Doctoral Dissertation

Model-tuning Approach to Quantized System Design

Yugo Ogio

January 2025

Graduate School of Engineering,
Osaka University

Abstract

The ultimate goal in the field of control engineering is to be able to manipulate the behavior of various systems freely. There are many methods for manipulating the behavior of a plant and controlling a system. One of the most sophisticated methods is the control system design based on the mathematical model of the plant. It is possible to achieve control performance by understanding the dynamics of a target plant. An accurate understanding of dynamics generally requires detailed models that reproduce various plant phenomena. However, there are cases where a detailed model is not required in control system design. For example, in the case of control around an equilibrium point for a nonlinear plant, a linear approximated model around the equilibrium point is sufficient, and a detailed nonlinear model is unnecessary. In another case, detailed models cannot be used when the cost of controller design or computation time is constrained. In addition, even if a detailed model can be constructed, it may be difficult to use the detailed model in control system design. For example, Neural Network (NN) models are expected to be detailed models, but there is no design theory for NN models in conventional linear control theory.

As described above, a detailed model is not always necessary in control system design; it is necessary to change models according to the design intent of the control system, such as control objectives and design specifications. To reflect the design intent in the models, I attribute the control system design problem to the model search problem, reflecting the design intent in the models. Motivated by the above background, this thesis focuses on the model-tuning approach for control system design as an optimization problem of models. In this thesis, I address several specific problems: problem formulation, application to the design of a dynamic quantizer in a control system, and application to the design of a quantization process in an image and graph signal processing system.

In Chapter 2, I formulate the problem of a model-tuning approach for control system design. The proposed method first fixes the controller design procedure. Fixing the design procedure means characterizing the controller by a particular model. In this thesis, the model characterizing the controller is called the ”Design model,” and I search for the design model to satisfy the design specifications. In other words, the controller design problem is converted to the design model search problem. The design procedure of the controller mathematically corresponds to the mapping from models to controllers. Since the mapping is given in the proposed method, the search space is the model space, not the controller space, and the controller is automatically designed once the model is constructed.

In Chapter 3 and Chapter 4, I formulate the design problem of a dynamic quantizer based on my proposed method. Specifically, I deal with a sampled-data dynamic quantizer for a linear continuous-time system in Chapter 3 and a nonlinear continuous-time system in Chapter 4. The validity of my proposed method is demonstrated through the numerical examples and the comparison with the method that directly searches for the parameters of the dynamic quantizer.

In Chapter 5 and Chapter 6, I extend the problem settings to the design of a system using Topological Data Analysis (TDA). In Chapter 5, I design a binarization process of images in a system that segments gray-scale images using TDA of binary images. In Chapter 6, I design a quantization process of weights of NNs in a system, which evaluates the performance of a Quantized Neural Network (QNN). In both chapters, I demonstrate the effectiveness of the proposed method through numerical experiments.

Acknowledgement

本論文は，著者が大阪大学大学院機械工学専攻博士後期課程に在籍した間に従事した研究を集成したものです．

本論文の執筆にあたり，石川将人教授，南裕樹准教授，増田容一助教，鈴木朱羅助教には数え切れないほどのご助力をいただきました．特に，脱線しやすい私の興味に対し常に冷静な視点でアドバイスをくださった南准教授には，本研究の方向性を見失わずに済み，無事に修了できたことを感謝致します．さらに，博士後期課程の年の近い先輩として，お金のことや研究との向き合い方について親身なアドバイスを頂いた増田助教，鈴木助教に厚く御礼を申し上げます．

忙しい合間を縫って本論文の副査を快く引き受けて下さり，そして懇切丁寧なご指導を賜った大須賀公一教授，佐藤訓志教授，石川将人教授に厚く御礼を申し上げます．大須賀公一教授には，メールでのやり取りとなってしまいましたが，お忙しい中で博士論文のアイデアに対する貴重なご意見を頂きました．佐藤訓志教授には，問題設定から具体例まで，多大なご質問とご指摘を頂き，研究の深掘りを促して頂きました．また，石川教授には，時には冗談も交えながら，研究会や論文執筆などの折りに助言を頂き，研究の進行を支えて頂きました．

本研究は，周りの方の温かい支援と協力が無ければ，成り立ちませんでした．研究室の先輩方には，研究についての様々なアドバイスとプラモデルとぬいぐるみを頂きました．研究室の同期には，共に良く遊び，研究においては大きな刺激を受けました．研究室の後輩には，頼りない先輩の私に話しかけて下さり，おしゃべりや麻雀という私が研究室に行く原動力を与えてくれました．大学時代の友人たちには，研究室外のコミュニティとして，飲み会や旅行などでリフレッシュさせてくれました．高校時代の友人たちは，一緒に飲んでくれるのに加えて，私が落ち込んでいる時に励ましてくれました．また，家族は，博士課程の最後の日々を支えてくれました．石川・南研究室の先生方，先輩方，同期，後輩，友人，家族に感謝の意を表します．

最後に，本研究に対してご支援を頂いた JST 次世代挑戦的研究者育成プロジェクト様，JSPS 日本学術振興会様，公益財団法人 立石科学技術振興財団様に感謝を申し上げます．

Contents

List of Figures

Notation and definitions

The following notation is used in this thesis.

$\mathbb{R}$	Set of real number
$\mathbb{R}_+$	Set of positive real number
$\mathbb{N}$	Set of natural number
$\in$	Belong to
$\subset$	Subset of (strict or not)
$\cup$	Union
$\cap$	Intersection
$\emptyset$	Empty set
$\setminus$	Set difference
$M^\top$	Transpose of the matrix M
M^{-1}	Inverse of the matrix M
$M^\dagger$	Moore-Penrose pseudo-inverse of the matrix M
d	Quantization width
T_s	Sampling period
N_bit	Number of quantization bits
$\|x\|$	Euclidean norm of the vector x
$L_2[0,T]$	Set of the square-integrable functions on the interval $[0,T]$
$\mathrm{sgn}(\cdot)$	Signum function
$\lfloor\cdot\rfloor$	Floor function

The following notation is also defined.

- For the quantization width d and the number of quantization bits N_bit, the set of quantization steps is denoted as $\mathbb{V} := \{\pm d,\ \pm 2d,\ \dots,\ \pm N_\mathrm{bit} d\}$.

Chapter 1

Introduction

1.1 Background

The ultimate goal in the field of control engineering is to be able to manipulate the behavior of various systems freely. There are many methods for manipulating the behavior of a plant and controlling a system. One of the most sophisticated methods is the control system design based on the mathematical model of the plant. It is possible to achieve control performance by understanding the dynamics of a target plant.

An accurate understanding of dynamics generally requires detailed models that reproduce various plant phenomena. However, there are cases where a detailed model is not required in control system design. First, a detailed model may not be necessary because the control objectives do not require it. For example, in stabilization control near an equilibrium point for a nonlinear system, it is common to use a linear approximation model near the equilibrium point. Also, a detailed model is not necessary in some cases due to constraints on the controller, such as discrete-valued inputs and outputs or a fixed order of the controller. Furthermore, even if a detailed model can be constructed, it may not be helpful for control. For example, Neural Network (NN) models have high description capability, but it is difficult to obtain the frequency characteristics and inverse models necessary for control from the model.

As described above, a detailed model is not always necessary in control system design; it is necessary to change models according to the design intent of the control system, such as control objectives and design specifications. To reflect the design intent in the models, I attribute the control system design problem to the model search problem. Motivated by the above background, this thesis focuses on the model-tuning approach for control system design as an optimization problem of models.

My proposed method first fixes the controller design procedure, converting the controller design problem to a model search problem. Fixing the design procedure means characterizing the controller by a particular model. In this thesis, the model characterizing the controller is called the ”Design model,” and I search for the design model to satisfy the design specifications. In other words, the controller design problem is attributed to the design model search problem. The design procedure of the controller mathematically corresponds to the mapping from models to controllers. Since the mapping is given in the proposed method, the search space is the model space, not the controller space, and the controller is automatically designed once the model is constructed.

Here, I focus on the model matching method for the PID control system as one of the system design methods based on constructing a model that reflects the design intent[1]. In the model matching method, first, a transformation equation is derived to determine the controller's parameters from the parameters of the closed-loop system model. Next, the desired control properties are obtained as the reference model for the closed-loop system model. In other words, the model matching method converts the controller search problem to the model search problem by imposing a design procedure on the controller.

In addition to the model matching method, some studies have designed control systems using a model tuning method based on the design procedure of the controller. For example, Shikada et al. considered the problem of designing a robust state feedback controller with an observer for a system with polytope-type uncertainty[2, 3]. In this case, they showed that optimizing the model and its linear transformation performs better than the conventional nominal model. In addition to this previous study, Okajima also confirmed that the maximum likelihood model is not always suitable as a nominal model for model error compensator[4]. In another related study, Wada and Tsurushima proposed a method of adding an integrator to the model as a servo compensator to achieve tracking to the target signal in the design of a servo system by model predictive control[5, 6]. Minami and Kashima designed a quantizer for non-minimum phase systems[7, 8]. This study showed that output divergence can be prevented by designing a filter for a dynamic quantizer with a partial model obtained by serial system decomposition. The partial model ignores unstable zeros in the plant. Kusui et al. extended the study of Minami and Kashima and proposed a dynamic quantizer design method for the MIMO non-minimum phase system using serial system decomposition[9].

These studies do not design based on a model that mimics the characteristics of the plant but tunes the model by fixing the design procedure of the controller or quantizer, which is different from directly tuning from data. It is interesting in that it is positioned between model-based and data-driven methods.

1.2 Goal of my study

In this thesis, I redefine the above studies as a more generalized problem setting and propose a system design method based on model tuning. The target plant and the design procedure of the controller for the system are given, and the problem is regarded as searching for a design model that reflects the design intent. The design model does not necessarily reproduce the behavior of the plant but is necessary for control system design and is obtained through tuning.

The proposed method has the advantage of obtaining a model suitable for the design procedure without necessarily requiring the design model to reproduce the dynamics of the plant. In addition, the proposed method may potentially make the search space small by solving the model search problem. For example, in optimization, search algorithms such as genetic algorithms, particle swarm optimization, and simulated annealing are used, and by imposing some structure on the update rule of the design parameters in these algorithms, fast search can be performed. Similarly, by imposing a particular structure on the controller to reduce the search space to the model, the proposed method has the potential to design faster than direct search of the controller.

1.3 My approach and applications

In this thesis, I propose a system design method based on model tuning. I define the following sub-issues to clarify the basic properties and usefulness of the proposed method in this study.

The main problem to be solved is the design of a control system with desired control properties. In basic model-based design, to solve this problem, the design was to construct a model first and then design a controller. The model itself was considered to be known as the information necessary for system design, and what was important in system design was the controller design after model construction. The model was given, and the problem was regarded as finding the controller design method.

On the other hand, in this study, I propose a system design method based on model tuning. In the proposed method, the design procedure of the controller is given, and I find the design model. It should be noted that the parameters of the controller must change by tuning the model. Therefore, the procedure for designing the controller from the model, which is the mapping from the model to the controller, is actually given. In the proposed method, since the mapping from the model to the controller is given, the search space is the model space, not the controller space. Once the model is constructed, the controller is automatically designed.

Next, I describe how to fix the controller design procedure. One way is to use analytical results of the fundamental control system design problem. For example, in control theory, there are cases where analytically optimal controllers such as pole assignment and LQR control are derived. The analytically derived equations are fixed as the design procedure, and the model given to the equations is tuned. In other words, it is an approach to searching for the model that uses the controller design procedure as the designer's knowledge. In addition to the proposed method of model search, there is a method to search for the parameters of the system directly. However, model search uses the structure of the controller as knowledge, so it may be possible to find a more valid solution more efficiently. Furthermore, unlike direct search, model search uses the structure of the system as knowledge and obtains the design model for a control system. It is considered to have more explainability in system design.

In Chapter 2, as an important sub-issue, I define the problem of system design using the model tuning approach. The formulated problem is a generalization of the problems considered in the following chapters, and by solving this problem, I can consider the properties and characteristics of the design model. However, this problem is very abstract, and the solution is difficult to find at this point. Therefore, in the following chapters, I will reduce each problem to a specific problem setting, propose a method to solve the problem and evaluate the solution.

In Chapter 3 and Chapter 4, I formulate the problem of system design using the model tuning approach and describe the design of a dynamic quantizer as a specific example. A quantizer is a device that converts a continuous signal into a discrete signal. A dynamic quantizer is a device that makes the output of the discrete-valued input system close to the output of the continuous-valued input system by feeding back the quantization errors. In the design of conventional dynamic quantizers, discrete-time and continuous-time quantizers have been proposed for discrete-time and continuous-time systems, respectively. In Chapters 3 and Chapter 4, I propose the problem of designing a discrete-time quantizer for a continuous-time system and demonstrate the effectiveness of system design using model search. The controller design procedure to be fixed here is the design formula for the discrete-time optimal dynamic quantizer for the discrete-time plant. In the design of dynamic quantizers, I fix the procedure for the optimal dynamic quantizer for the discrete-time plant and search for models that fill the gap between continuous and discrete time.

In Chapter 5 and Chapter 6, I extend the problem setting of the proposed method and consider the application to a system using Topological Data Analysis (TDA). In Chapter 5, I use a system that segments gray-scale images using TDA of binary images. The signal of this system is an image signal, the design procedure of the binarization process is fixed, and I search for the binarization algorithm that works well for segmentation with TDA. In Chapter 6, I use a system that evaluates the quantization performance of Neural Networks (NNs). The signal of this system is a graph signal of NN, the design procedure of a quantization process of NN weights is fixed, and I search for the error diffusion filter that keeps the performance of QNN high. The proposed method can be applied to various system design problems, not just control system design.

1.4 Contributions

In this thesis, the contributions of this study are as follows.

- I proposed a method to solve the system design problem using model tuning.
- I defined the problem of system design using model search and formulated it as an optimization problem
- I solved various examples by the proposed method and demonstrated the effectiveness of the proposed method through numerical experiments.

This study contributes by proposing a model-tuning solution for a general problem, though related work exists[1, 2, 3, 4, 5, 6, 7, 8, 9]. Based on these related works, I define a more general problem and consider searching for a model for the problem. Furthermore, I apply the proposed method to various specific problem settings and demonstrate its effectiveness.

Chapter 2

Tuning of design models for control system design

2.1 Previous studies of model-tuning method for control system

As described in the previous chapter, my proposed design method differs from the conventional model-based system design method. I show the positioning of the proposed method in the design method of the control system in Fig. 2.1.

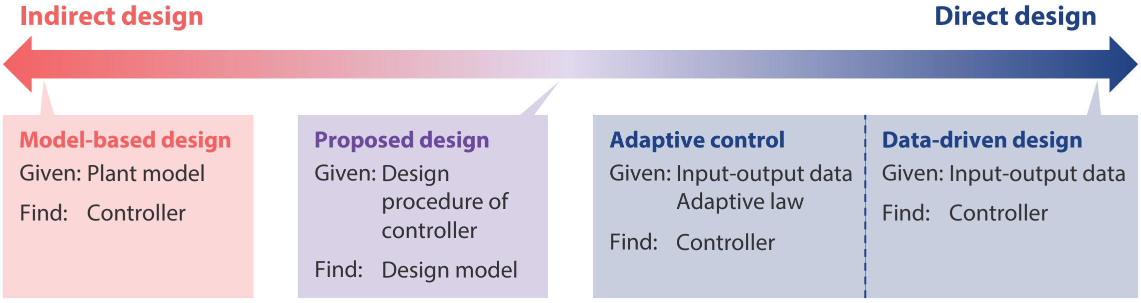


Fig. 2.1: Positioning of the proposed method in the design methods of control system.

