The reward function has two parts. The first part corresponds to the objective function of

the optimization problem (17a) and is given by

$$r_g^{(t)} = -\sum_{i=1}^{N} \left(C_{i1}^{(t)} C_2 \frac{m_i^{(t)} k_i^{*(t)}}{\Omega} \left[\exp\left(\frac{Q^{-1}(1-\delta)^{\frac{1}{k_i^{*(t)}}}}{\sqrt{m_i^{(t)}}} + \frac{\ln(2) L_i}{m_i^{(t)}} \right) - 1 \right] + C_2 W_i^c \frac{m_i^{(t)} k_i^{*(t)}}{\Omega} \right), \tag{2.26}$$

where $C_{i1}^{(t)} = \frac{\sigma^2}{|g_i^{(t)}|}$. The second part of the reward function corresponds to the schedulability constraint (17c). If the agents violate the schedulability constraint, they will get penalized based on their distance to the satisfaction of this constraint. On the other hand, a small positive reward is gained if the agents satisfy the constraint. The second part is then written as

$$r_{i,l}^{(t)} = \begin{cases} r_{pos}, & \sum_{j=1}^{N} \frac{C_{2j} m_j^{(t)} k_j(m_j^{(t)})}{\Omega} \leq \beta \\ r_{neg}, & \text{otherwise}, \end{cases} \tag{2.27}$$

where $r_{pos} \in \mathbb{R}^+$ is a small positive constant reward for not violating the constraint, and $r_{neg} = \mathcal{C}(\sum_{j=1}^{N} \frac{C_{2j} m_j^{(t)} k_j(m_j^{(t)})}{\Omega} - \beta)$ is the penalty for violating the constraint condition where $\mathcal{C} \in \mathbb{R}^-$ is a constant. Finally, the reward function is the weighted combination of these two parts given by

$$r^{(t)} = w r_g^{(t)} + (1-w) \sum_{i=1}^{N} r_{i,l}^{(t)}, \tag{2.28}$$

where $w \in (0, 1]$ is the weight factor balancing the contribution of the utility from the problem objective function and the cost from the unsatisfied schedulability constraint. The values of w, r_{pos}, and $\mathcal{C}$ are hyperparameters and are tuned to produce the best learning outcome. The design of the reward function plays an important role in detecting outdated models. In a fast-changing

wireless environment, an outdated model tends to choose suboptimal actions, incurring higher penalties and yielding lower rewards. Therefore, outdated models can be detected by assessing their reward in a pre-defined time window.

Proposed DRL Framework

In this part, the proposed centrally-trained-centrally-executed DRL based algorithm for solving the resource allocation problem is presented. The proposed multi-agent DRL based resource allocation algorithm is summarized in Algorithm 1, which is comprised of three parts, namely, initialization, random experience accumulation, and training and execution.

In the initialization part, N local D3QNs $\mathcal{Q}(s_i^{(t)}, a_i^{(t)}; \Psi_{i,agent}^{(t)})$ with network parameters $\Psi_{i,agent}^{(t)}$ are constructed based on the architecture of the proposed framework shown in Figure 2.2. Additionally, N train D3QNs $\mathcal{Q}(s_i^{(t)}, a_i^{(t)}; \theta_{i,agent}^{(t)})$ and N target D3QNs $\mathcal{Q}(s_i^{(t)}, a_i^{(t)}; \bar{\theta}_{i,agent}^{(t)})$ with parameters $\theta_{i,agent}^{(t)}$ and $\bar{\theta}_{i,agent}^{(t)}$ are, respectively, established. The weights of each network are initialized randomly, and the experience replay buffer $\mathcal{D}$ is initialized with its corresponding parameters (Lines 1-2).

In the random experience accumulation phase, at the beginning of time frame t, each node $i(\forall i \in \{1, ..., N\})$ observes a local state $s_i^{(t)}$ and transmits it to its corresponding local network in the central server. Upon transmitting the information to the central server, each network randomly chooses a blocklength action $m_i^{(t)}$ to execute. The local network stores its local experience $e_i^{(t)}$ in the experience replay buffer $\mathcal{D}$ (Line 3). After collecting at least $\mathcal{B}$ experience samples from the agents, a mini-batch of size $\mathcal{B}$ is sampled and utilized to compute the loss value in (2.21). The loss is minimized using the gradient update method in (2.23) (Lines 4-5). After the gradient update, train network parameters $\theta_{i,agent}^{(t)}$ are updated in that time step, and target network is updated via a soft update method using (2.20) in that same time frame (Lines 6-7). Upon updating the training parameters in the central server, weights of the train network are sent to each local network in the central core every

Algorithm 1 Proposed Multi-Agent DRL Based Resource Allocation Algorithm

Initialization:

1: Initialize the experience replay buffer, $\mathcal{D}$.

2: For each node $i \in \{1, ..., N\}$, randomly initialize the local D3QN network $\mathcal{Q}(s_i^{(t)}, a_i^{(t)}; \Psi_{i,agent}^{(t)})$, the train D3QN network $\mathcal{Q}(s_i^{(t)}, a_i^{(t)}; \theta_{i,agent}^{(t)})$, and the target D3QN network $\mathcal{Q}(s_i^{(t)}, a_i^{(t)}; \bar{\theta}_{i,agent}^{(t)})$ with weights $\Psi_{i,agent}^{(t)}$, $\theta_{i,agent}^{(t)}$, and $\bar{\theta}_{i,agent}^{(t)}$, respectively.

Random experience accumulation:

3: At the beginning of the time frame t, each node $i(\forall i \in \{1, ..., N\})$ observes its local state $s_i^{(t)}$ and transmits it to its corresponding network in the central server, and the server randomly chooses a blocklength $m_i^{(t)}$, and accumulates the local experience $e_i^{(t)}$ in the memory buffer $\mathcal{D}$.

4: Upon storing at least $\mathcal{B}$ experiences from different agents, a batch of size $\mathcal{B}$ is sampled from the replay buffer.

5: The sampled mini-batch is used to compute the loss in (2.21) and minimize it using an optimization technique.

6: Update the train network parameters $\theta_{i,agent}^{(t)}$ using (2.23).

7: Update the target network parameters $\bar{\theta}_{i,agent}^{(t)}$ using (2.20).

8: Update D3QN parameters $\theta_{i,agent}^{(t)}$ to its corresponding local D3QN every time frame to update the weights $\Psi_{i,agent}^{(t)}$.

Training and execution:

9: Each node $i(\forall i \in \{1, ..., N\})$ observes a local state $s_i^{(t)}$ and transmits the local observations to its corresponding network in the central core. The network executes the blocklength action $a_i^{(t)}$ following the ϵ-greedy algorithm and stores the local experience $e_i^{(t)}$ in the replay buffer $\mathcal{D}$.

10: A mini-batch of $\mathcal{B}$ experience is sampled from the buffer $\mathcal{D}$ to minimize the loss function in (2.21).

11: The train network parameters $\theta_{i,agent}^{(t)}$ are updated using (2.23).

12: The target network parameters $\bar{\theta}_{i,agent}^{(t)}$ are updated using the soft update method in (2.20).

13: Update D3QN parameters $\theta_{i,agent}^{(t)}$ to its corresponding local D3QN every time frame to update the weights $\Psi_{i,agent}^{(t)}$.

14: Actions taken by the agents are sent back to each node to be executed.

time frame to update the local network of each node (Line 8).

In the training and execution phase, agents execute their actions based on ϵ-greedy policy rather than just choosing random actions. Each node observes a local state $s_i^{(t)}$ from the wireless environment and transmits it to the central server, and the network of each node executes a blocklength action and stores the local

experience gathered from that interaction in the global replay memory buffer, and for the learning part, a mini-batch $\mathcal{B}$ is sampled to be processed to minimize the loss function (Lines 9-10). After minimizing the loss with the gradient update method used in (2.23), train network parameters are adjusted, and the target network is updated using the same soft update method used in random experience accumulation (Lines 11-12). Finally, the parameters of the train network are utilized to update the parameters of the local agents, and the chosen actions in the execution phase are sent back to each node to be executed in the wireless environment (Lines 13-14).

2.6 Performance Evaluation

In this section, we evaluate the performance of the proposed optimization theory based DRL algorithm in comparison to the optimization based and pure DRL based benchmarks. In the optimization theory based benchmark, we develop an efficient approximation algorithm based on the further analysis of the optimality conditions, the relaxation of the resulting integer optimization problem and then searching of the integral solution by using a greedy algorithm, similar to the one proposed in [Sadi et al., 2014] but adapted to the finite blocklength information theory. In the pure DRL approach, we adopt Branching Dueling Q-networks (BDQ) as the learning model to solve the optimization problem (2.13) without further analysis of the optimality conditions. There are three kinds of decision variables to optimize in order to have an optimal solution. To handle all three actions, continuous actions are discretized into L levels. Applying DRL algorithms to high-dimensional action spaces increases the number of action dimensions. To overcome the problem of the complexity of commonly used sequential architectures, BDQ is proposed in [Tavakoli et al., 2018] based on the usage of a shared decision model and several branching networks for each individual action dimension. The architecture of BDQ allows the selection of joint actions in a multidimensional setting without increasing the complexity of the network. Additionally, the DDQN approach serves as an additional benchmark to compare the dueling and double architectures, and show the efficacy of the D3QN algorithm across diverse scenarios.

2.6.1 Simulation Setup

Simulations are performed for a network where nodes are uniformly distributed within a circular area of a radius of 50 meters, communicating with a central controller. The links between the sensor nodes and the central controller are impaired by both large-scale and small-scale fading. The large-scale fading α_i occurs due to path loss and shadowing and is modeled as $PL(d_i) = PL(d_0) + 10\alpha \log(\frac{d_i}{d_0}) + Z$ (in dB), where d_i is the distance of node i from the central controller, $PL(d_i)$ is the path loss of node i at distance d_i from the controller measured in decibels, $PL(d_0) = 35.3$ dB is the path loss at reference distance $d_0 = 1$ m, path loss exponent $\alpha = 3.76$ [Access, 2010], Z is a Gaussian random variable with zero mean and standard deviation equal to 4 dB corresponding to log-normal shadowing. For the small-scale fading, Jake's model [Liang et al., 2017] is deployed, expressed as a first-order complex Gauss-Markov process:

$$f_i^{(t)} = \rho f_i^{(t-1)} + \sqrt{1-\rho^2} e_i^{(t)}, \tag{2.29}$$

where $f_i^{(t)}$ and $e_i^{(t)}$ are channel coefficient and channel innovation process of node i at time t, respectively; ρ is the correlation coefficient. $f_i^{(0)}$ and the channel innovation process $e_i^{(1)}, e_i^{(2)}, ...$ are independent and identically distributed circularly symmetric complex Gaussian (CSCG) random variables with unit variance. ρ is set to 0.6. The channel gain of node i at time frame t is then computed by

$$g_i^{(t)} = |f_i^{(t)}|^2 \alpha_i, \quad t = 1, 2, ... \tag{2.30}$$

The noise power spectral is $\sigma^2 = -174$ dBm/Hz. The parameters used in the simulations are given in Table 2.2.

2.6.2 DRL Experimental Framework

We next describe the neural network design and hyperparameters used for the architecture of our algorithm. Since our goal is to ensure that the agents make their decisions as quickly as possible, we do not over-parameterize the network archi-

Table 2.2: Simulation parameters

Parameter	**Value**	**Parameter**	**Value**
B	100 kHz	M_{th}	200 Symbols
L_i	100 bits	δ	0.99
Δ	1 ms	Ω	100 ms
W_{max}	250 mW	W_c	54 mW
N	50	σ^2	-174 dBm/Hz

tecture, and we use a relatively small network for training purposes. The optimal hyperparameters are selected by using the grid search algorithm, which systematically explores a grid of all possible combinations of hyperparameter values. Our algorithm trains a DDQN and D3QN with one input layer, three hidden layers, and one output layer. The hidden layers have $N_1 = 32$, $N_2 = 64$, and $N_3 = 300$ neurons, and the input layer has $3N + 3$ neurons. The leaky ReLU activation function is utilized to prevent the vanishing gradient problem. Furthermore, the activation function for the input and output layers is a linear function. In DDQN, the number of neurons in the last layer is equal to the number of possible actions for every agent, denoted by N_m, which is equal to M_{th}. On the other hand, in D3QN, the final layer corresponds to the output of the advantage and state value functions. The number of neurons for the final layer is denoted by N_A and N_V, which are equal to M_{th} and 1, respectively. A mini batch of size 32 is sampled for the training phase for the all algorithms.

The pure DRL-based approach utilizes the BDQ algorithm to train and interact with the environment. In the output layer, the number of neurons dedicated to each output is chosen by the number of actions available for each output layer to take. The number of actions for the blocklength, sampling period, and packet error probability are N_m, N_h, and N_p, and are set to 200, 512, and 512, respectively. The size of the input layer is $5N + 3$. The Leaky ReLU is chosen as the activation function of the hidden layers, and a linear activation function is used for the input and output layers. The mini-batch size is set to 32.

We use the RMSprop algorithm with an adaptive learning rate $\alpha^{(t)}$. For a more stable outcome, the learning rate is reduced as $\alpha^{(t+1)} = (1 - \lambda)\alpha^{(t)}$, where $\lambda \in (0, 1)$

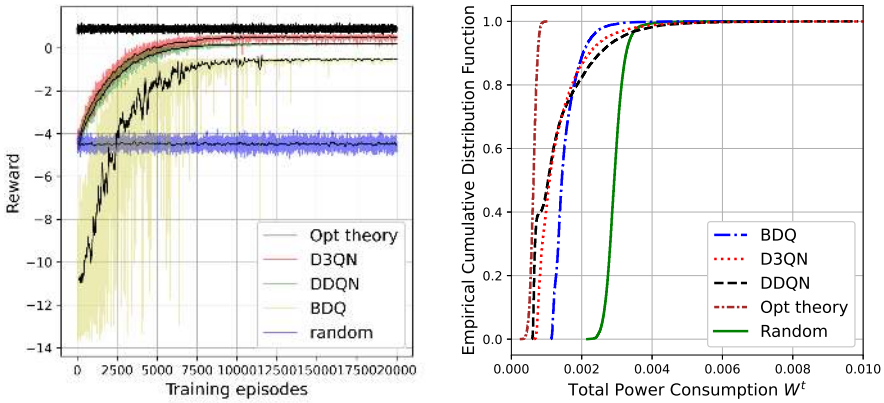
is the decay rate of $\alpha^{(t)}$. Here, $\alpha^{(0)}$ is 0.03 and λ is 10^{-3}. We also apply an adaptive ϵ-greedy algorithm: $\epsilon^{(0)}$ is initialized to 1, and it follows $\epsilon^{(t+1)} = (1-\beta)^{(t)}\epsilon^{(0)}$, where $\beta = 10^{-4}$. Discount factor (η) in DRL setting is set to a moderate value of 0.666, since the connection between the actions taken by the agent and its forthcoming rewards tends to diminish due to fading, resulting in a lower correlation. Policy and target networks are updated via the soft update method, with $\tau = 0.001$.

The proposed algorithm is implemented in PyTorch [Paszke et al., 2019]. Training is done by building the proposed architecture with the network parameters introduced in this section. Each testing result is an average of 10 randomly initialized simulations of $2,500$ episodes.

2.6.3 Performance comparison and analysis

Figure 2.3a shows the training convergence of the reward for different algorithms in a network of 50 nodes with $\Omega = 0.1$ and $\Delta = 0.001$. The proposed and benchmark algorithms are shown to perform better than random selection, which selects random actions in each time frame. The conventional optimization technique has the best performance compared to cutting-edge DRL methods. However, in order to have optimal results, full CSI is needed to generate results, whereas DRL methods can use delayed or non-perfect CSI. DDQN method outperforms BDQ in terms of getting higher rewards and converging faster in the training phase because the utilization of the optimization theory to reduce the constraints and decision variables has facilitated for the DRL model to learn the environment model. Due to its robust architecture and its ability to capture patterns in a stochastic environment, D3QN performs the best among the DRL algorithms in terms of converging and getting sub-optimal rewards.

In Figure 2.3b, test results are shown for the proposed and benchmark algorithms. Testing is done based on the saved parameters of the trained networks in the training phase. The proposed optimization theory based DRL algorithms exhibit superior performance when compared to both benchmark approaches and random method. Nevertheless, they fall short of surpassing the capabilities of optimization



(a) (b)

Figure 2.3: Training and testing results for different algorithms in a network of 50 nodes with $\Omega = 0.1$ and $\delta = 0.001$.

theory. D3QN outperforms DDQN in the testing part, which validates the training convergence graph. Similar to the training, both benchmark and proposed algorithms perform better than random selection. Similar to the training results, BDQ algorithm cannot outperform D3QN and DDQN model due to its complexity and joint action selection feature.