Here, I introduce various design methods for conventional model-based design. First, in modern control theory, state feedback control and optimal control are major model-based methods[10, 11, 12]. In these control system designs, the system is modeled as a State Space Model (SSM), and the controller is designed using the SSM. In robust control, a single model is determined for the system (nominal model) in the presence of uncertainty, and a controller that conservatively acts on the uncertainty is designed for the model[13, 14]. In system identification, the system is identified and expressed as a model, and the identified model is used for control[15, 16, 17]. In disturbance observer, the value of the disturbance is estimated using the inverse model of the system, and the estimated disturbance and the true disturbance are canceled out to achieve the desired control performance[18, 19, 20]. These methods assume that the model of the plant is known in advance or can be obtained. Moreover, the model must sufficiently reflect the characteristics of the plant.

The simultaneous design of structural system and control systems is a method similar to the model-based system design method[21, 22, 23, 24]. In this method, the controller and some parameters of the plant model are designed simultaneously with the structure of the model fixed. Thus, simultaneous design assumes that some plant parameters are tunable but are based on the structure of the plant model. In that sense, this method is similar to model-based design.

The quality of the model greatly influences the performance of the control system in model-based methods. If the model accurately reflects the characteristics of the plant and can be used for controller design, the desired control performance can be achieved. On the other hand, when the system is complex, or there are constraints on the control system design, it is difficult to construct the model and design the control system.

In contrast, data-driven methods that design systems directly from data without relying on models are also known. For example, classical PID controller design methods are known, such as the Ziegler-Nichols method[25, 26, 27]. The method tunes the parameters of the PID controller through trial and error, using empirical rules and data such as step responses. One-shot data-driven methods such as Fictitious Reference Iterative Tuning (FRIT) are also known[28, 29, 30, 31]. FRIT is a method that determines the parameters of the controller to approach the response of the control system to the response of the ideal system using the obtained input-output data. Furthermore, some studies design learning-based controllers[32, 33, 34, 35]. These methods do not require a model, so it is possible to design a controller based only on data without the need to know the model of the plant in advance. However, it is necessary to obtain sufficient data to design the controller from experiments or simulations.

Adaptive control is another known method[36, 37, 38]. Adaptive control is a method of designing a controller that adjusts its parameters online to respond to the uncertainties of the plant. Adaptive control requires information on the relative degree of the plant to construct a stable adaptive control system. However, adaptive control needs data about the control system, adjusts the parameters of the controller online, and is similar to data-driven methods.

Data-driven methods do not explicitly use models, and compared to model-based methods, it is not necessary to know the plant model in advance. However, since data-driven methods design controllers based only on data, it is difficult to design a controller when sufficient data is unavailable. Moreover, data-driven methods may be limited in the class of systems to which they can be applied.

The proposed method aims to design control systems through model tuning. Compared to model-based methods, the proposed method fixes the controller design procedure and searches for the model. In the actual construction of the model, I search for the model so that the model satisfies the performance using the input-output and state data of the system. Moreover, the model may not sufficiently reflect the characteristics of the plant. The proposed method does not incorporate detailed information about the system into the model compared to model-based methods.

Unlike data-driven methods that adjust the parameters of the controller directly, the system is constructed through the model. The proposed method fixes the design procedure and searches the model. Therefore, the proposed method is positioned between conventional model-based methods and data-driven methods.

As a method for designing systems focusing on the model, Ikezaki et al. proposed Virtual Internal Model Tuning (VIMT)[39, 40, 41]. VIMT is a data-driven controller design method for closed-loop systems. First, the ideal response of the closed-loop system is fixed, and the plant model is virtually obtained using the ideal response and the controller that realizes the ideal response. Next, the target response is represented using the virtual plant model and the initial response. Finally, the virtual model is replaced by a parameterized controller, and an optimization is performed to close the predicted response of the output data represented by the controller and the target response. VIMT fixes the ideal response, and a controller that reflects the designer's intention can be designed. However, in VIMT, the virtual model of the plant is replaced by a parameterized controller at the end, and the parameters of the controller are directly tuned. In that sense, it differs from the proposed method, which tunes the model.

In addition, the study by Fujimoto et al. designs a dynamic quantizer from input-output data by fixing the structure of the quantizer and searching for parameters[42]. A quantizer is a device that converts a continuous signal into a discrete signal, and a dynamic quantizer is a quantizer that minimizes output error by feeding back the input quantization error. As a previous study, it is known that the inverse model of the plant is required to design the optimal dynamic quantizer[43]. Using this knowledge, Fujimoto et al. proposed a data-driven method for designing a dynamic quantizer by setting the evaluation function so that the input-output data satisfies the relationship of the inverse model of the system.

These two studies use only the information of the model when learning the parameters and do not tune the parameters of the model. However, it is shown that the parameters can be efficiently learned by fixing the structure of the system from the desired input-output characteristics and knowledge of previous studies.

Similar to the method I propose, there are previous studies that directly search for models. As one of the model-tuning methods, Kitamori proposed a model-matching method for designing a PID controller[1]. Also, there is a study that satisfies the control performance by searching for a mathematical model for uncertainty[2, 3]. For example, Shikada et al. proposed a method of searching for a model while fixing the structure of a controller with an observer for uncertainty. Specifically, they considered the problem of designing a robust controller that estimates the states using a linear observer for a system with polytope-type uncertainty. In the conventional method, the center of uncertainty is regarded as the nominal model, and a controller is designed for the nominal model. However, Shikada et al. introduced the degree of freedom of linear transformation of the model and searched for the model with the highest control performance among the linearly transformed models. As a result, they searched for the model using numerical optimization and achieved higher control performance than the conventional method. In addition to the related work, Okajima also confirmed that the maximum likelihood model is not suitable as a nominal model for model error compensator[4]. In another related study, Wada and Tsurushima proposed a method of adding an integrator to the model as a servo compensator to achieve tracking to the target signal in the design of a servo system by model predictive control[5, 6].

There is also another related work that designs a control system using a model that does not accurately reflect the characteristics of the non-minimum phase plant[7, 8]. It is known that a quantizer that minimizes the output error can be designed by using the inverse model of the control system. However, since the non-minimum phase system has unstable zeros, the quantizer becomes unstable when the inverse model is used, and the output diverges. Therefore, for the non-minimum phase system, it has been proposed that a high-performance quantizer be designed using a partial model obtained by serial system decomposition, which ignores the unstable zeros. Kusui et al. extended the study of Minami et al. and proposed a method using serial system decomposition of MIMO non-minimum phase systems[9].

The above studies show the possibility of a model-tuning design method for the various control system design problems. Based on these previous studies, I define a more general problem of using a model-tuning method for control system design.

2.2 General problem formulation for tuning of design models

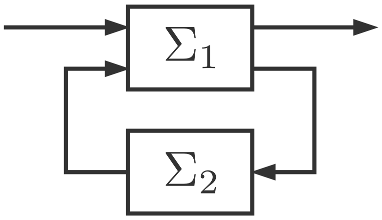


Fig. 2.2: General block diagrams for the problem formulation of the proposed method.

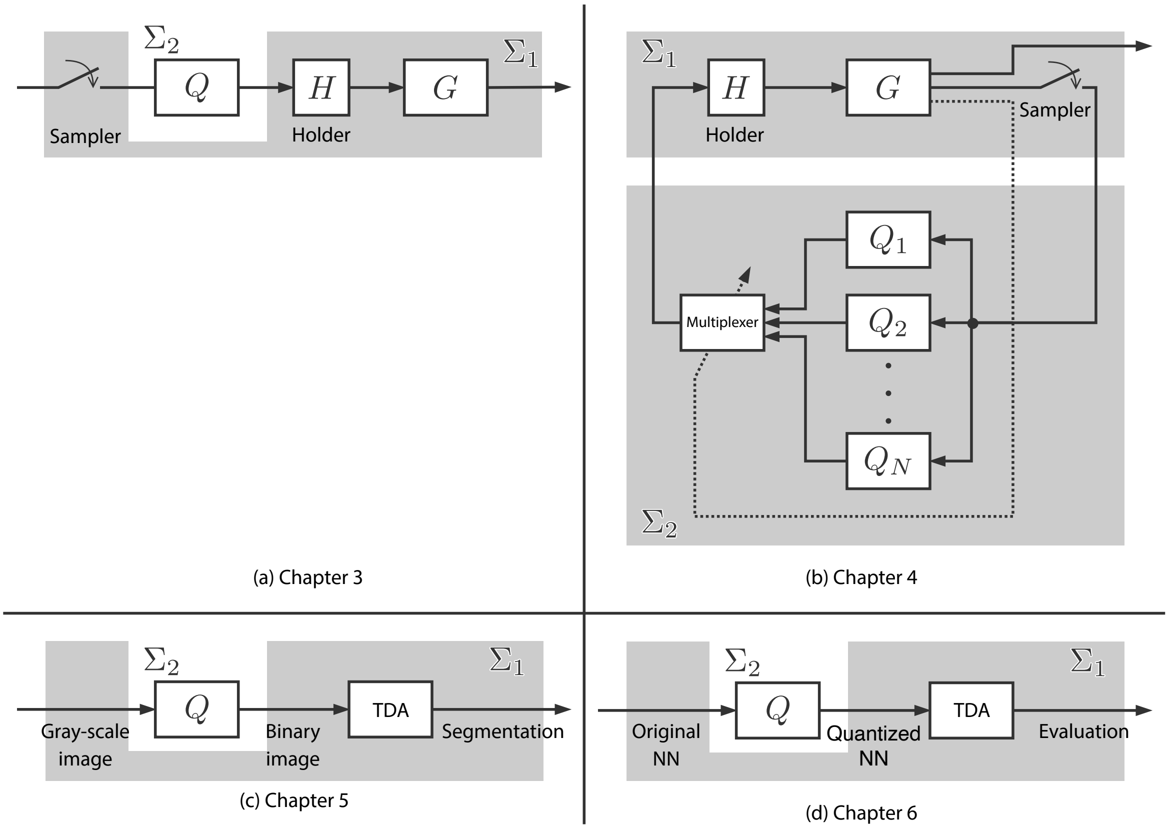


Fig. 2.3: Each block diagrams of following chapters.

Based on the above positioning, I describe the general problem formulation of this study. Here, I consider the system Σ_1, Σ_2, as shown in the Fig. 2.2. The system Σ_1 is the system to be controlled, and the system Σ_2 is the system to be designed, such as a controller and a quantizer. The other parameters are defined as follows: the model to be tuned is P, the mapping from the model to the system is $\mathcal{M} : P \mapsto \Sigma_2$, the hyperparameters such as data are θ, and the evaluation function is J. I fix the system design procedure $\mathcal{M}$, search for the model P according to a specific evaluation index J, and construct the system Σ_2 in the proposed method. In summary, I aim to obtain $P^\star$ and design the system Σ_2 by tuning the model P based on the evaluation function J as follows:

$$P^\star = \underset{P}{\operatorname{argmin}} \ J(\mathcal{M}(P); \Sigma_1, \theta). \tag{2.1}$$

In the following chapter, I will consider the specific problem formulation of the proposed method for the control system. The specific problem formulation is shown in Fig. 2.3. In Chapter 3 and Chapter 4, I apply my proposed method to design of a discrete-time quantizer for a continuous-time system. The subsystem Σ_1 comprises the continuous-time system G, holder H, and sampler; the subsystem Σ_2 is a quantizer Q. The mapping $\mathcal{M}$ is the design procedure for the dynamic quantizer, composed of zero-order hold and the equation of the optimal dynamic quantizer for a discrete-time system. I fix the design procedure $\mathcal{M}$ and find the continuous-time design model P. In Chapter 3, the continuous-time system G is linear, and quantizer Q is a dynamic quantizer with a linear filter. In Chapter 4, the continuous-time system G is nonlinear, and the quantizer Q is a switching-type dynamic quantizer, which is composed of a linear quantizer$Q_1, Q_2, \ldots, Q_N$. Each design model $P_1, P_2, \ldots, P_N$ is obtained for each sub-quantizer $Q_1, Q_2, \ldots, Q_N$.

In Chapter 5 and Chapter 6, I apply my proposed method to the system composed of TDA and quantization process. The subsystem Σ_1 comprises the TDA process, and the subsystem Σ_2 is a quantization process. I aim to design a quantization process Σ_2 under the constraint that the procedure of designing the quantization process is fixed. In Chapter 5, the whole system is the segmentation process for gray-scale images, Σ_1 is the TDA process for the segmentation of binary images, and Σ_2 is the binarization process Q, which transform a gray-scale image to a binary image. I fix the design procedure $\mathcal{M}$ for the binarization process and find the binarization algorithm P, such as Otsu's method, random dithering, etc. In Chapter 6, the whole system is the evaluation process of quantized NNs, Σ_1 is the TDA process for quantized NNs, and Σ_2 is the quantization process Q, which quantizes the weights of an original real-valued NNs. I fix the design procedure $\mathcal{M}$ for the quantization process and find the error-diffusion filter P.

Chapter 3

Model-tuning approach to sampled-data dynamic quantizer design

In this chapter, I confirm the usefulness of the proposed method by designing a sampled-data dynamic quantizer for a continuous-time plant. Compared to Fig. 2.2, the subsystem Σ_1 is composed of the continuous-time system G, holder H, and sampler, and the subsystem Σ_2 is a discrete-time dynamic quantizer Q. In this chapter, the continuous-time system G and quantizer Q are linear.

A dynamic quantizer converts a continuous-valued signal into a discrete-valued signal. In the real world, there are some systems controlled by discrete-valued signals, such as systems with built-in ON/OFF actuators[44, 45, 46] and network systems that include digital communication channels[47, 48, 49]. In addition, there are cases where discrete-valued signals are used to compensate for nonlinear elements, such as stick-slip compensation[50, 51, 52]. It is generally difficult to design a control system that includes discrete-valued signals. However, by incorporating a quantizer into the control system, it can be designed like the control system designed for systems without discrete-valued signals. One of these quantizers is a dynamic quantizer that consists of a uniform static quantizer and a linear filter, which has been proposed in many previous studies[43, 53, 54, 55, 56, 57, 58].

In previous studies[54, 56], discrete-time dynamic quantizers for discrete-time plants were designed. These studies analytically derived optimal dynamic quantizers that minimize the maximum value of the output difference between a discrete-valued input system, including dynamic quantizers, and an ideal continuous-valued input system.

On the other hand, the previous works[59, 60, 61] target continuous-time dynamic quantizers consisting of a static quantizer, a continuous-time linear filter, and a sampler and a holder. In one of the previous studies[59], a method for designing dynamic quantizers using invariant set analysis is proposed. In the other studies[60, 61], a dynamic quantizer called Feedback Modulator is proposed based on a Delta-Sigma modulator.

Previous studies have attributed the problem of designing a dynamic quantizer to the problem of designing a discrete-time linear filter when the plant is a discrete-time system and to the problem of designing a continuous-time linear filter when the plant is a continuous-time system. However, the problem of designing a discrete-time linear filter for a continuous-time system has not been adequately discussed. It is important to formulate the problem settings of digital device controls. For example, when implemented within the framework of sampled-data control, "continuous-time dynamic quantizer" would be replaced by "discrete-time dynamic quantizer + sampler and holder." This study examines the problem of designing a discrete-time linear filter for a continuous-time system in the above setting.

In this chapter, I employ a method of discretizing the continuous-time design model and constructing an optimal dynamic quantizer[54] based on the obtained discrete-time model. Therefore, I attribute the problem of designing the linear filter of the dynamic quantizer to the problem of selecting a continuous-time model by giving the design procedure of the dynamic quantizer as a mapping from the design model to a linear filter. The goal is to find a design model such that the behavior of the discrete-valued input system, including the dynamic quantizer, is close to that of an ideal continuous-valued input system.

3.1 Design problem of a sampled-data dynamic quantizer

Figure 3.1 shows the continuous-valued input ideal system and the discrete-valued input system Σ composed of the plant G and the sampled-data quantizer Q_{s}.

The plant G is a continuous-time system given by the following equations:

$$G : \begin{cases} \dot{x}(t) = A_G x(t) + B_G u(t), \\ y(t) = C_G x(t), \end{cases} \tag{3.1}$$

where $x \in \mathbb{R}^n$ is tha state (the initial state is x_0), $u \in \mathbb{R}^m$ is the input, $y \in \mathbb{R}^p$ the output $A_G \in \mathbb{R}^{n\times n}$, $B_G \in \mathbb{R}^{n\times m}$, $C_G \in \mathbb{R}^{p\times n}$ are the coefficient matrices, and A_G is Hurwitz.