Effect of MATI

In this part, we focus on the impact of varying the MATI value on the performance of the algorithms in the training and testing phase. Figure 2.4a shows the training convergence of the reward for different algorithms in a network of 50 nodes with $\delta = 0.001$ and $\Omega = 0.06$. Similar to the previous results, all of the proposed and benchmark algorithms outperform the random-selection method. Similarly, D3QN outperforms DDQN, which outperforms BDQ. The fluctuations of DDQN and D3QN are less than that of BDQ, demonstrating their robustness. The BDQ algorithm fluctuates more than that in Figure 2.3a because of the more strict stochastic MATI constraint, which increases the cost function while training. Furthermore, the proposed algorithm performs very close to the pure optimization theory based approach. However, the performance of the proposed D3QN is within 25% of that of the optimization theory based approach compared to 10% in Figure 2.3b, due to the more

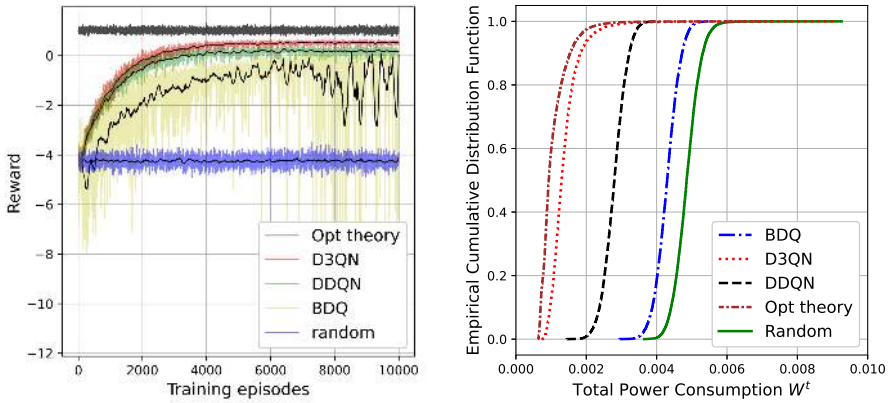


Figure 2.4: Training and testing results for different algorithms in a network of 50 nodes with $\Omega = 0.06$ and $\delta = 0.001$.

strict stochastic MATI constraint.

Figure 2.4b shows the testing part of different algorithms. D3QN and DDQN approaches significantly outperform random selection and BDQ algorithm. When the MATI condition is strict, there is no significant difference between BDQ and the random selection method, which leads to high fluctuations in the last episodes of the training phase. Lowering the MATI value leads to a rise in power consumption aimed at ensuring successful transmission under reduced MATI conditions. Although the gap between the convergence rate of the D3QN and the DDQN is not large, testing results clearly demonstrate the advantage of the dueling architecture and splitting the final layer into two branches to further refine the model in choosing optimal actions after training.

Effect of MAD

Next, we analyze the impact of varying MAD value on the performance of the algorithms in the training and testing phases. Figure 2.5a shows the training convergence of the reward in a network of 50 nodes with $\Omega = 0.1$ and $\delta = 0.01$. The increase in the MAD value relaxes the problem, which facilitates determining the optimal course of action. Consequently, the performance of BDQ is closer to that of DDQN. On the other hand, D3QN outperforms DDQN, and the disparity between them is

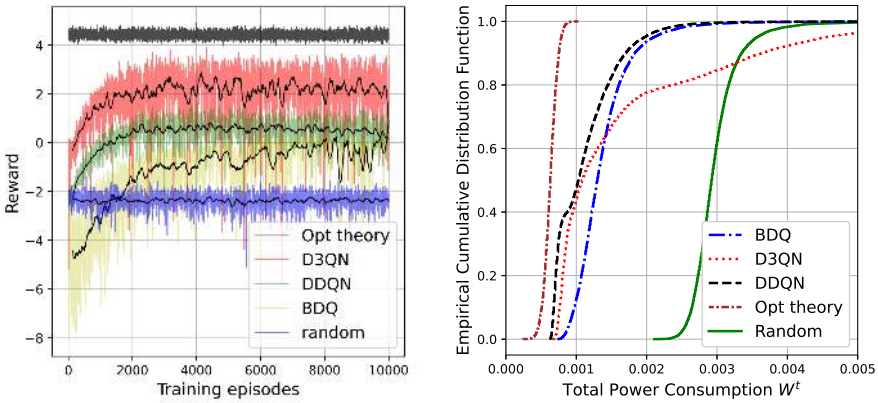


Figure 2.5: Training and testing results for different algorithms in a network of 50 nodes with $\Omega = 0.1$ and $\delta = 0.01$.

more pronounced compared to prior scenarios. Moreover, the gap between the outcomes of optimization theory and the proposed D3QN method is more significant than in previous results (close to 50% compared to 20% of that in Figure 2.3a). This discrepancy arises from the fact that optimization results, assuming perfect CSI, yield suboptimal outcomes when the conditions within the system model are less stringent. However, DRL-based algorithms are less adept at efficiently capturing this effect than optimization methods when the conditions change slightly.

Figure 2.5b shows the testing phase of the different algorithms with similar settings as the training part. Since the MAD constraint is more relaxed, the power consumption values are smaller, and there is no need to increase power in the control system. On one side, akin to previous situations, the suggested techniques demonstrate the capability to surpass random-selection and benchmark schemes. Conversely, the BDQ algorithm is able to perform near to D3QN and DDQN algorithms when compared to the previous results. As opposed to the previous results, D3QN results fluctuate more than DDQN total power consumption results. The reason is that with a more relaxed scenario, an intricate approach like D3QN can overfit in the training phase, preventing the extraction of the best outcome in the testing part.

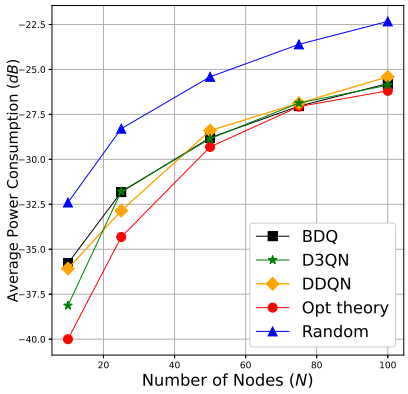


Figure 2.6: Testing results of different algorithms as a function of the number of nodes in a network with $\delta = 0.001$ and $\Omega = 0.1$.

Effect of the Number of nodes

We analyze the effect of the number of nodes on the performance of the proposed algorithms. Figure 2.6 shows the average power consumption of the algorithms for different numbers of nodes in a network with $\delta = 0.001$ and $\Omega = 0.1$. The small gap between DRL-based models and optimization theory-based model indicates the capability of the DRL-based models in choosing the right action across diverse scenarios. Moreover, as the number of nodes increases, the gap diminishes, demonstrating the scalability of the proposed method without compromising the performance. In contrast, random selection consistently exhibits inferior performance, with the performance gap widening as the number of nodes increases.

Figure 2.7 shows the average execution time of the algorithms. This execution time only includes the operation time of the pure optimization theory-based and DRL-based algorithms, where DRL-based methods are trained on the environment and execute optimal actions. A substantial disparity exists between the optimization method and DRL-based approaches as the number of nodes increases. This gap indicates that DRL-based methods can be effectively harnessed for real-time applications with close to optimal results. Additionally, the operation time difference between DDQN and D3QN is negligible, indicating that splitting the network does not introduce additional time and computational complexity. Instead, it contributes

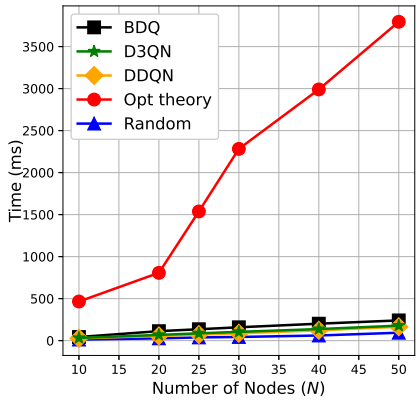


Figure 2.7: Average execution time (ms) per step for different algorithms and number of nodes.

to enhancing the performance of the agents.

2.7 Conclusion

In this chapter, we propose a novel optimization theory based DRL framework for the joint optimization of control and communication systems with the goal of minimizing power consumption while considering ultra-reliable transmission in the finite blocklength regime. Following the formulation of the optimization problem based on the abstraction of the control and communication systems, the proposed methodology derives the optimality conditions to decompose the problem into multiple building blocks. Then, the building blocks that are not tractable are replaced by DRL, which employs DDQN and D3QN to choose optimal actions. The proposed optimization theory based DRL algorithms, DDQN and D3QN, have been demonstrated to outperform the benchmark pure DRL-based algorithm, BDQ, and the random selection method while performing very close to the conventional optimization theory based approach for different network sizes, MATI, and MAD values. In the future, we plan to incorporate alternative control system abstractions, such as Age-of-Information, Age-of-Loop, into the proposed optimization theory based deep learning solution methodology and analyze the interpretability and robustness of the proposed algorithms through the development of explainable AI techniques. Moreover, we plan to investigate the usage of Meta-Learning to reduce the transient

time of the learning scheme for better adaptability in a fast-changing wireless environment. Another interesting research direction for future work would be to extend our results to encompass resource allocation for massive MIMO systems.

Chapter 3

DIFFUSION MODEL BASED RESOURCE ALLOCATION STRATEGY IN ULTRA-RELIABLE WIRELESS NETWORKED CONTROL SYSTEMS

The content of this chapter has been submitted to IEEE Communication Letters (COMML).

3.1 Overview

Diffusion models are vastly used in generative AI, leveraging their capability to capture complex data distributions. However, their potential remains largely unexplored in the field of resource allocation in wireless networks. This chapter introduces a novel diffusion model-based resource allocation strategy for Wireless Networked Control Systems (WNCSs) with the objective of minimizing total power consumption through the optimization of the sampling period in the control system, and blocklength and packet error probability in the finite blocklength regime of the communication system. The problem is first reduced to the optimization of blocklength only based on the derivation of the optimality conditions. Then, the optimization theory solution collects a dataset of channel gains and corresponding optimal blocklengths. Finally, the Denoising Diffusion Probabilistic Model (DDPM) uses this collected dataset to train the resource allocation algorithm that generates optimal blocklength values conditioned on the channel state information (CSI). Via extensive simulations, the proposed approach is shown to outperform previously proposed Deep Reinforcement Learning (DRL) based approaches with close to optimal performance regarding total power consumption. Moreover, an improvement of up to eighteen-fold in the reduction of critical constraint violations is observed, further underscoring the accuracy of the solution.

3.2 Introduction

WNCSs are control systems with control loops closed through a wireless communication network [Park et al., 2018a]. WNCSs play an important role in supporting emerging applications in sixth-generation (6G) networks, such as remote driving [Kakkavas et al., 2024] and cooperative robots (cobots) [Kerboeuf et al., 2024]. The joint optimization of the performance of the control and communication systems is the main challenge in WNCSs due to the high complexity of modeling the interactions between these two systems, the stringent ultra-reliability requirements of control systems, and the non-ideal propagation characteristics of wireless communication systems.

Earlier research on the joint design of control and communication systems for WNCSs focused on the usage of optimization theory-based solution strategies following the derivation of the right abstractions of these two systems [Sadi and Coleri Ergen, 2017, Cao et al., 2023]. In joint optimization frameworks that uniquely abstract the performance of control and communication systems, an optimization problem is formulated with both the control and communication decision variables and solved using iterative or heuristic methods following the derivation of the optimality conditions. However, the high complexity of the proposed model-based methods may hinder their application in low-latency WNCS scenarios.

In order to solve the time complexity issue of the optimization theory-based solutions and additionally operate with incomplete CSI, DRL is introduced to allocate resources optimally through trial and error in a complex scenario [Zhao et al., 2024]. In our previous work [Ali et al., 2024], we have proposed an optimization theory-based DRL approach, where first, the model is simplified based on the optimality conditions of the optimization problem. Then, the new simplified problem is fed to a Dueling Double Deep Q-network (D3QN) to output the optimal block-length values. Although DRL methods use online interactions with the environment to learn, the amount of data needed to train the model is large. Moreover, DRL models may produce infeasible solutions that violate the constraints of the control and communication systems, which may have detrimental effects on critical safety

applications.

Generative AI-based approaches are considered a solution to enhance the performance of 5G/6G networks. [Kasgari et al., 2020] integrates Generative Adversarial Networks (GANs) with a DRL model to improve training and adaptability in extreme conditions over traditional DRL models. However, the GAN used in the proposed model is solely used to generate data to pre-train the DRL model and is not involved in the decision-making process. Additionally, GANs are known to be unstable in training time and cannot generate high-quality samples in the inference phase. To fix the instability and poor quality of the generated samples, DDPMs introduced in [Ho et al., 2020] are proposed to be utilized in [Letafati et al., 2023] to enhance the performance of the wireless networks in constellation shaping problem. However, to this day, diffusion models are not directly applied to resource allocation problems in wireless networks.

In this chapter, we propose a novel DDPM-based resource allocation scheme for the joint design of control and communication systems in WNCSs for the first time in the literature. The DDPM is used to generate optimized blocklength values from an isotropic Gaussian distribution by using the CSI as conditional information.

3.3 System Model

The WNCS consists of $\mathcal{N}$ sensor nodes with blocklength m_i, sampling period h_i, and packet error probability p_i for $i \in \{1, 2, \ldots, \mathcal{N}\}$. Sensor nodes connected to a physical plant measure and send the plant's state to a controller via a wireless channel. Based on the recent state update information, the controller decides on a new control command and sends it to the actuator to be executed. The outdated packets are not retransmitted since old state information can harm time-critical control systems. The packet error is modeled as a Bernoulli random process to simplify the problem. The Time Division Multiple Access (TDMA) method is utilized for a deterministic access delay widely preferred in various automation applications [ANSI, 2011]. We assume that the channel time is segmented into frames, and each frame is subdivided into time slots. The initial slot is allocated for the beacon frame, which

the controller sends out periodically to disseminate synchronization and scheduling updates among the nodes within the WNCS. During the scheduling update, nodes are allocated time slots for their respective data transmissions and additional parameters, such as the optimal transmission power and blocklength. We assume that nodes within the network do not transmit simultaneously and that the network manager continuously monitors the packet error rate.

3.3.1 Optimization of Control and Communication Systems

The joint optimization of control and communications systems for ultra-reliable communication in the finite blocklength (FBL) regime is adopted from our previous paper [Ali et al., 2024] and is presented below.

$$\underset{\substack{h_i,m_i,p_i \\ i\in[1,\mathcal{N}]}}{\text{minimize}} \quad \sum_{i=1}^{\mathcal{N}} C_{i1}C_2\frac{m_i}{h_i}\Big[\exp\Big(\frac{Q^{-1}(p_i)}{\sqrt{m_i}}+\frac{\ln 2L_i}{m_i}\Big)-1\Big] + \frac{C_2W_i^c m_i}{h_i} \tag{3.1a}$$

subject to

$$\lfloor\frac{\Omega}{h_i}\rfloor \ln p_i - \ln(1-\delta) \leq 0,\ \forall i \in [1,\mathcal{N}] \tag{3.1b}$$

$$0 < d_i(m_i) \leq \min(\Delta, h_i),\ \forall i \in [1,\mathcal{N}] \tag{3.1c}$$

$$0 < h_i \leq \Omega, \forall i \in [1,\mathcal{N}] \tag{3.1d}$$

$$0 < p_i \leq 1, \forall i \in [1,\mathcal{N}] \tag{3.1e}$$

$$m_i \leq M_{th},\ \forall i \in [1, \mathcal{N}] \tag{3.1f}$$

$$C_{i1}\Big[\exp\Big(\frac{Q^{-1}(p_i)}{\sqrt{m_i}} + \frac{\ln 2L_i}{m_i}\Big) - 1\Big] \leq W_{tx,max} \tag{3.1g}$$

$$\sum_{i=1}^{\mathcal{N}} \frac{d_i(m_i)}{h_i} \leq \beta, \tag{3.1h}$$

where $C_{i1} = \frac{\sigma^2}{|g_i|}$ and $C_2 = \frac{1}{B}$; σ^2 denotes the noise power spectral density (PSD); g_i denotes the channel gain of sensor node i; B is the bandwidth; $Q^{-1}(\cdot)$ denotes the inverse of Q function; L_i is the packet length, and W_i^C is the circuit power for node i. The objective function (3.1a) is to minimize the total power consumption in the network considering both the transmit power and the circuit power of the nodes when sending packets. Constraints (3.1b) and (3.1c) represent stochastic MATI (Ω) defined as the probability of maximum allowed time interval between the reception of the state vector reports above MATI being greater than a predefined value δ and MAD (Δ) defined as the maximum packet delay smaller than a maximum limit, respectively, used as an abstraction of the requirements to guarantee a certain control system performance. Constraints (3.1d) and (3.1e) give the lower and upper bounds of the sampling period and packet error probability. Equation (3.1f) states that plant state information is transmitted in small packets using a finite blocklength, which cannot exceed a threshold value M_{th} because the transmission must finish before the maximum allowable channel uses is reached. Additionally, a maximum transmit power constraint in (3.1g) limits the nodes from exceeding a certain transmit power level due to the limited power source of the sensor nodes and government regulations. Moreover, the schedulability constraint (3.1h) ensures that transmission times are assigned to multiple sensor nodes without any two nodes transmitting

simultaneously. Each node i is allocated a fraction of the total schedule length, denoted as $\frac{d_i}{h_i}$. Since no two nodes can transmit simultaneously, the sum of these terms represents the total time allocated to all nodes relative to the schedule length. The problem is a non-convex Mixed-Integer programming problem, so searching for a global optimum solution is difficult [Boyd and Vandenberghe, 2004b].