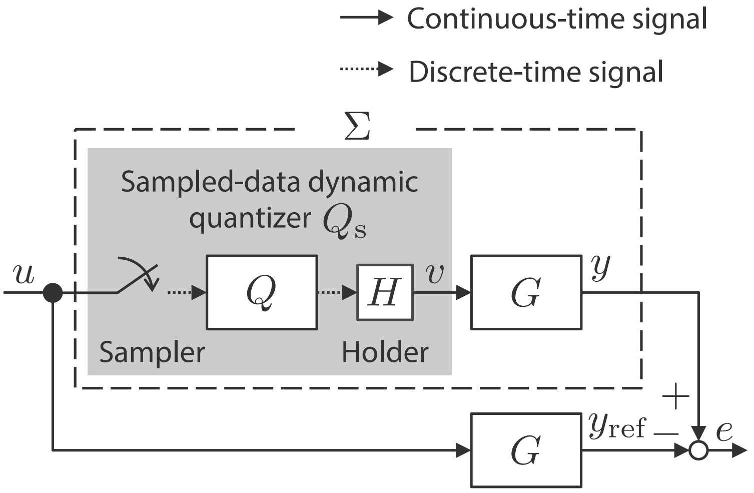


Fig. 3.1: The discrete-valued input system Σ, which is composed of the plant G and the sampled-data quantizer Q_{s}, and the ideal system.

The sampled-data dynamic quantizer Q_{s} in the discrete-valued input system Σ consists of a discrete-time dynamic quantizer Q, a sampler, and a holder H. The discrete-time dynamic quantizer Q is represented by the following equations:

$$Q : \begin{cases} \xi[k+1] = \mathcal{A}\xi[k] + \mathcal{B}(v[k] - u[k]), \\ \quad v[k] = q(\mathcal{C}\xi[k] + u[k]), \end{cases} \tag{3.2}$$

where $k \in \{0\} \cup \mathbb{N}$ is the discrete time, $\xi \in \mathbb{R}^{\mathcal{N}}$ is the state of Q (the initial state is $\xi[0] = 0$), $u \in \mathbb{R}^m$ is the continuous-valued signal, $v \in \mathbb{V}^m$ is the discrete-valued signal, $\mathcal{A} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $\mathcal{B} \in \mathbb{R}^{\mathcal{N}\times m}$, $\mathcal{C} \in \mathbb{R}^{m\times\mathcal{N}}$ are the coefficient matrices, $q : \mathbb{R}^m \to \mathbb{V}^m$ is the static uniform quantizer. Note that the number of quantization bits is $N_{\mathrm{bit}} = \infty$ in this chapter. When I represent the quantization error caused by the static quantizer q as $w[k] \coloneqq v[k] - u[k]$, the discrete-valued signal $v[k]$ can be calculated by

$$v(z) = \mathcal{L}(z)w(z) + u(z), \tag{3.3}$$

$$\mathcal{L}(z) = \mathcal{C}\{zI - (\mathcal{A} + \mathcal{B}\mathcal{C})\}^{-1}\mathcal{B} + 1, \tag{3.4}$$

where $\mathcal{L}(z)$ is the discrete-time linear filter. The stability of the linear filter $\mathcal{L}(z)$ is equivalent to the Schur of $\mathcal{A} + \mathcal{B}\mathcal{C}$ [62, 63].

The continuous-valued signal $u(t)$ is converted to the discrete-valued signal $u[k] = u(kT_{\mathrm{s}})$ with the sampling period $T_{\mathrm{s}} \in \mathbb{R}_+$ by the sampler. The input of the continuous-time system G is converted to the continuous-valued signal $v(t) = v(kT_{\mathrm{s}}) \quad (kT_{\mathrm{s}} \leq t < (k+1)T_{\mathrm{s}})$ by the holder H.

I consider the design of the quantizer Q, which minimizes the error $e = y - y_{\mathrm{ref}}$ between the output y of the system Σ in Fig. 3.1 and the output y_{ref} of the ideal system in this chapter. To evaluate the performance of the quantizer Q, I define the evaluation function J as

$$J(Q; u(t), x_0) := \int_0^T \|y(t) - y_{\mathrm{ref}}(t)\|_2 \,\mathrm{d}t = \int_0^T \|e(t)\|_2 \,\mathrm{d}t, \tag{3.5}$$

where $T \in \mathbb{R}_+ \cup \{\infty\}$ is the evaluation interval. The evaluation function J is the function that depends on the dynamic quantizer Q, the input $u(t) \in L_2[0, T]$, and the initial state x_0. I can make the output of the system Σ closer to the output of the ideal continuous-valued input system by using the optimal dynamic quantizer, which minimizes the evaluation function J. Note that this paper uses the L_2 norm to evaluate the impact of small quantization errors on the output over the entire evaluation interval. However, other norms, such as the L_∞ norm, can also be used.

The design problem for the quantizer Q is formulated as follows:

[Problem 3.1]

Suppose that the continuous-time plant G, the quantization width $d \in \mathbb{R}_+$, the sampling period $T_{\mathrm{s}} \in \mathbb{R}_+$, the holder H, the input siglnal $u(t)$, and the initial state x_0 are given in the discrete-valued input system Σ. Then, find the stable discrete-time dynamic quantizer Q minimizing the evaluation function $J(Q; u(t), x_0)$.

3.2 Model-tuning approach to design a dynamic quantizer

3.2.1 Design of a quantizer using a discretized model for a linear system

I design a discrete-time dynamic quantizer Q for the continuous-time plant G using the same method as in the previous study[54]. First, the continuous-time plant G is converted to the discrete-time plant G_{d} by Zero Order Hold (ZOH), which is expressed as

$$G_{\mathrm{d}} : \begin{cases} x[k+1] = e^{A_G T_{\mathrm{s}}} x[k] + \displaystyle\int_0^{T_{\mathrm{s}}} e^{A_G t} \,\mathrm{d}t B_G u[k], \\ \quad y[k] = C_G x[k]. \end{cases} \tag{3.6}$$

Then, combining ZOH and Equation (23) in the previous study [54], which converts a discrete-time model G_{d} to an optimal dynamic quantizer, the equation to convert a continuous-time model to a dynamic quantizer is expressed as

$$Q : \begin{cases} \mathcal{A} = e^{A_G}, \\ \mathcal{B} = \int_0^{T_\mathrm{s}} e^{A_G t}\,\mathrm{d}t B_G, \\ \mathcal{C} = -\left(C_G \int_0^{T_\mathrm{s}} e^{A_G t}\,\mathrm{d}t B_G \right)^{-1} C_G e^{A_G}. \end{cases} \tag{3.7}$$

As an example, I designed a dynamic quantizer Q for the continuous-time plant G (the true plant) expressed as

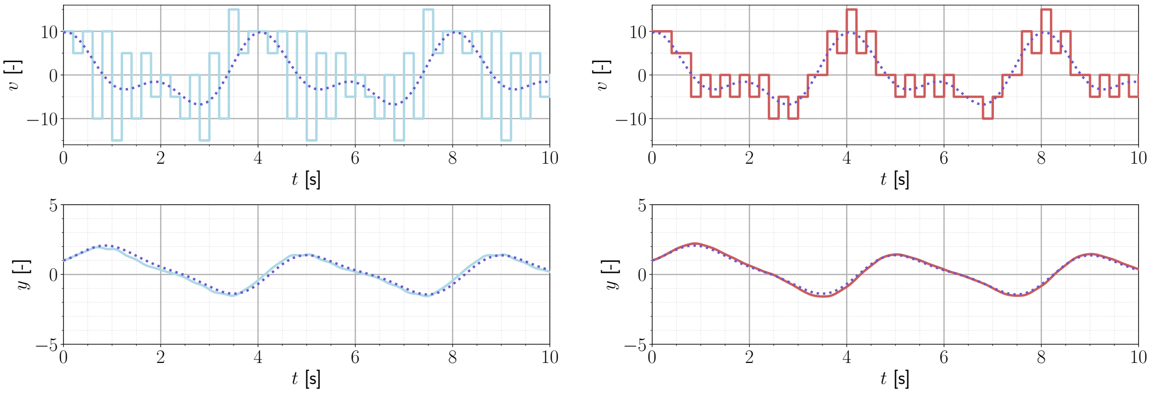
$$G : A_G = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, B_G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_G = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{3.8}$$

by Equation (3.7). The input and output in the case that the quantization width is $d = 5\,[\text{-}]$, the sampling period is $T_\mathrm{s} = 0.2[\mathrm{s}]$, and the input is $u(t) = 6\sin(0.5\pi t + 0.4\pi) + 4\cos\pi t$, which are shown in Fig. 3.2(a). The solid light blue line in Fig. 3.2(a) represents the signal of the discrete-value input system in the upper part of Fig. 3.1. The dotted blue line represents the signal of the continuous-value input system consisting only of G in the lower part of Fig. 3.1 (hereafter, I call the system an ideal system). The upper and lower figures show the time evolution of the inputs and outputs, respectively. Comparing the respective lines, the value of the output $y(t)$ and the output of the ideal system $y_\mathrm{ref}(t)$ are close. The value of the evaluation function J represented by the expression (3.5) is 1.61 when the evaluation interval is $T = 10$.

Now, the design specification is to make the output of the discrete-valued input system closer to the output of the ideal system G, and the design model is selected. As an example of the design models, the design model P

$$P : A = \begin{bmatrix} 0 & 1 \\ -80 & -10 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} -50 & 1 \end{bmatrix}, \tag{3.9}$$

differed from the plant G is used to design the quantizer. The solid red line in Fig. 3.2(b) represents the signal of the discrete-value input system, and the dotted blue line represents the signal of the ideal system, and the value of the evaluation function J represented by the expression (3.5) is 0.851. As a result, I can see that the value of the evaluation function J using the design model P is smaller than that using the true plant G. One reason is that the sampler and the holder are in the control system, which differs from the design of the dynamic quantizer for the discrete-time



(a) In the case of the quantizer designed using the model G.

(b) In the case of the quantizer designed using the model P

Fig. 3.2: The input and the output of the ideal system and the system with the quantizer under the condition that the sampling period $T_{\rm s}$ is 0.2[s].

system. This result implies that when designing a dynamic quantizer for a continuous-time plant, a design model P different from the plant G can be used as a model for designing the dynamic quantizer, which can make the difference between the output of the ideal system and the output of the system with a dynamic quantizer smaller. I consider a method to design a dynamic quantizer by tuning the continuous-time design model.

3.2.2 Problem formulation of model-tuning approach to quantizer design

In this chapter, I design a dynamic quantizer by discretizing the continuous-time design model and constructing an optimal dynamic quantizer from the discrete-time model. In other words, I design a dynamic quantizer by substituting a continuous-time design model $P(A, B, C)$ into the following equation

$$Q\ :\ \begin{cases} \mathcal{A} = e^{AT_{\rm s}}, \\ \mathcal{B} = \displaystyle\int_0^{T_{\rm s}} e^{At}\,{\rm d}tB, \\ \mathcal{C} = -\left(C\displaystyle\int_0^{T_{\rm s}} e^{At}\,{\rm d}tB\right)^{-1} Ce^{AT_{\rm s}}. \end{cases} \tag{3.10}$$

The design model P is determined as the continuous-time model minimizing the evaluation function J represented by Equation (3.5).

[Problem 3.2]

Suppose that the plant G and the design procedure $\mathcal{M} : P \mapsto Q$ for a dynamic quantizer are given. Find the design model $P^\star$ minimizing the evaluation function $J(Q; u(t), x_0) = J(\mathcal{M}(P); u(t), x_0)$ expressed as Equation (3.5) to make the output of the ideal system close to the output of the system Σ including the dynamic quantizer.

By minimizing the evaluation function in Equation (3.5), I can obtain a dynamic quantizer that makes the output of the system Σ close to that of the ideal system, including the behavior between sample points. A candidate solution to **[Problem 3.2]** is the plant model G, but it is possible to obtain a different model from G by optimizing the evaluation function J. In addition, various dynamic quantizers in a broader class can be designed by fixing the order and structure of design model P differed from those of the plant model G and optimizing it.

3.3 Illustrative example

In this section, an illustrative example of finding the design model for the **[Problem 3.2]** in the section 3.2.2 by numerical optimization.

The plant G is given by Equation (3.8) in the same way as the subsection 3.2.1. The structure of a design model is set as a second-order system expressed in the following equation

$$P : A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} c_0 & c_1 \end{bmatrix}, \tag{3.11}$$

like the plant G. The design parameters for the optimization are $[a_0,\, a_1,\, c_0,\, c_1]^\top$.

In this example, the quantization width, the sampling period, the initial state of the plant, and the evaluation time of the simulation are set as $d = 5\,[\text{-}], T_\mathrm{s} = 0.2\,[\mathrm{s}], x_0 = [1.0\ 1.0]^\top$, and $T = 10.0\,[\mathrm{s}]$, respectively. To design a stable dynamic quantizer Q, I modify the evaluation function J as

$$\bar{J}(Q) := \begin{cases} \tan^{-1} J(Q) - \dfrac{\pi}{2} & \text{if } |p_1|, \ldots, |p_\mathcal{N}| < 1, \\ \max(|p_1|, \ldots, |p_\mathcal{N}|) - 1 & \text{otherwise}, \end{cases} \tag{3.12}$$

where $p_1, \ldots, p_\mathcal{N}$ are the eigenvalues of the matrix $\mathcal{A} + \mathcal{BC}$. The dynamic quantizer Q minimizing the evaluation function J in Equation (3.5) must be selected from the class of stable dynamic quantizers. Thus, I optimize the evaluation function $\bar{J}$ in Equation (3.12) to fulfill the conditions that all absolute values of the eigenvalues $p_1, \ldots, p_N$ of $\mathcal{A} + \mathcal{BC}$ are less than 1 because the eigenvalues determine the stability of the dynamic quantizer Q. In this chapter, I use the modified evaluation function $\bar{J}$ in Equation (3.12) to convert the constrained optimization problem into an unconstrained optimization problem.

Covariance Matrix Adaption Evolution Strategy (CMA-ES)[64, 65], a stochastic and heuristic method, is used as the optimization method, and the hyperparameters of optimization were set as the Table 2 in the paper [65]. As a result, the optimal design model $P^\star$ parameters are

$$a_0^\star = 71.9, a_1^\star = 18.0, c_0^\star = -34.5, c_1^\star = 1.60. \tag{3.13}$$

The light blue and red lines in Fig. 3.3 represent the error between the output of the ideal system and the system with the quantizer designed using the plant G and the optimal design model $P^\star$, respectively, and the error is smaller when the optimal design model is used. The input and output of the system with the quantizer designed using the plant G are shown in Fig. 3.2(a), and the input and output of the system with the quantizer designed using the optimal design model $P^\star$ are shown in Fig. 3.4. The value of the evaluation function J using the optimal design model $P^\star$ is 0.273, and the value using the model (3.9) is 0.851 (Fig. 3.2(b)). I can see that the performance of the dynamic quantizer is improved compared to the design example in the subsection 3.2.1.

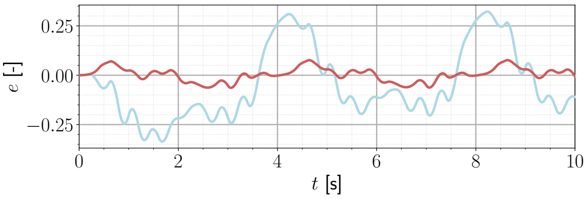


Fig. 3.3: The error $e(t)$ between the output $y(t)$ and the reference output $y_{\mathrm{ref}}(t)$ under the condition that the sampling period T_{s} is 0.2[s].