3.3.2 Simplified Optimization Problem

The problem in (3.1) is not tractable and has to be simplified in order to reach suboptimal results. As a result, the solution is grouped into multiple blocks based on the derivation of the optimality conditions, and the decision variables are reduced to consider blocklength only. Then, the other decision variables can be obtained through optimality conditions.

The optimality conditions are derived in [Ali et al., 2024] as

$$\frac{\Omega}{h_i^*} = \frac{\ln(1-\delta)}{\ln p_i^*} = k_i, \tag{3.2}$$

where k_i is a positive integer. Next, the optimal value of k_i is derived in terms of m_i as

$$k_i^*(m_i) = \max\left[1, \left\lceil \frac{\ln(1-\delta)}{\ln\left[Q\left(\sqrt{m_i}\ln\left(\frac{W_{tx,max}}{m_i C_{i1}}+1\right)-\frac{\ln(2)L}{\sqrt{m_i}}\right)\right]}\right\rceil\right]. \tag{3.3}$$

Then, the problem (3.1) is simplified to reduce the decision variables and the constraints in the problem. The model is optimized using one decision variable of blocklength m_i instead of three decision variables, and the other variables are derived using the optimality conditions described in (3.2) and (3.3). The modified

joint optimization problem is formulated as

$$\begin{aligned}\underset{m_i, i\in[1,\mathcal{N}]}{\text{minimize}} \quad & \sum_{i=1}^{\mathcal{N}} C_{i1} C_2 \frac{m_i k_i^*(m_i)}{\Omega} \\ & \Big[\exp\Big(\frac{Q^{-1}\left((1-\delta)^{\frac{1}{k_i^*(m_i)}}\right)}{\sqrt{m_i}} + \frac{\ln 2 \cdot L_i}{m_i}\Big) \\ & - 1\Big] + \frac{C_2 W_i^c m_i k_i^*(m_i)}{\Omega}\end{aligned} \tag{3.4a}$$

subject to

$$m_i \leq M_{th},\ \forall i \in [1, \mathcal{N}] \tag{3.4b}$$

$$\sum_{i=1}^{\mathcal{N}} \frac{C_2 m_i k_i^*(m_i)}{\Omega} \leq \beta. \tag{3.4c}$$

3.4 DDPM-Based Resource Allocation Algorithm

The proposed diffusion-based resource allocation algorithm is a centrally-trained-centrally-executed model consisting of two stages: optimization theory-based data collection and DDPM-based training and data generation, as depicted in Figure 3.1. In the optimization theory-based data collection stage, various values of channel gains and the corresponding optimal blocklength values are collected. The optimal blocklength values are determined by solving the optimization problem in (3.4). The resulting dataset is then the input to the the diffusion model. In the diffusion model stage, the collected dataset is used to train a diffusion model and learn the optimal parameters to choose an action for blocklength adaptation for given channel gains.

The model is implemented in the controller, where control commands are sent to each sensor node after execution. DDPM consists of input states and conditional information as provided below:

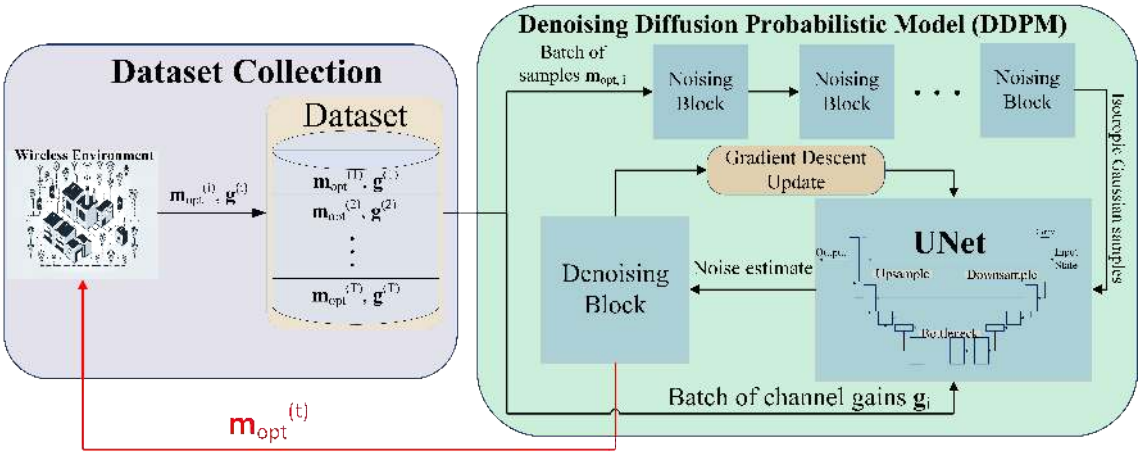


Figure 3.1: The DDPM-based resource allocation algorithm.

- *Input States:* The objective of the DDPM-based method is to generate outputs drawn from a similar distribution to the input states. In the training phase, the input states are the optimal blocklength values from the dataset. Batches of data are sampled from the dataset. They are input to the model to modify and update the parameters of the neural network so the model can learn the solution space distribution and generate desired outputs after the training phase. On the other hand, in the inference phase, the input states are drawn from an isotropic Gaussian distribution with mean zero and standard deviation one, and the outputs are generated through the denoising process of the diffusion models.
- *Conditional Information:* The conditional information given to the network is the CSI of the links to condition the learning process on the environmental variables to ensure the model is trained to execute actions based on the current channel state. This is the most important part of the model since the environment directly affects the learning process. In the training time, in addition to the CSI, uniformly sampled time steps are given to the model to train the model to denoise the samples efficiently.

The proposed algorithm is summarized in Algorithm 2, which is comprised of three parts, namely, initialization and dataset collection from optimization theory-based solution, training phase, and execution phase.

Algorithm 2 Proposed Diffusion Based Resource Allocation Algorithm

Initialization and Dataset Collection from Optimization Theory-based Solution:

1: Initialize network with parameters $\boldsymbol{\theta}$
2: Collect dataset $(\mathbf{m}_{\text{opt}}, \mathbf{g})$ from optimization theory-based solution over $\mathcal{T}$ time frames
3: Set total noising steps T and variance scheduler β_t

Training Phase:

4: For each time frame:
5: Feed batch of blocklength values $\mathbf{m}_i$ as states
6: Sample time step $t \in \{1, \ldots, T\}$
7: Inject noise with variance β_t
8: Pass noisy samples to model with t and channel gains $\mathbf{g}_i$
9: Predict noise $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$
10: Calculate loss $\mathcal{L}_t$
11: Update network parameters
12: Continue training until convergence

Execution Phase:

13: Pass channel gains into model during inference
14: Predict noise and denoise to get optimal blocklength values
15: Broadcast actions to sensor nodes

The network is initialized with parameters $\boldsymbol{\theta}$, and a dataset, which includes the optimal blocklength and channel gain values $(\mathbf{m}_{\text{opt}}, \mathbf{g})$, is collected from the environment solving the optimization problem (3.4). The optimization theory-based solution develops an approximation algorithm based on the analysis of the optimality conditions and the relaxation of the resulting integer optimization problem. Then, it searches for the integral solution using a greedy algorithm developed in [Ali et al., 2024] for a duration of $\mathcal{T}$ time frames. The dataset enables the model to learn the environment and channel states so that it can generate the desired blocklength based on environmental changes. Moreover, the number of time steps T and variance scheduling values $(\beta_t, t \in \{1, \ldots, T\})$, which is an essential part of the DDPM algorithm, for the forward process, are determined based on a scheduling algorithm (Lines 1-3). The choice of scheduling values and time steps play a crucial role in the effective training of the proposed model and should be determined carefully based on the complexity of the model.

After the dataset collection process, at the beginning of the training phase, since the model's objective is to generate actions for nodes to allocate resources, it samples

a mini-batch of blocklength values, denoted as $\mathbf{m}_i$, as the input state and a uniformly sampled time step $t \in \{1, \ldots, T\}$. The batched data are infused with Gaussian noise based on predetermined variance β_t and fed into the model as the input states. The time step t and the batch of channel gains $\mathbf{g}_i$ are given to the model as the conditional information and the time step t is encoded using a positional encoding algorithm to convert discrete values into vectors to enable matrix calculus. Moreover, the channel gains batch $\mathbf{g}_i$ is normalized because small values of channel gains combined with encoded time step t can be negligible, which negatively affects the learning process (Lines 4-8). The model predicts the injected noise $\boldsymbol{\epsilon_\theta}$ using the input and the conditional information at that time frame (Line 9). The channel gains provided as conditional information ensure that the model becomes familiar with the actions taken based on the channel states. The loss function $\mathcal{L}_t$, which is Mean Squared Error (MSE), is used to calculate the distance between the actual noise and the predicted noise of the model, and the model is updated through backpropagation, and its performance improves until convergence (Lines 10-12).

After the training, the execution process starts with the trained model generating high-quality outputs from an isotropic Gaussian distribution, using only the channel gains at that time frame. The predicted noise of the model based on the conditional information is utilized to denoise the isotropic samples and generate the blocklength values (Lines 13-14). After executing the actions, the controller broadcasts the blocklength values to each sensor node so that the nodes can execute the command (Line 15).

3.5 Performance Evaluation

In this section, the performance of the diffusion-based resource allocation technique is compared to the benchmark models, including the pure optimization theory-based model and the DRL-based models. The DRL benchmarks, Branching Deep Q-networks (BDQ), which are suitable to solve problems with multi-variable action space, are utilized to solve the optimization problem in (3.1) with decision variables of blocklength, sampling period, and packet error probability. The D3QN is used for

Table 3.1: Simulation parameters

Parameter	Value	Parameter	Value
B	100 kHz	M_{th}	200 Symbols
L_i	100 bits	δ	0.99
Δ	1 ms	Ω	100 ms
$W_{tx,max}$	250 mW	W_c	5 mW
$\mathcal{N}$	64	σ^2	-174 dBm/Hz

the simplified optimization problem (3.4) with only blocklength decision variable. First, the simulation settings for the environment and the diffusion model are given in part 3.5.1, and then, performance comparison and analysis of the proposed and benchmark methods are presented in part 3.5.2.

3.5.1 Simulation Setup

Simulations are conducted for a network where the nodes are uniformly spread out within a circular area with a 50-meter radius, all communicating with a central controller. The path loss and shadowing effect result in large-scale fading, and it is modeled as $PL(d_i)[dB] = PL(d_0)[dB] + 10\alpha \log \frac{d_i}{d_0} + Z$, where d_i is the distance of node i from the central controller, $PL(d_i)$ is the path loss of node i at distance d_i in decibels, $PL(d_0) = 35.3\ dB$ is the path loss at the reference distance $d_0 = 1m$, path loss exponent $\alpha = 3.76$, Z is a Gaussian random variable with zero mean and standard deviation equal to 4 dB corresponding to log-normal shadowing. For small-scale fading, Jake's model [Ali et al., 2024] is used, expressed as a first-order complex Gauss-Markov process. The simulation parameters are listed in Table 3.1.

The neural network used for noise prediction has a modified UNet architecture. The grid search algorithm selects the optimal hyperparameters by systematically exploring all possible combinations of their values. The simulations are performed using the PyTorch library. Our algorithm trains the reverse process of the diffusion model with one input layer, nine convolutional layers, upsample and downsample layers, and four self-attention layers. The input $\mathbf{m}$ is a vector of optimal blocklengths, and the output is the denoised version of the same vector that is refined as the network parameters are updated. As for the conditional information, noising time step

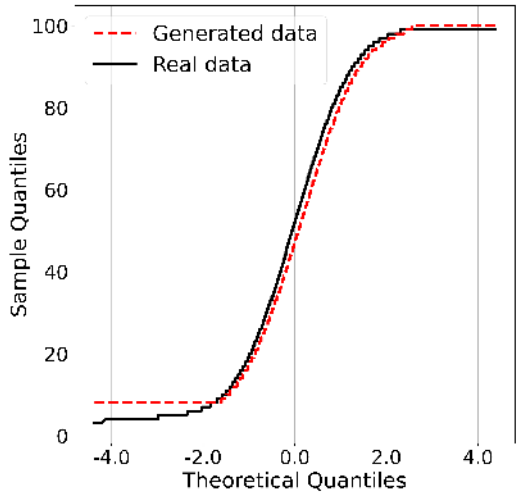
t is encoded using a positional encoding algorithm to convert discrete information and values into embedding vectors. The batch size is 32 in the training phase, and the training is done for 100 epochs. We use the AdamW optimizer [Loshchilov and Hutter, 2017] with an adaptive learning rate $\alpha^{(t)}$ for stable and robust training. The initial learning rate $\alpha^{(0)}$ is 10^{-5}. An Exponential Moving Average (EMA) method is applied to the network after a specific time in the training phase to ensure a stable weight update mechanism, where the weight factor for the EMA is set to 0.995. For testing, we perform 50 random initializations with 2500 episodes and average the results.

3.5.2 Performance Comparison and Analysis

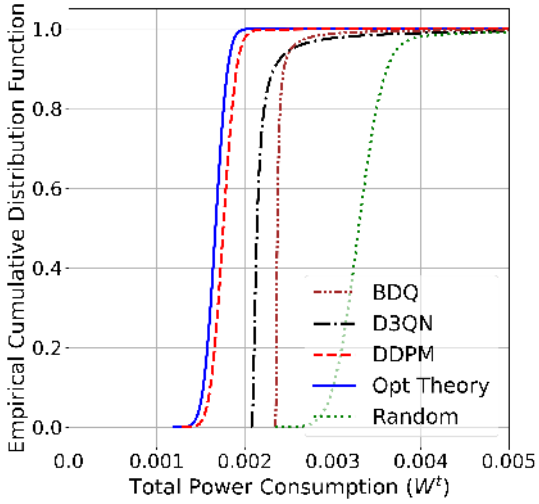
We validate the accountability of the generated sample distributions compared to the original ones. The q-q plots validate the similarity between the generated and true samples. Both samples are compared to a Gaussian distribution with zero mean and unit standard deviation. Figure 3.2a shows that samples exhibit similar behavior with very small discrepancy in the tail of the Gaussian distribution. This demonstrates that DDPM-based and optimization theory-based solutions are drawn from two PDFs with similar parameters.

Figures 3.3a and 3.2b show the average power consumption as a function of the number of nodes and the Empirical Cumulative Distribution Function (ECDF) of power consumption for 64 nodes, respectively.

The optimization-based method demonstrates the highest performance. The DDPM-based resource allocation algorithm falls slightly short of the optimization method but is very close in performance and similarity to the optimization algorithm. As the number of nodes increases, the DDPM-based model's performance becomes similar to that of the pure optimization method. This demonstrates the scalability of diffusion models, indicating that increasing the number of nodes in the network does not result in poor performance due to the complicated structure of the network. The DRL approaches cannot surpass the DDPM-based algorithm both in average power consumption and ECDF. Finally, random selection has the poorest performance



(a)

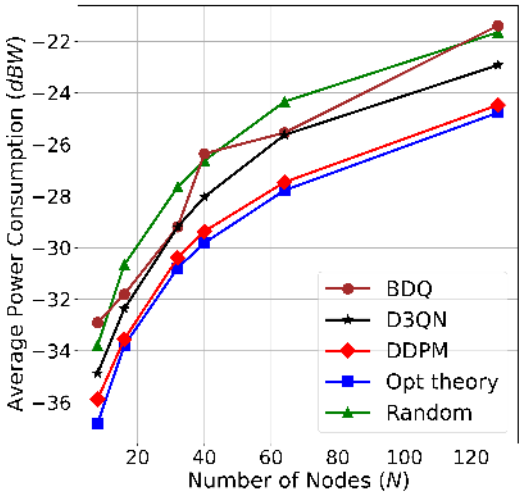


(b)

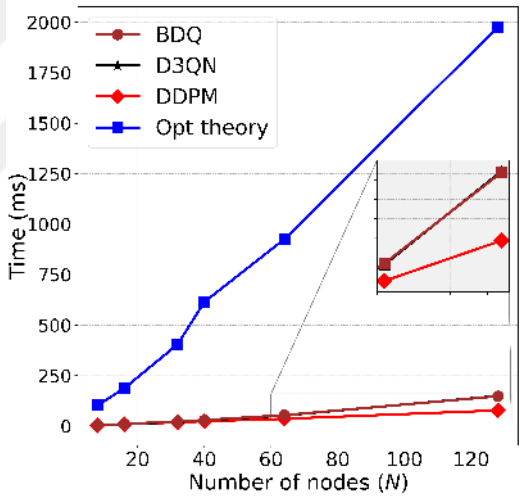
Figure 3.2: a) Q-Q plot for the true and generated samples. b) Testing results for different algorithms.

among the algorithms.