In addition, when I change the sampling period to $T_{\mathrm{s}} = 0.02\,[\mathrm{s}]$ and optimize a design model, the parameters of the optimal design model $P^\star$ are

$$a_0^\star = 35.8, a_1^\star = 9.89, c_0^\star = -42.2, c_1^\star = -31.0. \tag{3.14}$$

The input and output of the system with the quantizer designed using the plant G and the optimal design model $P^\star$ are shown in Fig. 3.5(a) and Fig. 3.5(b), respectively. The error between the output of the ideal system and the system with the quantizer is shown in Fig. 3.6. I can also see that the output of the system with the quantizer designed using the optimal design model $P^\star$ is close to the output of the ideal system.

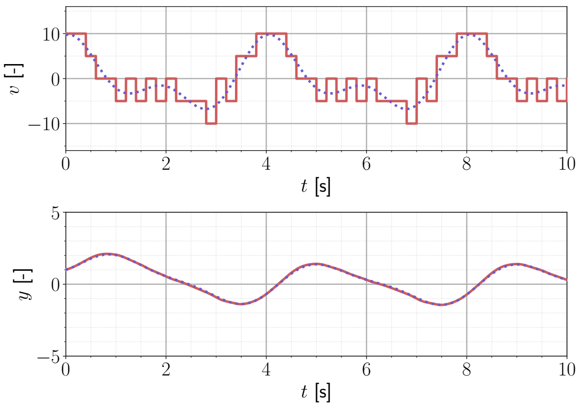
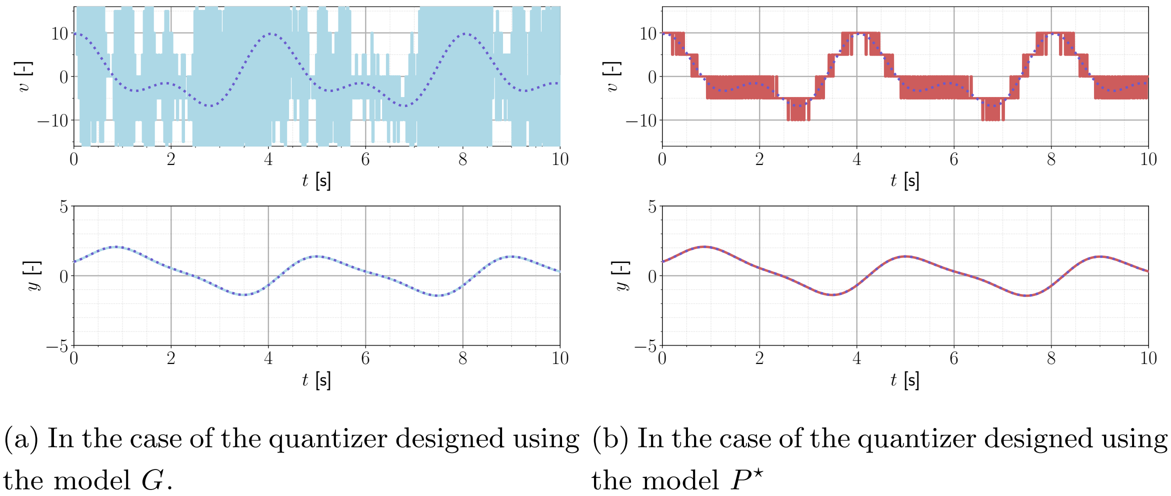


Fig. 3.4: The input and the output of the ideal system and the system with the quantizer designed using the model $P^{\star}$ under the condition that the sampling period $T_{\rm s}$ is 0.2[s].



(a) In the case of the quantizer designed using the model G.

(b) In the case of the quantizer designed using the model $P^{\star}$

Fig. 3.5: The input and the output of the ideal system and the system with the quantizer under the condition that the sampling period $T_{\rm s}$ is 0.02[s].

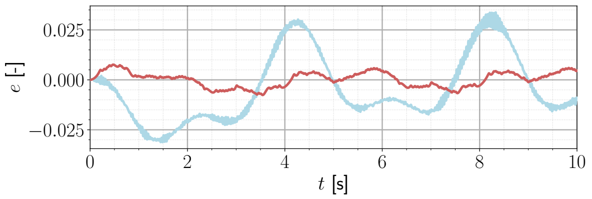


Fig. 3.6: The error $e(t)$ between the output $y(t)$ and the reference output $y_{\rm ref}(t)$ under the condition that the sampling period $T_{\rm s}$ is 0.02[s].

Finally, I summarize the Bode diagrams, poles, and zeros for the numerical examples. The Bode plots of the plant G expressed by Equation (3.8) and the design models P expressed by Equation (3.9), (3.13), and (3.14) are shown in Fig. 3.7. In Fig. 3.7, the gain and phase diagrams of Equation (3.9) and (3.13) with the same sampling period 0.2[s] are similar. In the Bode plot of the design model expressed by Equation (3.14) with the sampling period 0.02[s], the DC gain and bandwidth are larger in the gain plot, and the phase lag is smaller mainly in the high-frequency region in the phase plot compared to the case with the sampling period 0.2[s]. In summary, the sampling period changes the DC gain of the design model and phase lag in the high-frequency region.

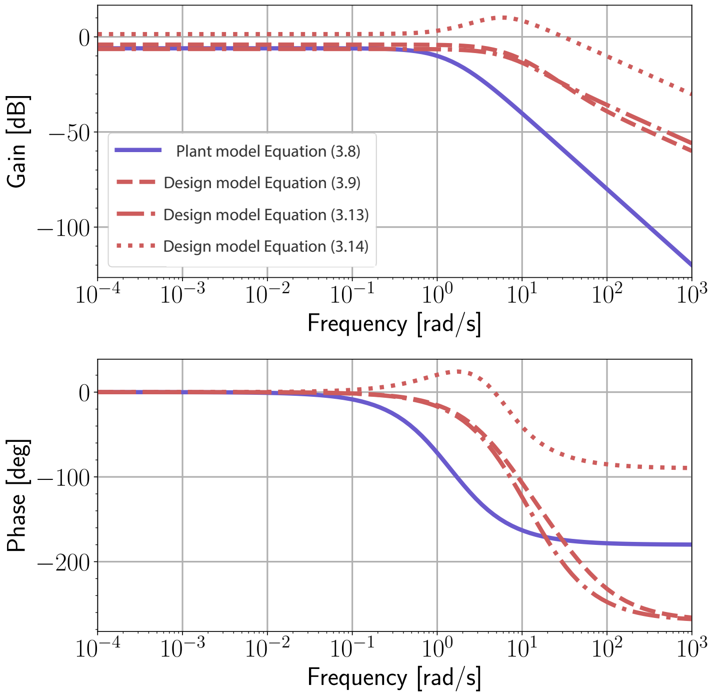
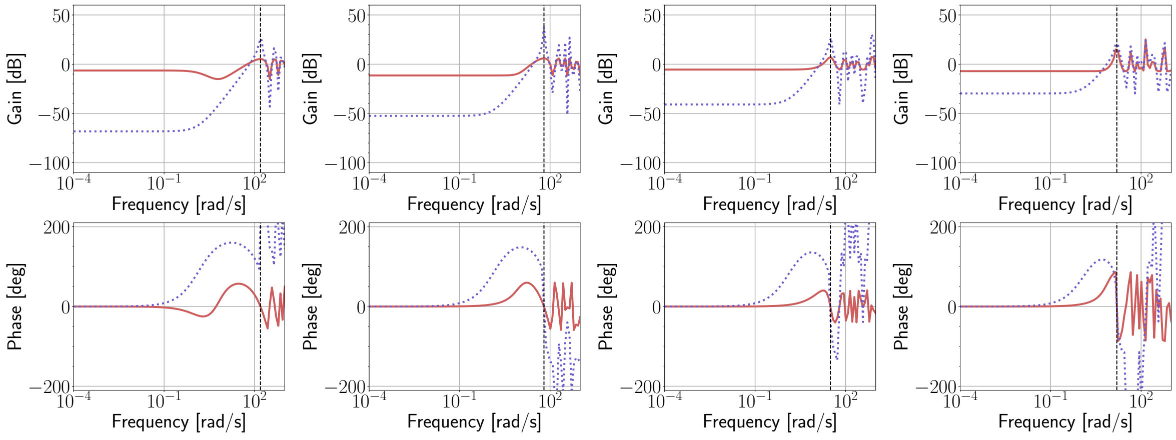


Fig. 3.7: The bode plot of the plant G and design models P.

3.4 Discussion of the model-tuning approach

3.4.1 Relationship between the sampling period T_s and the design model $P^\star$

In the previous section, I simulate the illustrative example where the sampling period T_s is fixed as 0.2 [s] and 0.02 [s]. Note that the performance of the quantizer also depends significantly on the sampling period. To evaluate the performance of the



(a) $\Delta = 0.02$ [s]

(b) $\Delta = 0.05$ [s]

(c) $\Delta = 0.10$ [s]

(d) $\Delta = 0.20$ [s]

Fig. 3.8: The bode plot of the transfer function $\mathcal{L}(z)$ from w to y in case of quantization width $d = 5$ under changing the sampling period T_{s}.

quantizer, the frequency-domain properties of the transfer function $\mathcal{L}$ from the quantization error w to the discrete input v expressed by Equation (3.3) are essential. In this subsection, I compare the frequency-domain properties of the transfer function $\mathcal{L}$ for various sampling periods T_{s}.

The bode plot is shown in the Fig. 3.8 when the quantization width is fixed as $d = 5\,[\text{-}]$ and the sampling period is changed to $T_{\mathrm{s}} = 0.02, 0.05, 0.10, 0.20\,[\mathrm{s}]$. The blue dotted line represents the Bode plot when the dynamic quantizer is designed using the plant G, the red solid line represents the Bode plot when the dynamic quantizer is designed using the design model obtained by optimization, and the black dashed line represents the Nyquist frequency corresponding to each sampling period T_{s}. In Fig. 3.8, the gain plot of the linear filter $\mathcal{L}(z)$ shows that the gain of the optimal design model is larger in the low-frequency region, and the difference in gain between the design model and the plant is smaller in the high-frequency region. On the other hand, the phase plot shows that the phase of the design model and the plant generally match in the low-frequency region, but the difference in phase between the design model and the plant is larger in the high-frequency region. The Bode plots of the design model differ depending on the sampling period T_{s}, and in particular, the difference of DC gain decreases in the gain plot, and the position of the phase maximum shifts in the high-frequency region as the sampling period increases. Thus, I found that the difference between the design model and the plant is significant in the low-frequency region of the gain plot and the high-frequency region of the phase plot for each sampling period T_{s}.

3.4.2 Comparison of the proposed method and the direct search of the parameters of the dynamic quantizer

In this section, I propose the model-tuning method, but it is also possible to search the parameters of the dynamic quantizer directly. Here, I compare the search performance of the solutions in the proposed method and the direct search method. In addition to the minimum phase system expressed in Equation (3.8), I also adopt the non-minimum phase system as a plant expressed in the following equation

$$G\,:\,A_G = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, B_G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_G = \begin{bmatrix} -1 & 1 \end{bmatrix}. \tag{3.15}$$

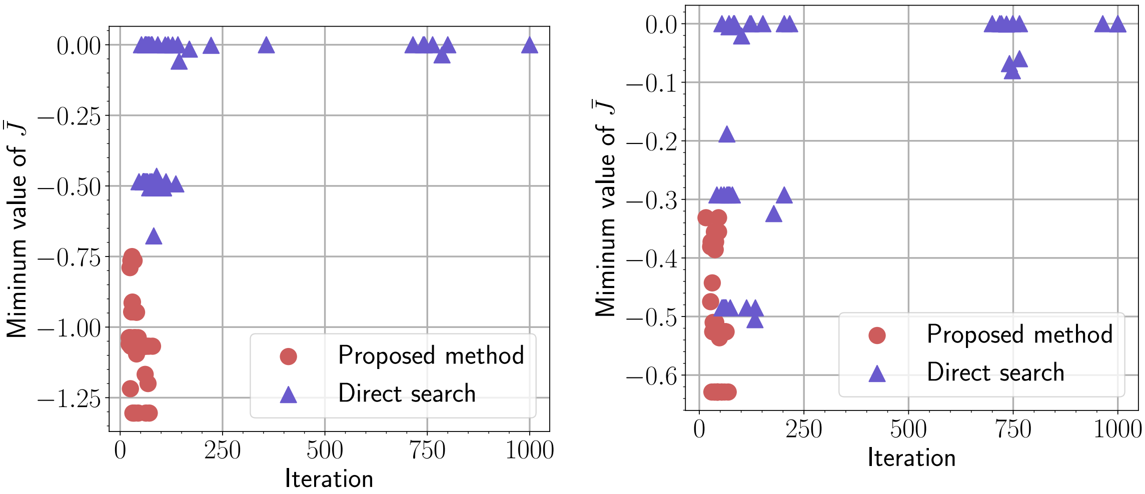
If the design model for the dynamic quantizer is a non-minimum phase system, the dynamic quantizer designed by the procedure in Equation (3.7) becomes unstable. In general, the design problem of the dynamic quantizer for the non-minimum phase system is more difficult than that for the minimum phase system. Therefore, I verify that a stable dynamic quantizer can be designed by the proposed method.

I fix the quantization width as $d = 5\,[\text{-}]$ and the sampling period as $T_\mathrm{s} = 0.2\,[\mathrm{s}]$, and I perform optimization 50 times using CMA-ES for the minimum phase system expressed in Equation (3.8) and the non-minimum phase system expressed in Equation (3.15) to search the design model. In the case of each optimization for the minimum phase system and the non-minimum phase system by the proposed method, the minimum value of the evaluation function $\bar{J}$ and the number of trials are plotted in Fig. 3.9(a) and Fig. 3.9(b) as red circles, respectively. The upper row of Table 3.1 shows the statistical data of the value of the evaluation function $\bar{J}$ obtained as a result of the optimization by the proposed method.

In the direct search, I set the parameters $a_{Q0}, a_{Q1}, c_{Q0}, c_{Q1}$ as optimization parameters, and the following equation

$$\mathcal{A} = \begin{bmatrix} 0 & 1 \\ -a_{Q0} & -a_{Q1} \end{bmatrix},\ \mathcal{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathcal{C} = \begin{bmatrix} c_{Q0} & c_{Q1} \end{bmatrix}, \tag{3.16}$$

represents $\mathcal{A}, \mathcal{B}, \mathcal{C}$ as the coefficient matrices of the dynamic quantizer. I fix the quantization width as $d = 5\,[\text{-}]$ and the sampling period as $T_\mathrm{s} = 0.2\,[\mathrm{s}]$, and I perform optimization 50 times using the direct search method for the minimum phase system expressed in Equation (3.8) and the non-minimum phase system expressed in Equation (3.15), respectively. In the case of each optimization for the minimum



(a) In the case of the minimum phase system.

(b) In the case of the non-minimum phase system.

Fig. 3.9: Minimum value of the Evaluation function $\bar{J}$ and iteration.

phase system and the non-minimum phase system by the direct search method, the minimum value of the evaluation function $\bar{J}$ and the number of trials are plotted in Fig. 3.9(a) and Fig. 3.9(b) as blue triangles, respectively. The lower row of Table 3.1 shows the statistical data of the value of the evaluation function $\bar{J}$ obtained as a result of the optimization by the proposed method. When I compare the Fig. 3.9(a) and 3.9(b), the value of the evaluation function is smaller in the proposed method, and the convergence to the optimal parameters is faster in both the minimum phase system and the non-minimum phase system. One of the reasons for this is the following discussion. In the direct search method, it is necessary to search the vast parameter space of $\mathbb{R}^4$. However, the proposed method may limits the parameter space by the mapping in Equation (3.7), and the efficient search can be performed.

3.5 Summary

In this chapter, I proposed a method for designing a discrete-time dynamic quantizer for a continuous-time system by extending the optimal dynamic quantizer design method for discrete-time systems in the previous study [56]. Specifically, I fixed the design method for the optimal dynamic quantizer for discrete-time systems and the method for converting continuous-time models to discrete-time models. I also tuned the continuous-time model to design a dynamic quantizer for continuous-time systems.

Here, I give a supplementary explanation about the benefits of reducing the design

Table 3.1: Settings and values of the evaluation function.

Method	Plant	Values of the evaluation function (3.12)			
		Best	Worst	Mean	St. dev.
Proposed method	Minimum phase system	−1.30	−0.750	−1.04	0.135
	Non-minimum phase system	−0.629	−0.331	−0.520	0.089
Direct search	Minimum phase system	−0.677	0.00	−0.260	0.249
	Non-minimum phase system	−0.505	0.00	−0.194	0.208

problem of the dynamic quantizer to the design model search problem. One benefit is that the characteristics of the design model that determines the filter of the optimal dynamic quantizer can be understood. Moreover, it is possible that searching for the design model is more efficient than directly searching for the dynamic quantizer to find the optimal solution. Furthermore, the method proposed in this study can be applied to a closed-loop system, including a continuous-time controller. In the previous study [56], the optimal dynamic quantizer was designed for the expanded system of the discrete-time plant and the discrete-time controller. Therefore, by applying the proposed method to the design model of the continuous-time expanded system and designing the dynamic quantizer from the obtained design model, it is possible to design the dynamic quantizer in the closed-loop system.

One of the future works is to attribute the design problem of the nominal model to the design model search problem for a system with uncertainty. In the previous works [66, 67], an approach to control system design by selecting the design model is being considered. This study proposed the controller design method for the system with polytope-type uncertainty. The method fixes the controller design method and design the controller by searching for the model. By considering the design of the dynamic quantizer when there is uncertainty in the plant, it is possible to utilize the knowledge of the previous studies [66, 67]. Another prospect is to challenge optimization with an analytical approach and design the dynamic quantizer for practical problems.

Chapter 4

Model-tuning approach to switching-type dynamic quantizer design

In this chapter, I confirm the usefulness of the proposed method by designing a continuous-time plant and a discrete-time dynamic quantizer. Compared to Fig. 2.2, the subsystem Σ_1 is composed of the continuous-time system G, holder H, and sampler, and the subsystem Σ_2 is a discrete-time quantizer Q. The continuous-time system G is nonlinear, and the quantizer Q is a switching-type quantizer, which is composed of a linear quantizer$Q_1, Q_2, \ldots, Q_N$ in this chapter.

Quantizers are classified into two types: static and dynamic. Dynamic quantizers are generally able to achieve higher performance than static quantizers. There are two types of dynamic quantizers: (a) the case that the quantization width is changed, (b) the case that the quantizer have memory and feedback structure. In the case of (a), zooming-in and zooming-out as time-varying dynamic quantizers, in which the quantization width varies with time, are known in network control[68, 69, 70]. In the case of (b), the quantization errors are fed back and converted to discrete values by filtering them. There is much research on dynamic quantizers with memory and feedback structure[47, 43, 54, 56, 58, 61, 71, 72]. In this chapter, I call the quantizer of the case (b) as a dynamic quantizer.

Many studies have been made of dynamic quantizer design for *linear* systems. One of the studies for the dynamic quantizer is using time-invariant linear filters[43, 54, 56, 58, 47, 61, 71].

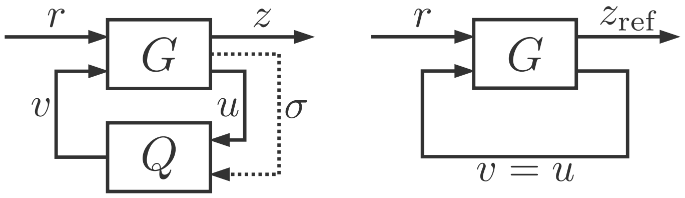
On the other hand, there have also been studies of dynamic quantizers for *nonlinear* systems. One dynamic quantizer design for nonlinear systems uses a time-invariant nonlinear filter. Azuma and Sugie have addressed the problem of designing a dynamic quantizer for a nonlinear system in case the output signal set is fixed[72]. The key idea of this quantizer design is to copy the model information of a system to dynamic quantizers. Thus, I need to implement the nonlinear function, which precisely captures the dynamics of systems, in a computer. In addition, the quantizer proposed in the previous study[72] is optimized for a given discrete-time nonlinear system, and it is not directly applied to continuous-time nonlinear systems.

For nonlinear systems, this chapter designs a dynamic quantizer with a linear filter rather than a nonlinear filter that reflects the properties of the system. However, applying a single linear model to complex nonlinear systems may degrade performance. In the design of controllers, many nonlinear systems cannot be stabilized with a state feedback controller but can be stabilized with switching control schemes[73, 74]. My approach is to prepare multiple linear models for nonlinear systems and design a dynamic quantizer that switches between these models appropriately.

The previous works[43, 54, 56] have shown that the inverse model of a discrete-time linear system gives the optimal filter for a dynamic quantizer. According to these points, some linear models can characterize the dynamic quantizer for nonlinear systems. In summary, if I use the results of previous studies and assume that the dynamic quantizer is constructed from linear models, the quantizer design problem is attributed to the problem of searching for multiple linear models. The design approach of this study is based on this concept. In this problem, the gap between nonlinear and linear, as well as between continuous time and discrete time, must be considered in the design. The interesting aspect of this approach is to fill these gaps with linear models.

4.1 Design problem of a switching-type dynamic quantizer

Consider the two general nonlinear feedback systems illustrated in Fig. 4.1: (a) is a feedback system composed of a continuous-time nonlinear system G and a switching-type dynamic quantizer Q and (b) is a feedback system without quantization, called ideal system.



(a) The feedback system composed of the nonlinear system G and the switching-type quantizer Q.

(b) The feedback system without a quantizer.

Fig. 4.1: Two feedback systems.

The continuous-time nonlinear system G is given by

$$G:\begin{cases}\dot{x}(t)=f(x(t),r(t),v(t)),\\ z(t)=g(x(t),r(t)),\\ \sigma(t)=\phi(x(t),r(t),v(t)),\\ u(t)=h(x(t),r(t),\sigma(t)),\end{cases}\tag{4.1}$$

where $x\in\mathbb{R}^n$ is the state, $r\in\mathbb{R}^p$ is the reference input, $v\in\mathbb{V}^m$ are the control input, $\sigma\in\mathbb{S}$ is the output for switching, $z\in\mathbb{R}^l$ is the control output, and $u\in\mathbb{R}^m$ is the observation output. Note that the number of quantization bits is $N_{\rm bit}=\infty$ in this chapter. The set $\mathbb{S}$ is assumed to be the subset of the s-dimensional Euclidean space $\mathbb{R}^s$. The functions $f:\mathbb{R}^n\times\mathbb{R}^p\times\mathbb{R}^m\to\mathbb{R}^n, g:\mathbb{R}^n\times\mathbb{R}^p\to\mathbb{R}^l, h:\mathbb{R}^n\times\mathbb{R}^p\times\mathbb{S}\to\mathbb{R}^m$ are assumed to be smooth. $\phi:\mathbb{R}^n\times\mathbb{R}^p\times\mathbb{R}^m\to\mathbb{S}$ is a mapping. The characteristics of the system G determine the mapping ϕ. I consider a swing-up and stabilization control system of a cart-type inverted pendulum as an example. To switch between swing-up controller and stabilizing controller depending on the angle, the output of the switching σ is in the interval $\mathbb{S}:=[-\pi,\pi]$. In this case, the mapping ϕ is the function that round the values of the angle of the pendulum to the set $\mathbb{S}$.

On the other hand, the sampled-data and switching-type dynamic quantizer Q (hereinafter referred to as the switching-type quantizer) is depicted in Fig. 4.2. In Fig. 4.2, the subsystem Σ_1 is composed of the continuous-time system G, holder H, and sampler, the subsystem Σ_2 is a quantizer Q. The switching-type quantizer Q is composed of a sampler, a holder H, and $N\in\mathbb{N}$ dynamic quantizers $Q_i,\ i\in\mathbb{I}:=$

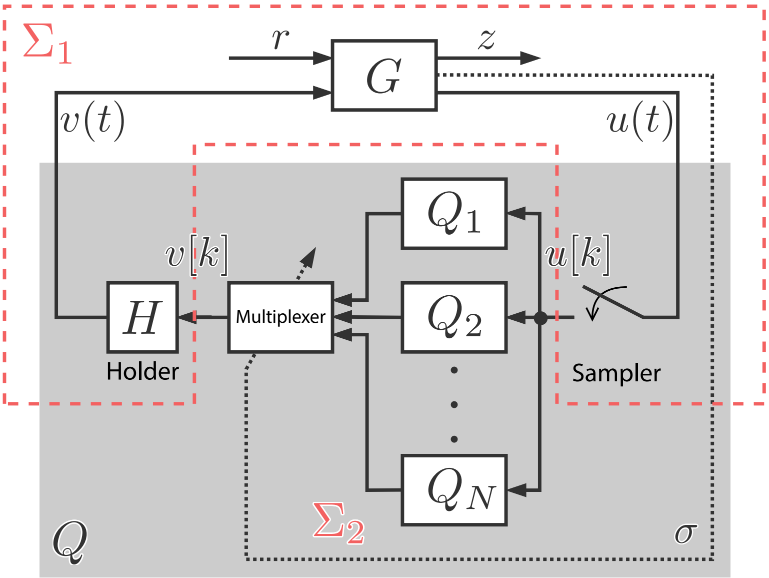


Fig. 4.2: The sampled-data and switching-type dynamic quantizer.

$\{1, 2, \ldots, N\}$. The each sub-quantizer Q_i is denoted as

$$Q_i \ : \begin{cases} \xi_i[k+1] = \mathcal{A}_i\xi_i[k] + \mathcal{B}_i(v_i[k] - u[k]), \\ \qquad v_i[k] = q(\mathcal{C}_i\xi_i[k] + u[k]), \end{cases} \tag{4.2}$$

where $k \in 0 \cup \mathbb{N}$ is the discrete time, $\xi_i \in \mathbb{R}^{\mathcal{N}_i}$ is the state, the initial state is $\xi_i[0] = 0$, and the function $q(\cdot)$ is the static uniform quantizer with quantization width d. The coefficient matrices, which determine the performance of each sub-quantizer Q_i, $\mathcal{A}_i \in \mathbb{R}^{\mathcal{N}_i \times \mathcal{N}_i}, \mathcal{B}_i \in \mathbb{R}^{\mathcal{N}_i \times m}, \mathcal{C}_i \in \mathbb{R}^{m \times \mathcal{N}_i}, i \in \mathbb{I}$ are design parameters. The quantization error $\epsilon_i[k]$ generated in Q_i is denoted as

$$\begin{aligned} \epsilon_i[k] &= q(\mathcal{C}_i\xi_i[k] + u[k]) - (\mathcal{C}_i\xi_i[k] + u[k]) \\ &= v_i[k] - (\mathcal{C}_i\xi_i[k] + u[k]). \end{aligned}$$

If the quantization error $\epsilon_i[k]$ and the continuous-valued signal $u[k]$ are regarded as inputs, the discrete-valued signal $v_i[k]$ is represented as

$$v_i(\mathsf{z}) = \mathcal{L}_i(\mathsf{z})\epsilon_i(\mathsf{z}) + u(\mathsf{z}), \tag{4.3}$$

using a linear filter $\mathcal{L}_i(\mathsf{z}) := \mathcal{C}_i\{\mathsf{z}I - (\mathcal{A}_i + \mathcal{B}_i\mathcal{C}_i)\}^{-1}\mathcal{B}_i + I$. Note that $\mathsf{z} \in \mathbb{C}$ is the complex number for z-transformation and different for the control output z of system G. The linear filter $\mathcal{L}_i$ in state-space representation is given by

$$\mathcal{L}_i : \begin{cases} \xi_i[k+1] = (\mathcal{A}_i + \mathcal{B}_i\mathcal{C}_i)\xi_i[k] + \mathcal{B}_i\epsilon_i[k], \\ \qquad v_i[k] = \mathcal{C}_i\xi_i[k] + \epsilon_i[k]. \end{cases} \tag{4.4}$$

I can see that the linear filter $\mathcal{L}_i$ is stable if the matrix $\mathcal{A}_i + \mathcal{B}_i\mathcal{C}_i$ is Schur.

The input u is sampled every T_{s} seconds by the sampler such as $u[k] = u(kT_{\mathrm{s}})$. The sampling period T_{s} is given in advance. The zero-order holder H is

$$H \,:\, v(t) = v[k] \quad kT_{\mathrm{s}} \le t < (k+1)T_{\mathrm{s}}. \tag{4.5}$$

The multiplexer is expressed as

$$v[k] = \sum_{i=1}^{N} \mathbf{1}_{\mathbb{W}_i}(\sigma[k])v_i[k], \tag{4.6}$$

where $\mathbf{1}_{\mathbb{W}_i}(\cdot)$ is the indicator function expressed as

$$\mathbf{1}_{\mathbb{W}_i}(\sigma) = \begin{cases} 1 & \text{if } \ \sigma \in \mathbb{W}_i, \\ 0 & \text{otherwise,} \end{cases} \tag{4.7}$$

for the sets $\mathbb{W}_i, i \in \mathbb{I}$. The set $\mathbb{W}_i$ is the switching condition for the sub-quantizer Q_i and defined using the functions $w_i, i \in \mathbb{I} \setminus \{N\}$. The input for functions of switching conditions $w_i(\sigma)$ is the output for switching $\sigma \in \mathbb{S} \subset \mathbb{R}^s$. The sets $\mathbb{W}_i$ derived from the functions of switching conditions w_i satisfy the following conditions:

$$\begin{aligned} \mathbb{W}_i &:= \{\sigma \in \mathbb{S} \mid w_i(\sigma) > 0\} \ (i = 1, \dots, N-1), \\ \mathbb{W}_N &:= \bigcap_{i=1}^{N-1} \mathbb{S} \setminus \mathbb{W}_i, \\ \mathbb{W}_i \cap \mathbb{W}_j &= \emptyset \ (i \neq j), \\ \bigcup_{i=1}^{N} \mathbb{W}_i &= \mathbb{S} \subset \mathbb{R}^s. \end{aligned} \tag{4.8}$$

The meaning of Equation (4.6), ..., (4.8) is as follows:

- Each value of the signal $\sigma \in \mathbb{S}$ determines the only one set $\mathbb{W}_i$.
- Each set $\mathbb{W}_i$ corresponds to the sub-quantizer Q_i.

I suppose that the nonlinear system G, the sampling period T_{s}, the zero-order holder H, the quantization width d, the initial value x_0, reference input r are given. Then, I design the switching-type dynamic quantizer Q that minimizes the evaluation function:

$$J(Q) = \int_0^T \|z(t) - z_{\mathrm{ref}}(t)\|_2 \,\mathrm{d}t, \tag{4.9}$$

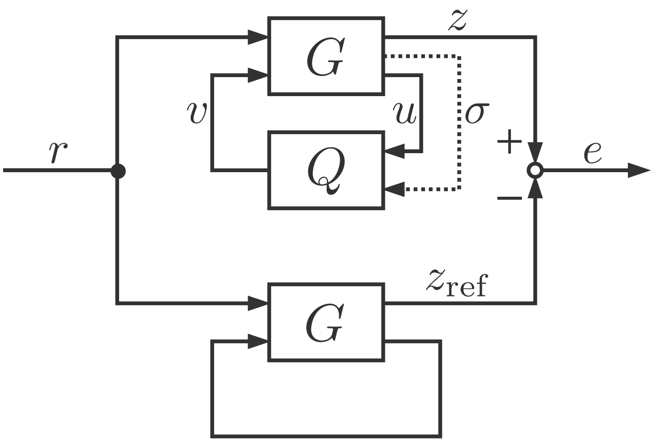


Fig. 4.3: The error system composed of discrete-valued and continuous-valued input systems.

where $T > 0$ is the evaluation time and z_{ref} is the output of the ideal system, i.e., of the system without the dynamic quantizer in Fig. 4.3. Also, I denote the eigenvalues of the matrices $\mathcal{A}_i + \mathcal{B}_i\mathcal{C}_i$ as $\lambda_1^{Q_i}, \ldots, \lambda_{\mathcal{N}_i}^{Q_i}$, and all absolute values of eigenvalues must be less than 1 to satisfy the stability of the sub-quantizer Q_i. I can formulate the above as the optimization problem.