Figure 3.3b shows the execution time comparison of different algorithms as a function of the number of nodes. As expected, optimization theory-based method execution time grows exponentially as the number of nodes increases, which is not applicable in practical scenarios, especially URRLC, due to stringent conditions in



(a)



(b)

Figure 3.3: a) Average power consumption in the testing phase. b) Average execution time for different algorithms.

the environment. On the other hand, the DDPM-based method and DRL methods exhibit linear growth as the number of nodes increases. Moreover, although the training time of the DDPM-based method is higher than the DRL-based methods due to the number of parameters in the network, the DDPM-based model shows superior performance during the testing phase. This is because the structure of the

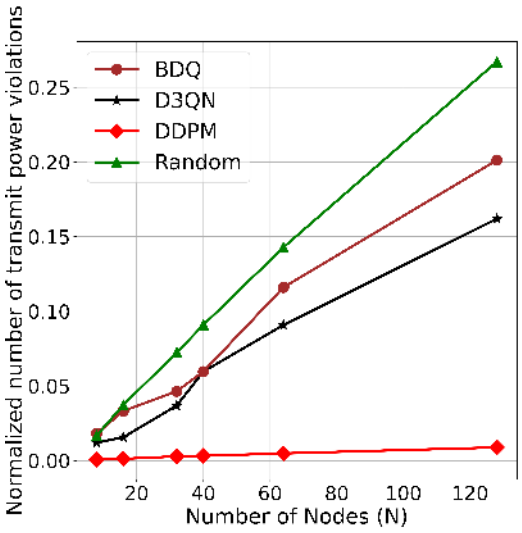


Figure 3.4: Average number of violations for different algorithms as a function of the number of nodes.

DDPM-based model allows for more effective utilization of GPU power, enabling faster performance when no further training is needed.

Finally, the proposed methodology demonstrates substantial resilience in terms of violating critical constraints because it is trained to mimic the optimization theory results, but the DRL-based approaches try to avoid constraint violation by incorporating penalties in their reward function, which is not always reliable. Figure 3.4 depicts the comparison between different algorithms for the number of times that the maximum transmit power constraint is violated as a function of the number of nodes. The result is the average number of times that algorithms have violated the power constraint in the testing phase, normalized over the total number of time steps. The proposed DDPM-based model demonstrates up to eighteen-fold improvement over cutting-edge DRL methods in terms of reliability for not violating the constraints. D3QN performs better than BDQ since it chooses actions from a smaller action space range, improving its performance and reliability. Random selection has the worst performance in terms of constraint violations.

3.6 Conclusion

In this chapter, we propose a novel diffusion-based resource allocation framework for joint optimization of communication and control systems, aiming to minimize total power consumption in URLLC with finite blocklength. Using a DDPM model and data from an optimization-based solution, our algorithm learns environmental variables and generates optimal blocklength values for each node. This approach outperforms existing DRL-based models regarding power consumption and avoiding constraint violations. Future work will explore a DDPM-based online learning algorithm to combine generative AI with DRL approaches in scenarios where datasets are unavailable or costly, such as massive MIMO communication systems.

Chapter 4

CONCLUSION

In this thesis, we propose a novel deep learning-based resource allocation framework for the joint optimization of communication and control systems in the finite blocklength regime, considering abstractions of the control and the communication systems to improve the model's performance compared to the previously proposed model-based approaches. The proposed algorithms outperform the conventional methods and demonstrate superior performance, indicating their efficacy.

First, a DRL-based approach is proposed to allocate blocklength values in URLLC domain. The formulated optimization problem with three decision variables of sampling period, blocklength, and packet error probability is simplified to consider blocklength only utilizing the optimality conditions in optimization theory. Then, a DRL algorithm, D3QN, is proposed to solve the problem in an online manner. The algorithm is trained through trial and error, and the trained model is used to take optimal action. The proposed methodology shows near-optimal performance compared to its benchmark optimization theory-based solution, outperforming existing algorithms in run-time comparison.

Finally, DDPMs are proposed to solve the resource allocation problem in WNCSs. The optimization theory-based solution collects a dataset of CSI and their corresponding optimal blocklength values. The dataset is the input state of the DDPM-based solution to learn the distribution of the optimal solutions and generate optimal values in the inference time. During the testing, the DDPM-based solution generates optimal blocklength values based on CSI as conditional information. The model outperforms previously proposed DRL-based models regarding total power consumption and improves the model performance in avoiding critical constraint violation.

In the future, we plan to extend the methodology to combine the power of

diffusion models and DRL methods for the scenario in which a dataset is unavailable or costly to collect, such as in massive MIMO communication systems, to train an online algorithm to capture the complex distribution of the environment and generate optimal actions in rapidly changing wireless environment. Furthermore, transfer learning and meta-learning will be used to adapt the model for various environments to enhance the generalization performance of the proposed models.

BIBLIOGRAPHY

[ANS,] Ansi/isa-100.11a-2011 wireless systems for industrial automation: Process control and related applications. `https://www.isa.org/products/ansi-isa-100-11a-2011-wireless-systems-for-industr`. Accessed: 2023-01-21.

[WHa,] Driving digital transformation in process automation. `https://www.fieldcommgroup.org/`. Accessed: 2023-01-01.

[IIO,] Industry iot consortium. https://www.iiconsortium.org/about-us/.

[Win,] Technical characteristics and spectrum requirements of wireless avionics intra-communications systems to support their safe operation. `https://www.itu.int/pub/R-REP-M.2283`. Accessed: 2023-01-02.

[Access, 2010] Access, E. U. T. R. (2010). Further advancements for e-utra physical layer aspects (release 9). *European Telecommunications Standards Institute*.

[Al-Dabbagh and Chen, 2016] Al-Dabbagh, A. W. and Chen, T. (2016). Design considerations for wireless networked control systems. *IEEE Transactions on Industrial Electronics*, 63(9):5547–5557.

[Ali et al., 2024] Ali, H. Q., Babazadeh Darabi, A., and Coleri, S. (2024). Optimization theory based deep reinforcement learning for resource allocation in ultra-reliable wireless networked control systems. *IEEE Trans. Commun.*, pages 1–1.

[ANSI, 2011] ANSI, I. (2011). Isa-100.11 a-2011 wireless systems for industrial automation: Process control and related applications. *ISA: San Diego, CA, USA*, pages 1–792.

[Arampatzis et al., 2005] Arampatzis, T., Lygeros, J., and Manesis, S. (2005). A survey of applications of wireless sensors and wireless sensor networks. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pages 719–724. IEEE.

[Baumann et al., 2018] Baumann, D., Zhu, J.-J., Martius, G., and Trimpe, S. (2018). Deep reinforcement learning for event-triggered control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 943–950. IEEE.

[Bello and Zeadally, 2014] Bello, O. and Zeadally, S. (2014). Intelligent device-to-device communication in the internet of things. *IEEE Systems Journal*, 10(3):1172–1182.

[Boyd and Vandenberghe, 2004a] Boyd, S. and Vandenberghe, L. (2004a). *Convex Optimization*. Cambridge University Press.

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

[Cao et al., 2023] Cao, J., Zhu, X., et al. (2023). Age of loop for wireless networked control system in the finite blocklength regime: Average, variance and outage probability. *IEEE Trans. Wireless Commun.*, 22(8):5306–5320.

[Castro et al., 2022] Castro, H., Pinto, N., Pereira, F., Ferreira, L., Ávila, P., Bastos, J., Putnik, G. D., and Cruz-Cunha, M. (2022). Cyber-physical systems using open design: an approach towards an open science lab for manufacturing. *Procedia Computer Science*, 196:381–388.