[Problem 4.1]

Suppose that $G, T_{\mathrm{s}}, H, d, x_0, r$ are given.

Then, find the parameters $\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i, i \in \mathbb{I}$ of Q_i and $w_i, i \in \mathbb{I} \setminus \{N\}$ which satisfy

$$\text{(C1)} \quad \max_{j \in \{1,\ldots,\mathcal{N}_i\}} \left|\lambda_j^{Q_i}\right| < 1 \quad i \in \mathbb{I},$$

and minimize the evaluation function

$$J(Q(Q_1(\mathcal{A}_1, \mathcal{B}_1, \mathcal{C}_1), \ldots, Q_N(\mathcal{A}_N, \mathcal{B}_N, \mathcal{C}_N), w_1, \ldots, w_{N-1})),$$

in Equation (4.9).

The evaluation function J calculates the gap between z and z_{ref}. If the evaluation function J is minimized, the output of the discrete-valued input system with the optimal switching-type quantizer will be similar to that of the ideal system.

4.2 Model-tuning approach to design a dynamic quantizer

4.2.1 Dynamic quantizer for a discrete-time linear system

In this section, I design the switching-type quantizer Q for a given nonlinear system G. In previous research on the design of dynamic quantizers [54], it was shown that, for a discrete-time linear system

$$P_{\mathrm{d}i} : \begin{cases} x_P[k+1] = A_{\mathrm{d}} x_P[k] + B_{\mathrm{d}} v[k], \\ \quad z_P[k] = C_{\mathrm{d}} x_P[k], \end{cases} \tag{4.10}$$

the optimal quantizer $Q^\star$ is given by

$$Q^\star : \begin{cases} \mathcal{N} = n_P, \\ \mathcal{A} = A_{\mathrm{d}}, \\ \mathcal{B} = B_{\mathrm{d}}, \\ \mathcal{C} = -(C_{\mathrm{d}} B_{\mathrm{d}})^{-1} C_{\mathrm{d}} A_{\mathrm{d}}. \end{cases} \tag{4.11}$$

Here, the parameter n_P is the dimension of the state x_P of P_{d}. The optimal quantizer $Q^\star$ minimizes the gap between the outputs of the discrete-valued input system with $Q_i^\star$ and the continuous-valued input system in terms of the ∞-norm. If I apply this method to **[Problem 4.1]**, I must linearize and discretize the continuous-time nonlinear system G to obtain the discrete-time linear model. However, the model obtained by linearization and discretization is not optimal for the dynamic quantizer and affect the performance of the quantizer.

4.2.2 Problem formulation of model-tuning approach to quantizer design

In this section, I propose a design method of the switching-type dynamic quantizer Q for a nonlinear system G based on Equation (4.11). The key idea is to attribute the design problem of the switching-type dynamic quantizer Q to the search problem of design models, which are continuous-time and linear models. Specifically, the design procedure $\mathcal{M}(\cdot)$, in which Q_i is designed based on the design model P_i, is expressed as

$$
\begin{aligned}
&Q_i = \mathcal{M}(P_i(A_i, B_i, C_i)) \\
&Q_i : \begin{cases} \mathcal{N}_i = n_{Pi}, \\ \mathcal{A}_i = e^{T_\mathrm{s} A_i}, \\ \mathcal{B}_i = \displaystyle\int_0^{T_\mathrm{s}} e^{T_\mathrm{s} A_i}\, \mathrm{d}t B_i \\ \mathcal{C}_i = -\left(C_i \displaystyle\int_0^{T_\mathrm{s}} e^{T_\mathrm{s} A_i}\, \mathrm{d}t B_i \right)^{-1} C_i e^{T_\mathrm{s} A_i}. \end{cases}
\end{aligned}
\tag{4.12}
$$

I explain the derivation of the conversion equation, which is Equation (4.12), from the design model P_i to the sub-quantizer Q_i below.

First, I consider N design models P_i:

$$
P_i \ : \begin{cases} \dot{x}_{Pi}(t) = A_i x_{Pi}(t) + B_i v(t), \\ z_{Pi}(t) = C_i x_{Pi}(t), \end{cases} \tag{4.13}
$$

where $x_{Pi} \in \mathbb{R}^{n_{Pi}}$ is the state, and $A_i \in \mathbb{R}^{n_{Pi} \times n_{Pi}}$, $B_i \in \mathbb{R}^{n_{Pi} \times m}$, $C_i \in \mathbb{R}^{l \times n_{Pi}}$ are constant matrices.

Then, I derive discrete-time design models $P_{\mathrm{d}i}$ in Equation (4.10) by discretizing the continuous-time design models P_i in Equation (4.13). I apply a mapping $\mathcal{D} : P_i \mapsto P_{\mathrm{d}i}$ such as the zero-order holder expressed by

$$
\mathcal{D} \ : \begin{cases} A_{\mathrm{d}i} = e^{T_\mathrm{s} A_i}, \\ B_{\mathrm{d}i} = \displaystyle\int_0^{T_\mathrm{s}} e^{T_\mathrm{s} A_i}\, \mathrm{d}t B_i, \\ C_{\mathrm{d}i} = C_i. \end{cases} \tag{4.14}
$$

Finally, I substitute Equation (4.14) into Equation (4.11) to obtain the design procedure (4.12). I search for the design model P_i to design the sub-quantizer Q_i based on the design procedure (4.12).

In summary, if there are the design procedure for the design models P_i and the evaluation function expressed as Equation (4.9), I can select the design models and design the switching-type dynamic quantizer Q.

Based on this procedure, **[Problem 4.1]** can be rewritten as **[Problem 4.2]**.

[Problem 4.2]

Suppose that $G, T_\mathrm{s}, H, d, x_0, r$ are given.

Then, find the parameters $A_i, B_i, C_i, i \in \mathbb{I}$ of $Q_i = \mathcal{M}(P_i(A_i, B_i, C_i))$ and $w_i, i \in \mathbb{I} \setminus \{N\}$ which satisfy

$$\text{(C1)} \quad \max_{j \in \{1,\ldots,\mathcal{N}_i\}} \left|\lambda_j^{P_i}\right| < 1 \quad i \in \mathbb{I},$$

and minimize

$$J(Q(\mathcal{M}(P_i(A_1, B_1, C_1)), \ldots, \mathcal{M}(P_i(A_N, B_N, C_N)), w_1, \ldots, w_{N-1})),$$

Note that $\lambda_1^{P_i}, \ldots, \lambda_{\mathcal{N}_i}^{P_i}$ are the eigenvalues of the matrix

$$e^{T_\mathrm{s} A_i} - \int_0^{T_\mathrm{s}} e^{T_\mathrm{s} A_i} \,\mathrm{d}t B_i \left(C_i \int_0^{T_\mathrm{s}} e^{T_\mathrm{s} A_i} \,\mathrm{d}t B_i \right)^{-1} C_i e^{T_\mathrm{s} A_i}.$$

These eigenvalues corresponds to 0 and zeros of the discrete-time linear system $P_{\mathrm{d}i}$ in Equation (4.10) and the poles of the system $\mathcal{L}_i$ in Equation (4.4).

The advantages and characteristics of my proposed method are as follows.

- Although direct tuning of the parameters of the sub-quantizers Q_i is possible, the design model tuning approach allows us to discuss nonlinear system G as linear systems P_i.
- I can fix several linear design models if there are reasonable models obtained from the plant and explore the remaining design models. For example, if some sub-quantizers are used around the equilibrium points of the nonlinear system G, I can fix the counterpart design models as the linear approximate models around the equilibrium points.
- I can target a broader class of linear models that are not linear approximations of a given nonlinear system. For example, I can accept the cases $n_{Pi} < n$ and $n_{Pi} > n$.
- This approach can fill the gap between nonlinear and linear, as well as between continuous time and discrete time with linear models.
- My proposed method generalizeds the target system and optimization problems fo the previous works [75, 76]

I also define **[Problem 4.3]**, in which the switching conditions w_i is fixed.

[Problem 4.3]

Suppose that $G, T_\mathrm{s}, H, d, x_0, r, w_i i \in \mathbb{I} \setminus \{N\}$ are given.

Then, find the parameters $A_i, B_i, C_i, i \in \mathbb{I}$ of $Q_i = \mathcal{M}(P_i(A_i, B_i, C_i))$ which satisfy

$$\text{(C1)} \quad \max_{j \in \{1,\ldots,\mathcal{N}_i\}} \left|\lambda_j^{P_i}\right| < 1 \quad i \in \mathbb{I},$$

and minimize

$$J(Q(\mathcal{M}(P_i(A_1, B_1, C_1)), \ldots, \mathcal{M}(P_i(A_N, B_N, C_N)))),$$

I verify the effectiveness of my proposed method through a control system for the swing-up and stabilization of a cart-type inverted pendulum depicted as Fig. 4.4.
[**Remark**] **[Problem 4.1]** represents the problem of designing a discrete-time switching-type dynamic quantizer for a continuous-time nonlinear system. **[Problem 4.2]** is an optimization problem incorporating my proposed model-tuning method into **[Problem 4.1]**. **[Problem 4.3]** is a derivative of **[Problem 4.2]** and is the optimization problem, where switching conditions are fixed. Note that **[Problem 4.3]** is a relaxation problem of **[Problem 4.2]**; I focus on the equivalence of **[Problem 4.1]** and **[Problem 4.2]** in Subsection 4.4.

4.3 Illustrative Example

4.3.1 Swing-up and stabilization control of inverted pendulum

I consider the cart-type inverted pendulum system such as Fig. 4.4 to evaluate the proposed method. The system G consists of the inverted pendulum S and the controller K including the stabilization controller K_1 and the swing-up controller K_2, and the switching-type quantizer Q is composed of Q_1 and Q_2. The inverted pendulum is shown in Fig. 4.5, where x_1 is the displacement of the cart, x_2 is the angle of the pendulum, m_{c} is the mass of the cart, m_{p} is the mass of the pendulum, v is the force on the cart, μ_B is the friction coefficient between the cart and the floor, and μ_C is the friction coefficient between the cart and the pendulum. I define the state vector by $x = [x_1, x_2, x_3, x_4]^\top = [x_1, x_2, \dot{x}_1, \dot{x}_2]^\top$. Then, the nonlinear state equations of the inverted pendulum are

$$S : \begin{cases} \dot{x} = f(x, v) = \begin{bmatrix} x_3 \\ x_4 \\ f_1(x) \\ f_2(x) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g_1(x) \\ g_2(x) \end{bmatrix} v, \\ \sigma = \phi(x_2), \\ z = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x, \end{cases} \tag{4.15}$$

with

$$\begin{aligned}
f_1(x) &= \frac{1}{6}m_{\mathrm{p}}^2 l^3 {x_4}^2 \sin x_2 - \frac{1}{4}m_{\mathrm{p}}^2 l^2 g \sin x_2 \cos x_2 \\
&\quad - \frac{1}{3}m_{\mathrm{p}} l^2 \mu_B x_3 + \frac{1}{2}m_{\mathrm{p}} l \mu_C x_4 \cos x_2 / D(x_2), \\
f_2(x) &= -\frac{1}{4}m_{\mathrm{p}}^2 l^2 {x_4}^2 \sin x_2 \cos x_2 + \frac{1}{2}m_{\mathrm{p}}(m_{\mathrm{p}} + m_{\mathrm{c}}) l g \sin x_2 \\
&\quad + \frac{1}{2}m_{\mathrm{p}} l \mu_B x_3 \cos x_2 - (m_{\mathrm{p}} + m_{\mathrm{c}})\mu_C x_4 / D(x_2), \\
g_1(x) &= \frac{1}{3}m_{\mathrm{p}} l^2 / D(x_2), \\
g_2(x) &= -\frac{1}{2}m_{\mathrm{p}} l \cos x_2 / D(x_2), \\
D(x_2) &= \frac{1}{3}m_{\mathrm{p}}(m_{\mathrm{p}} + m_{\mathrm{c}}) l^2 - \frac{1}{4}m_{\mathrm{p}}^2 l^2 \cos^2 x_2, \\
\phi(x) &= (x + \pi) + 2\pi \mathrm{sgn}(x + \pi)\left(\frac{1}{2} + \left\lfloor \frac{|x + \pi|}{2\pi} \right\rfloor\right).
\end{aligned} \tag{4.16}$$

Note that the set $\mathbb{S} \in \mathbb{R}^s$, in which the output for switching is, the closed interval $[-\pi, \pi]$ and the function ϕ is to round the angle x_2 into the set $\mathbb{S}$. By Taylor expansion around the origin $x = [0, 0, 0, 0]^\top$ of the function $f(x, v)$, I obtain a linear approximation model $\bar{S}_0$ of S:

$$\begin{aligned}
&\bar{A}_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}, \ \bar{B}_0 = \begin{bmatrix} 0 \\ 0 \\ b_{13} \\ b_{14} \end{bmatrix}, \bar{C}_0 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, \\
&a_{32} = -\frac{{m_{\mathrm{p}}}^2 l^2 g}{4D(0)}, \quad a_{33} = -\frac{(4I + m_{\mathrm{p}} l^2)\mu_B}{4D(0)}, \ a_{34} = \frac{m_{\mathrm{p}} l \mu_C}{2D(0)}, \\
&a_{42} = \frac{m_{\mathrm{p}} l (m_{\mathrm{p}} + m_{\mathrm{c}}) g}{2D(0)}, \ a_{43} = \frac{m_{\mathrm{p}} l \mu_B}{2D(0)}, \quad a_{44} = -\frac{(m_{\mathrm{p}} + m_{\mathrm{c}})\mu_C}{D(0)}, \\
&b_{13} = \frac{(4I + m_{\mathrm{p}} l^2)}{4D(0)}, \quad b_{14} = -\frac{m_{\mathrm{p}} l}{2D(0)}.
\end{aligned} \tag{4.17}$$

The switching-type controller K is defined as

$$K \ : \ u(t) = \begin{cases} K_1(x(t)) & \text{if } |\sigma| - \theta_K > 0, \\ K_2(x(t)) & \text{otherwise,} \end{cases} \tag{4.18}$$

where $\theta_K = \pi/6$ is a threshold value for the switching condition. I design the state

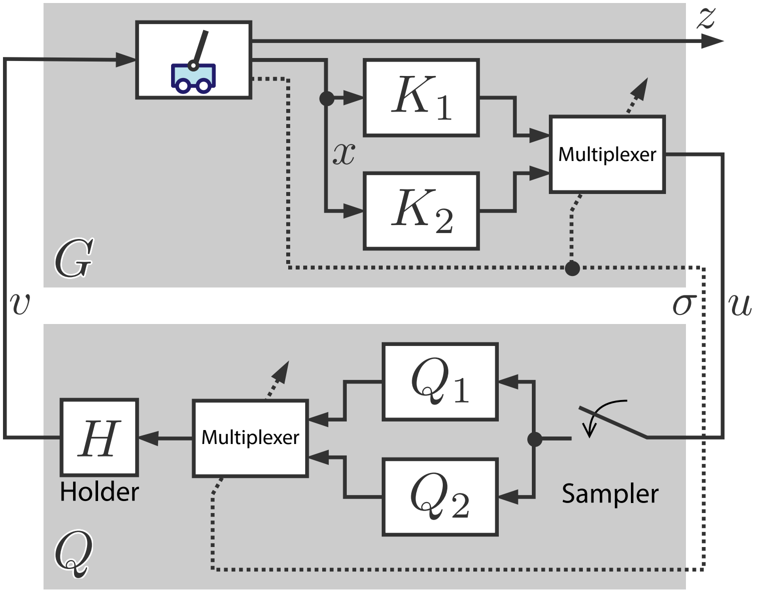


Fig. 4.4: The swing-up and stabilization control system for cart-type inverted pendulum.

feedback controller K_1 and the energy method controller K_2 as

$$K_1(x) = Fx, \tag{4.19}$$

$$\begin{aligned} K_2(x) =& \frac{1}{4l} - 2m_\mathrm{p} l^2 x_4 \sin x_2 + 3m_\mathrm{p} lg \sin x_2 \cos x_2 + 4\mu_B l x_3 \\ & - 6\mu_C x_4 \cos x_2 - k_\mathrm{e} l x_4 \cos x_2 \{4(m_\mathrm{p} + m_\mathrm{c}) - 3m_\mathrm{p} \cos^2 x_2\}. \end{aligned} \tag{4.20}$$

The linear state feedback gain F is determined to stabilize the linear approximated model $\bar{S}_0$ around the origin in Equation (4.17). Note that the controller K_2 is based on an energy method[77], and $k_\mathrm{e} > 0$ is an energy control gain.

Table 4.1 summarizes the parameters used in this numerical example.

4.3.2 Design of dynamic quantizer Q

In this simulation, I are considering two-stage control, with a swing-up phase and a stabilization phase. Therefore, I also switch the two dynamic quantizers Q_1 and Q_2 in two steps, as same as the controllers. The quantizer $Q_1 = \mathcal{M}(P_1)$ is for the stabilization phase, and the quantizer $Q_2 = \mathcal{M}(P_2)$ is for the swing-up phase in Fig. 4.4. I define the switching conditions of multiplexer of the quantizer Q as

$$w_1(\sigma) = |\sigma| - \theta_Q, \tag{4.21}$$

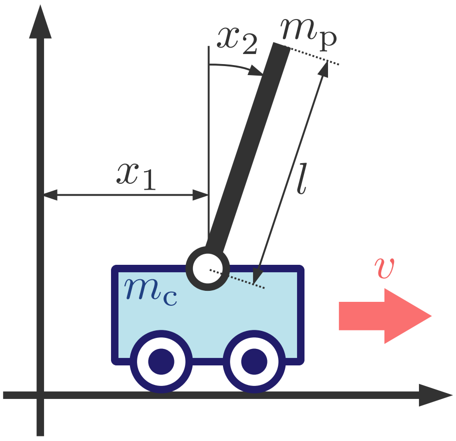


Fig. 4.5: Cart-type inverted pendulum.

and set the switching angle $\theta_Q = \pi/6$, which is the same as the threshold of the controller θ_K in Equation (4.18). Under this condition, I solve the **[Problem 4.3]**.

First, the sub-quantizer Q_1 is the quantizer for the dynamics of the cart-type inverted pendulum around the origin. Thus, I set the design model $P_1^\star$ as $P_1^\star = \bar{S}_0$ in Equation (4.17). Then, Q_1 is given by

$$\begin{aligned}
&\mathcal{N}_1 = 4,\\
&\mathcal{A}_1 = \begin{bmatrix} 1.0 & -0.00416 & 0.0194 & 0.0 \\ 0.0 & 1.05 & 0.00424 & 0.0203 \\ 0.0 & -0.415 & 0.944 & -0.00411 \\ 0.0 & 4.60 & 0.423 & 1.05 \end{bmatrix},\\
&\mathcal{B}_1 = \begin{bmatrix} 0.001 \\ -0.00424 \\ 0.0561 \\ -0.423 \end{bmatrix},\\
&\mathcal{C}_1 = \begin{bmatrix} 0.0 & 123 & 0.500 & 2.40 \end{bmatrix},
\end{aligned} \tag{4.22}$$

from the model $P_{\mathrm{c}1}^\star = \bar{S}_0$ and Equation (4.12). Note that I multiply the matrix $\mathcal{C}_1$ by the coefficient $\gamma = 0.5$ to prevent the inputs from being too large. For the quantizer Q_1 that works for the stabilization, it is reasonable to use the linear approximation model $\bar{S}_0$ near the origin, which is fixed to reduce the design parameters.

Next, I design Q_2 by finding the linear model P_2. I assume a fourth-order control-

Table 4.1: Parameters of the optimization, controller and dynamic quantizer.

Parameter		Values
m_{c} :	Mass of the cart	0.1 [kg]
m_{p} :	Mass of the pendulum	1.0 [kg]
l :	Length of the pendulum	0.2 [m]
g :	Acceleration of gravity	9.80665 [m/s^2]
I :	Moment of inertia	$m_{\mathrm{p}}l^2/12$ [kg m^2]
μ_B :	Friction coefficient (cart and floor)	0.1 [kg/s]
μ_C :	Friction coefficient (cart and pendulum)	0.0001 [kg m^2/s]
F :	State feedback gain	$[0.385, 13.4, 1.51, 0.627]$
k_{e} :	Energy control gain	0.2 [m/s]
d :	Quantization width	5.0 [N]
T_{s} :	Sampling period	0.02 [s]
x_0 :	Initial state of x	$[0, (4/3)\pi, 0, 0]^\top$
T :	Simulation time	10 [s]

lable canonical-form model P_2 expressed as

$$\begin{aligned}
n_{P2} &= 4, \\
A_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix}, \\
B_2 &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \\
C_2 &= \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix}.
\end{aligned} \tag{4.23}$$

I set the design parameters $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$. I use the evaluation function in Equation (4.9) and determine the parameters of the design model that minimizes the evaluation function.

Using Particle Swarm Optimization (PSO) [78], I found the optimal design model

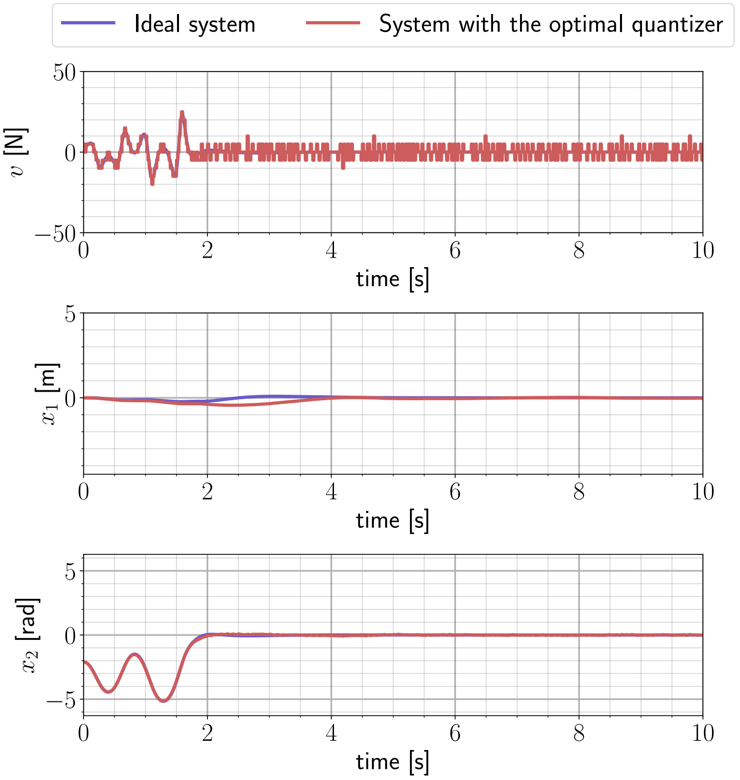


Fig. 4.6: The input v and states x_1, x_2 of the system with the optimal quantizers $Q_1 = \mathcal{M}(P_{\rm c1}^\star)$ and $Q_2 = \mathcal{M}(P_{\rm c2}^\star)$, compared to those of the ideal system.

$P_{\rm c2}^\star$ to be

$$\begin{aligned} A_{\rm c2}^\star &= \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 7.64 & -17.6 & -25.2 & -36.8 \end{bmatrix}, \\ B_{\rm c2}^\star &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \\ C_{\rm c2}^\star &= \begin{bmatrix} 2.67 & 15.4 & 8.17 & 6.17 \end{bmatrix}. \end{aligned} \tag{4.24}$$

Note that the number of particles and iterations are both set to 100, all the initial design parameters are set by random numbers following a uniform distribution from -10 to 10, and the simulation is implemented by Julia 1.10.4. Therefore, the dynamic

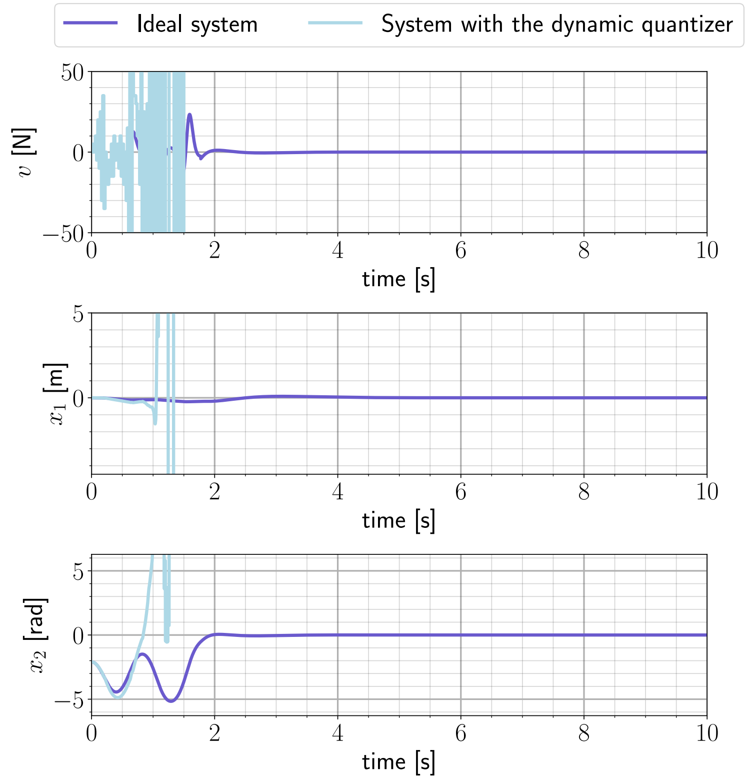


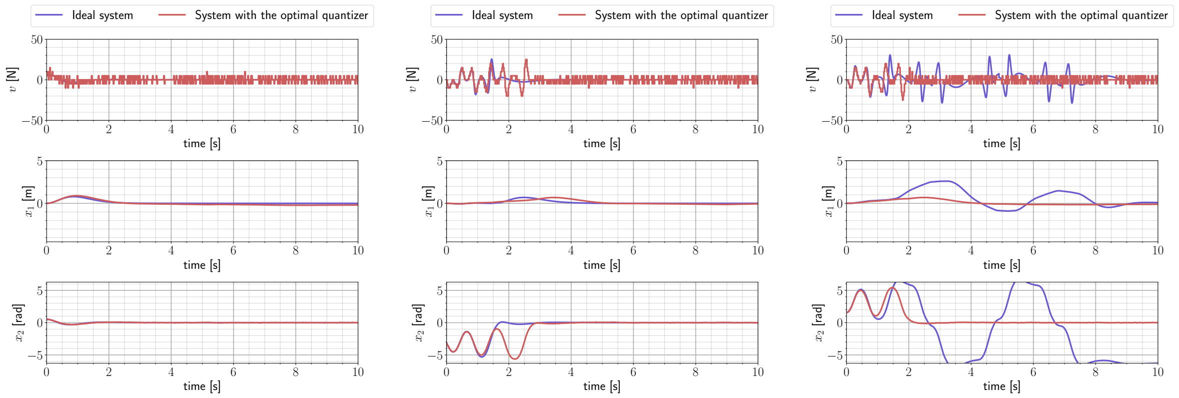
Fig. 4.7: The input v and states x_1, x_2 of the system with the quantizers $Q_1 = \bar{S}_\pi$ and $Q_2 = \mathcal{M}(P^\star_{\mathrm{c}2})$, compared to those of the ideal system.

quantizer Q_2 is

$$\begin{aligned}
\mathcal{N}_2 &= 4 \\
\mathcal{A}_2 &= \begin{bmatrix} 1.0 & 0.020 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.020 & 0.0 \\ 0.0012 & -0.0028 & 0.996 & 0.014 \\ 0.108 & -0.248 & -0.359 & 0.476 \end{bmatrix}, \\
\mathcal{B}_2 &= \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.014 \end{bmatrix}, \\
\mathcal{C}_2 &= \begin{bmatrix} -37.8 & -156.8 & -70.5 & -34.5 \end{bmatrix},
\end{aligned} \tag{4.25}$$

from Equation (4.12).

The simulation results are shown in Fig. 4.6. Figure 4.6 displays the inputs and



(a) $x_0 = [0, \pi/6, 0, -0.2]^\top$ (b) $x_0 = [0, \pi, 0, -10]^\top$ (c) $x_0 = [0, \pi/2, 0.1, -0.1]^\top$

Fig. 4.8: In case initial condition changes.

states of the ideal system and the discrete-valued input system, in which the coefficient matrices of the sub-quantizers Q_1, Q_2 are denoted as (4.22) and (4.25). In the figure, the red curve indicates the time response of the discrete-valued input system, and the blue one indicates that of the ideal system. I can see that the angle of the pendulum with the designed dynamic quantizer is closer to that of the ideal system from Fig. 4.6.

For comparison, I change the linear filter P_2 for the sub-quantizer Q_2. I use the linear approximation model $\bar{S}_\pi$ which is obtained by Taylor expansion around the equilibrium point $x = [0, \pi, 0, 0]^\top$ and expressed as

$$\begin{aligned}
&\bar{A}_\pi = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}, \ \bar{B}_\pi = \begin{bmatrix} 0 \\ 0 \\ b_{13} \\ b_{14} \end{bmatrix}, \bar{C}_\pi = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, \\
&a_{32} = -\frac{{m_\mathrm{p}}^2 l^2 g}{4D(\pi)}, \qquad a_{33} = -\frac{(4I + m_\mathrm{p} l^2)\mu_B}{4D(\pi)}, \ a_{34} = -\frac{m_\mathrm{p} l \mu_C}{2D(\pi)}, \\
&a_{42} = -\frac{m_\mathrm{p} l (m_\mathrm{p} + m_\mathrm{c}) g}{2D(\pi)}, \ a_{43} = -\frac{m_\mathrm{p} l \mu_B}{2D(\pi)}, \qquad a_{44} = -\frac{(m_\mathrm{p} + m_\mathrm{c})\mu_C}{D(\pi)}, \\
&b_{13} = \frac{(4I + m_\mathrm{p} l^2)}{4D(\pi)}, \qquad b_{14} = \frac{m_\mathrm{p} l}{2D(\pi)},
\end{aligned} \tag{4.26}$$

for the sub-quantizer Q_2. The simulation result is displayed in Fig. 4.7. Fig. 4.7 shows that the controller K and the quantizer $Q_2 = \mathcal{M}(\bar{S}_\pi)$ cannot achieve the swing-up and stabilization of the inverted pendulum. From the result, I find that the proposed method can design the satisfactory dynamic quantizer.

4.4 Discussion of the model-tuning approach

4.4.1 Results of the simulation on various initial conditions

The Equation (4.24) is designed by minimizing the evaluation function in Equation (4.9), which depends on initial values. Thus I should check whether it can be stabilized with other initial values. I show the results of three different initial conditions $x_0 = [0, \pi/6, 0, -0.2]^\top, [0, \pi, 0, -10]^\top, [0, \pi/2, 0.1, 0]^\top$ in Fig. 4.8. Note that the dynamic quantizers in Fig. 4.8 is the same as the dynamic quantizer in Fig. 4.6. The figures show that the system with the dynamic quantizer designed from the design model can be stabilized for three initial values. In Fig. 4.8(a) and Fig. 4.8(b), the red curve and blue curve are almost the same, and the system can be stabilized. However, in Fig. 4.8(c), the red and blue curves are different, but the states of the system with the dynamic quantizer Q converge more quickly than the ideal system. This indicates that the value of the evaluation function J increases but the dynamic quantizer Q contributes the stability of the system. Moreover, the dynamic quantizer possibly extends the range of initial values where I can swing up and stabilize.