[Cui et al., 2005] Cui, S., Goldsmith, A. J., and Bahai, A. (2005). Energy-constrained modulation optimization. *IEEE transactions on wireless communications*, 4(5):2349–2360.

[de Sant Ana et al., 2021] de Sant Ana, P. M., Marchenko, N., Popovski, P., and Soret, B. (2021). Age of loop for wireless networked control systems optimization.

In *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–7. IEEE.

[Dertouzos, 1974] Dertouzos, M. (1974). Control robotics: The procedural control of physical processes. *IFIP Congress 1974*.

[Dobslaw et al., 2014] Dobslaw, F., Zhang, T., and Gidlund, M. (2014). End-to-end reliability-aware scheduling for wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 12(2):758–767.

[Durisi et al., 2016] Durisi, G., Koch, T., and Popovski, P. (2016). Toward massive, ultrareliable, and low-latency wireless communication with short packets. *Proceedings of the IEEE*, 104(9):1711–1726.

[Eisenbrand and Rothvoß, 2010] Eisenbrand, F. and Rothvoß, T. (2010). Edf-schedulability of synchronous periodic task systems is conp-hard. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1029–1034. SIAM.

[Farjam and Charalambous, 2021] Farjam, T. and Charalambous, T. (2021). Effect of computational power of sensors on event-triggered control mechanisms over a shared contention-based network. *arXiv preprint arXiv:2104.03076*.

[Hespanha et al., 2007] Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162.

[Hessel et al., 2018] Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Hingmire, 2015] Hingmire, V. S. (18 December 2015). Zigbee crosses the chasm:

A market dynamics report on ieee 802.15.4 and zigbee, on world, 2010. PhD Dissertation.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

[Kagermann et al., 2013] Kagermann, H., Wahlster, W., , and Helbig, J. (2013). Recommendations for implementing the strategic initiative industrie 4.0.

[Kakkavas et al., 2024] Kakkavas, G., Diamanti, M., et al. (2024). 5G perspective of connected autonomous vehicles: Current landscape and challenges toward 6G. *IEEE Trans. Wireless Commun.*, pages 1–8.

[Karagiannis et al., 2011] Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K., and Weil, T. (2011). Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE communications surveys & tutorials*, 13(4):584–616.

[Kasgari et al., 2020] Kasgari, A. T. Z. et al. (2020). Experienced deep reinforcement learning with generative adversarial networks (GANs) for model-free ultra reliable low latency communication. *IEEE Trans. Commun.*, 69(2):884–899.

[Kerboeuf et al., 2024] Kerboeuf, S., Porambage, P., et al. (2024). Design methodology for 6G end-to-end system: Hexa-X-II perspective. *IEEE Open J. Commun. Soc.*, 5:3368–3394.

[Letafati et al., 2023] Letafati, M., Ali, S., and Latva-aho, M. (2023). Generative AI-based probabilistic constellation shaping with diffusion models. *arXiv preprint arXiv:2311.09349*.

[Liang et al., 2017] Liang, L., Kim, J., Jha, S. C., Sivanesan, K., and Li, G. Y. (2017). Spectrum and power allocation for vehicular communications with delayed csi feedback. *IEEE Wireless Communications Letters*, 6(4):458–461.

[Lima et al., 2022] Lima, V., Eisen, M., Gatsis, K., and Ribeiro, A. (2022). Model-free design of control systems over wireless fading channels. *Signal Processing*, 197:108540.

[Loshchilov and Hutter, 2017] Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[Montalban et al., 2020] Montalban, J., Iradier, E., Angueira, P., Seijo, O., and Val, I. (2020). Noma-based 802.11 n for industrial automation. *IEEE Access*, 8:168546–168557.

[of Automation, 2009] of Automation, I. S. (2009). *Wireless Systems for Industrial Automation: Process Control and Related Applications : ISA-100.11a-2009*. ISA.

[Park et al., 2018a] Park, P., Coleri, S., et al. (2018a). Wireless network design for control systems: A survey. *IEEE Communications Surveys & Tutorials*, 20(2):978–1013.

[Park et al., 2018b] Park, P., Coleri Ergen, S., Fischione, C., Lu, C., and Johansson, K. H. (2018b). Wireless network design for control systems: A survey. *IEEE Communications Surveys & Tutorials*, 20(2):978–1013.

[Park et al., 2014] Park, P., Khadilkar, H., Balakrishnan, H., and Tomlin, C. J. (2014). High confidence networked control for next generation air transportation systems. *IEEE Transactions on Automatic Control*, 59(12):3357–3372.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

[Pereira et al., 2007] Pereira, N., Andersson, B., and Tovar, E. (2007). Widom: A dominance protocol for wireless medium access. *IEEE Transactions on Industrial Informatics*, 3(2):120–130.

[Polyanskiy et al., 2010] Polyanskiy, Y., Poor, H. V., and Verdú, S. (2010). Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359.

[Promwongsa et al., 2020] Promwongsa, N., Ebrahimzadeh, A., Naboulsi, D., Kianpisheh, S., Belqasmi, F., Glitho, R., Crespi, N., and Alfandi, O. (2020). A comprehensive survey of the tactile internet: State-of-the-art and research directions. *IEEE Communications Surveys & Tutorials*, 23(1):472–523.

[Ren et al., 2019] Ren, H., Pan, C., Deng, Y., Elkashlan, M., and Nallanathan, A. (2019). Joint pilot and payload power allocation for massive-mimo-enabled urllc iiot networks. *IEEE Journal on Selected Areas in Communications*, 38:816–830.

[Ren et al., 2020] Ren, H., Pan, C., Deng, Y., Elkashlan, M., and Nallanathan, A. (2020). Joint pilot and payload power allocation for massive-mimo-enabled urllc iiot networks. *IEEE Journal on Selected Areas in Communications*, 38(5):816–830.

[Ren et al., 2022] Ren, H., Wang, K., and Pan, C. (2022). Intelligent reflecting surface-aided urllc in a factory automation scenario. *IEEE Transactions on Communications*, 70(1):707–723.

[Sadi and Coleri Ergen, 2017] Sadi, Y. and Coleri Ergen, S. (2017). Joint optimization of wireless network energy consumption and control system performance in wireless networked control systems. *IEEE Transactions on Wireless Communications*, 16(4):2235–2248.

[Sadi et al., 2014] Sadi, Y., Coleri Ergen, S., and Park, P. (2014). Minimum energy data transmission for wireless networked control systems. *IEEE Transactions on Wireless Communications*, 13(4):2163–2175.

[Schaul et al., 2015] Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

[She et al., 2018] She, C., Yang, C., and Quek, T. Q. S. (2018). Joint uplink and downlink resource configuration for ultra-reliable and low-latency communications. *IEEE Transactions on Communications*, 66(5):2266–2280.

[Subchan et al., 2021] Subchan, S., Zuhair, Z., Asfihani, T., Adzkiya, D., and Kim, S. (2021). Energy optimization on wireless-networked control systems (w-ncss) using linear quadratic gaussian (lqg). *International Journal of Control, Automation and Systems*, 19(12):3853–3861.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[Tavakoli et al., 2018] Tavakoli, A., Pardo, F., and Kormushev, P. (2018). Action branching architectures for deep reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

[Wu et al., 2010] Wu, Y., Buttazzo, G., Bini, E., and Cervin, A. (2010). Parameter selection for real-time controllers in resource-constrained systems. *IEEE Transactions on Industrial Informatics*, 6(4):610–620.

[Yilmaz et al., 2015] Yilmaz, O. N., Wang, Y.-P. E., Johansson, N. A., Brahmi, N., Ashraf, S. A., and Sachs, J. (2015). Analysis of ultra-reliable and low-latency 5g communication for a factory automation use case. In *2015 IEEE international conference on communication workshop (ICCW)*, pages 1190–1195. IEEE.

[Zhang and Burns, 2009] Zhang, F. and Burns, A. (2009). Schedulability analysis for real-time systems with edf scheduling. *IEEE Transactions Comput*, 58(9):250–1258.

[Zhao et al., 2024] Zhao, Z., Liu, W., et al. (2024). Deep learning for wireless-networked systems: A joint estimation-control–scheduling approach. *IEEE Internet of Things Journal*, 11(3):4535–4550.

[Zhao et al., 2023] Zhao, Z., Liu, W., Quevedo, D. E., Li, Y., and Vucetic, B. (2023). Deep learning for wireless networked systems: A joint estimation-control-scheduling approach. *IEEE Internet of Things Journal*, pages 1–1.