4.4.2 Results of the simulation with different design models

Then, I show the case in which the dimension of the dynamic quantizer Q_2 is less than that of the system. In this simulation, I fixed the design model P_2 as a second-order controllable canonical form and optimized it. I obtained the design model P_2 with the reduced order as

$$\begin{aligned} n_{P2} &= 2, \\ A^\star_{c2} &= \begin{bmatrix} 0.0 & 1.0 \\ -14.8 & -38.3 \end{bmatrix}, \\ B^\star_{c2} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ C^\star_{c2} &= \begin{bmatrix} 12.4 & 6.81 \end{bmatrix}. \end{aligned} \tag{4.27}$$

The input and states of the discrete-valued input system with the dynamic quantizer Q_2, which is designed by the model P_2, is shown the red curve in Fig. 4.9. The result shows that a satisfactory second-order quantizer can be designed. I stress that my approach is not only a method for designing fixed-order quantizers but also allows us

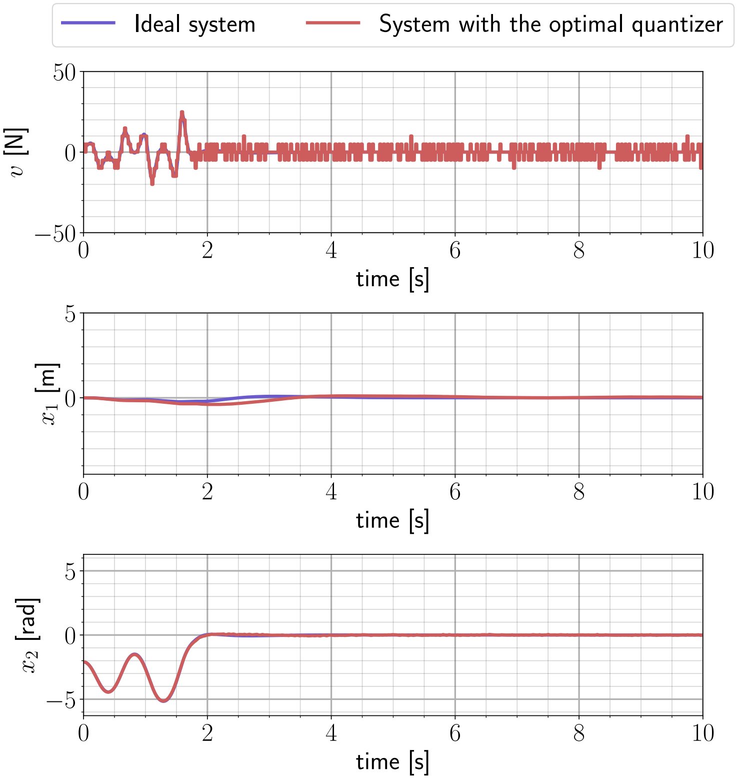


Fig. 4.9: The input v and states x_1, x_2 of the system with the fourth-order quantizer $Q_1 = \mathcal{M}(P_{\mathrm{c1}}^{\star})$ and the second-order quantizer $Q_2 = \mathcal{M}(P_{\mathrm{c2}}^{\star})$, compared to those of the ideal system.

to consider what type of linear model should be used.

In addition, I consider the case that not only the design model but also switching conditions for the dynamic quantizer are optimization parameters, which is formulated as **[Problem 4.2]**. The switching conditions for the multiplexer is

$$w_1(\sigma) = |\sigma| - \theta_Q, \tag{4.28}$$

and optimize the parameter θ_Q. Figure 4.10 shows the simulation results when the switching conditions are also optimized. In Fig. 4.10, I can swing up and stabilize the pendulum like Fig. 4.6. In this case, the optimal design model is expressed as Equation (4.29) and the switching angle is 2.37[rad] and this value is bigger than

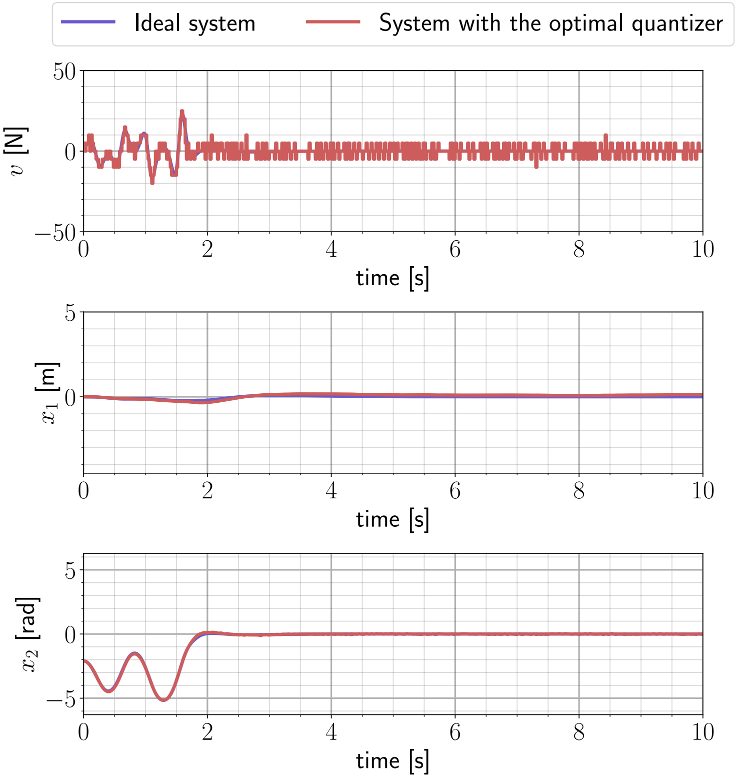


Fig. 4.10: Switching-condition is one of the optimization parameters.

$\pi/6$[rad], which is the controller switching condition.

$$\begin{aligned} n_{P2} &= 4, \\ A^{\star}_{\mathrm{c}2} &= \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 23.8 & -1.22 & 2.75 & -31.4 \end{bmatrix}, \\ B^{\star}_{\mathrm{c}2} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \\ C^{\star}_{\mathrm{c}2} &= \begin{bmatrix} 13.9 & 29.9 & 30.4 & 12.5 \end{bmatrix}. \end{aligned} \tag{4.29}$$

Table 4.2: Comparison of the efficiency in solving **[Problem 4.1]** and **[Problem 4.2]**.

Method	Statistic values of Equation (4.9)			
	Best	Worst	Mean	St. dev.
[Problem 4.1]	0.318	56.6	9.32	13.3
[Problem 4.2]	0.212	3.24	1.09	1.26

4.4.3 Comparison of the proposed method and the direct search of the parameters of the dynamic quantizer

Finally, I compare the efficiency in solving **[Problem 4.1]** and **[Problem 4.2]**. **[Problem 4.1]** and **[Problem 4.2]** are solved by a direct search method and my proposed method, respectively. The direct search method is a method to search for the parameters of the sub-quantizer Q_2. Specifically, I search for the parameters of the coefficient matrices $\mathcal{A}_2, \mathcal{B}_2, \mathcal{C}_2$ of the sub-quantizer Q_2 and the switching threshold θ_Q of the sub-quantizers, and the number of the design parameters are same as that in case of the proposed method. The statistical values of 20 trials for the optimization solved by each method are shown in Table 4.2. Table 4.2 shows that **[Problem 4.2]** has a smaller evaluation function value, indicating that the search is more efficient. One of the reasons for this is the following consideration. In **[Problem 4.1]**, it is necessary to search the vast parameter space of $\mathbb{R}^8$. However, **[Problem 4.2]** may limits the parameter space by the mapping in Equation (4.12), and the efficient search can be performed.

4.5 Summary

This study addressed the design problem of the sampled-data and switching-type dynamic quantizers for continuous-time nonlinear systems. In the proposed approach, N linear design models were designed to minimize the value of the evaluation function and make the performance of the dynamic quantizer better. A numerical example using a cart-type inverted pendulum was used to verify the effectiveness of the proposed method.

There are several future research directions. First, although I dealt with numerical optimization in this chapter, I should find an analytical method. Especially, I should investigate the relationship between the nonlinear system and the optimal linear design models. Second, I hope to use this method on more practical problems with complex systems in the future. In addition, it would also be an intriguing approach to replace the controller with one based on model predictive control or reinforcement learning[79, 80, 81] and then design the dynamic quantizer. I plan to explore this direction in future work.

Chapter 5

Model-tuning approach to quantization process in TDA-based segmentation system

In this chapter, I apply my proposed method to the system composed of TDA and the quantization process. The whole system is the segmentation process for gray-scale images, Σ_1 is the TDA process for binary image segmentation, and Σ_2 is the binarization process Q, which transforms a gray-scale image to a binary image. I fix the design procedure $\mathcal{M}$ of the binarization process and design the binarization process Σ_2 by tuning the binarization algorithm P.

TDA is a method to analyze topological structures in a dataset. For example, it can evaluate the robustness of the topological feature persistency and topological invariants, such as the presence or absence of holes. Recently, TDA has been used in various fields [82, 83, 84, 85], and applications include the analysis of molecular structures of materials [82] and the analysis of tree structures of gene evolution [84].

TDA can analyze topological structures; I consider it possible to apply it to image processing, such as image segmentation. Segmentation is the process of dividing an image into meaningful regions, and it is used in various fields such as medical care and autonomous driving [86]. For example, by associating the size of connected components (clusters of pixels) in an image with the size of holes in TDA, the area occupied by objects in the image may be extracted. Moreover, since TDA can distinguish differences in topological structures (hole sizes), it can likely separate signals such as high-spatial-frequency noise from low-spatial-frequency textures. In this chapter, I propose a segmentation method for gray-scale images, which combines quantization processing and TDA for binary images. In this method, there is a degree of freedom in the choice of the quantization process. Therefore, it is essential to select an appropriate quantization method so that the information necessary for segmentation can be obtained with TDA.

As a quantization method, thresholding is considered, but this method usually requires careful threshold adjustment. As an automatic adjustment of the threshold from image information, Otsu's method (discriminant analysis) [87] is one of the methods, but even if the threshold is automatically determined, it is not possible to preserve the gray-scale information of the image. Thus, I adopt halftoning as a quantization method and, in this chapter, specifically use the random dithering method.

The halftone process generates a binary image that pseudo-represents the gray-scale information by varying the density of white and black pixels. This process has the advantage of preserving the appearance of the gray-scale image as much as possible without the need for threshold adjustment. In general, halftoning adds high spatial frequency noise. Thus, it is difficult to extract objects from a halftone image without preprocessing, such as noise reduction. However, by using TDA to extract objects as topological structures, it is expected that segmentation of objects buried in noise will be possible.

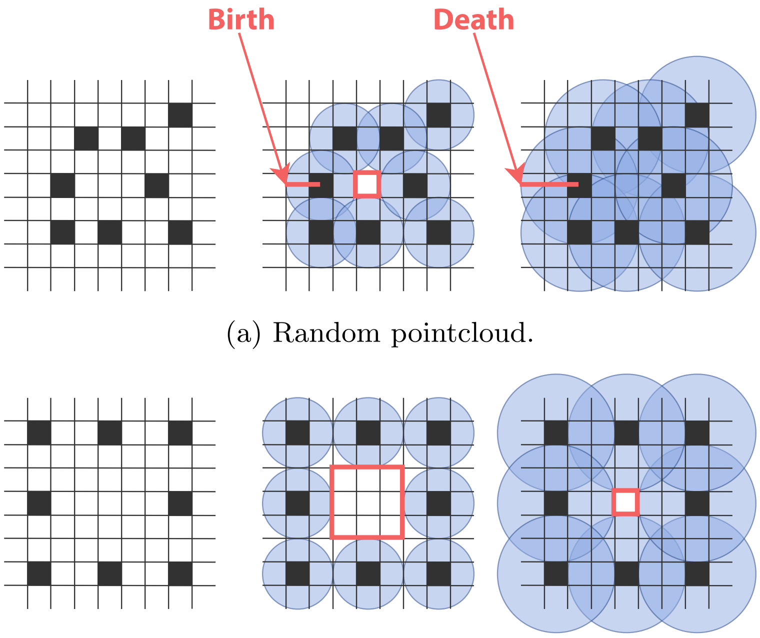
In this chapter, I first apply TDA to halftone images generated by the random dithering method and binary images generated by Otsu's method. I visualize the topological structure of the images and perform segmentation using the information obtained. I then provide examples to explain which of Otsu's and the random dithering methods is more appropriate as a quantization method. Next, I compare the results of applying TDA to blurred images, other standard images, and CT images using random dithering and Otsu's methods. Furthermore, I discuss the challenges of further developing the proposed method by combining the error diffusion method, another halftoning method, with TDA for segmentation.

Finally, I supplement the relationship with previous studies. The previous studies[88, 89] proposes a gray-scale image segmentation method that combines CNNs and TDA. This method reshapes the gray-scale image with a convolutional neural network, then applies thresholding to perform segmentation, and the role of TDA is to generate features used in the learning process of the NN. The proposed method in this chapter differs from the approach in [88, 89] in that it converts gray-scale images to halftone images and directly uses TDA without learning.

5.1 Topological data analysis

Here, I explain the concept of topological data analysis (TDA), which is the basis of this chapter. Then, I briefly describe the TDA algorithm for binary images used for segmentation.

5.1.1 Basics of TDA



(a) Random pointcloud.

(b) Ring pointcloud.

Fig. 5.1: Birth and death by increasing the radii of the circles in each pointcloud.

The basic idea of TDA[90, 91, 92, 93, 94, 95, 96, 97] is to infer the structure of pointcloud, image and graph data by measuring the gap between the radii of the birth and death of holes. For example, given the set of points shown on the left in Fig. 5.1(a), a circle is generated from each point, increasing its radius in TDA. The birth radius is defined as the radius at which the circles intersect, forming a hole, whereas the death radius is defined as the radius at which the hole collapses. When the radii of birth and death of holes are defined in this way, the holes can be investigated by measuring the gap between the radii of birth and death. This gap is called persistence, and it can be used to examine the presence or absence of holes and their size. The gap is larger for the ring pointcloud in Fig. 5.1(b) than for the random pointcloud in Fig. 5.1(a).

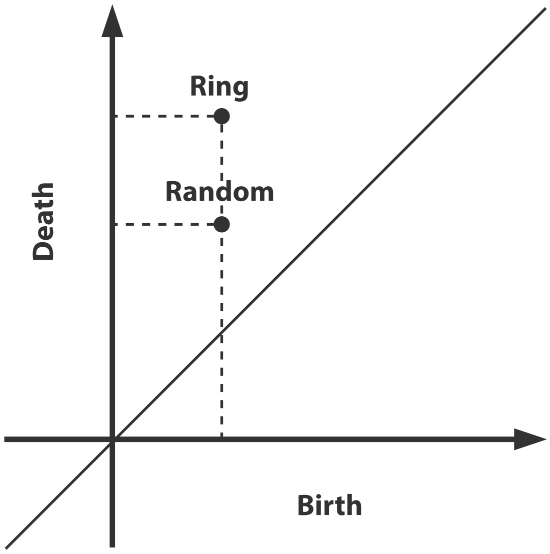


Fig. 5.2: Persistent diagram of the random and ring pointclouds, where I can see that the hole structure of the ring pointcloud is larger than that of the random pointcloud from the gap between the birth and death.

The gap between the radii of birth and death is shown as a persistent diagram. The horizontal and vertical axes of this persistent diagram represent the radii of birth and death, respectively. The further away from the diagonal, the larger the holes. For example, the persistent diagram for the random and ring pointclouds in Fig. 5.1(a) and Fig. 5.1(b) is shown in Fig. 5.2. Thus, the point close to the diagonal line is a hole in the random pointcloud, and the point far from the diagonal line is a hole in the ring pointcloud in Fig. 5.2.

5.1.2 TDA for binary images

I explain TDA for binary images using the binary image in Fig. 5.3[98]. The image of Fig. 5.3 consists of white and black pixels, and black areas surround two white holes. The first hole is a white area completely surrounded by black in the upper right, and the second hole is a white area in the lower left, which is not completely surrounded by black.

I can obtain the persistent diagram shown in Fig. 5.4 when I apply TDA to this binary image. Among the two points in Fig. 5.4, the point far from the diagonal line represents the largest hole completely surrounded by black in Fig. 5.3, and the point close to the diagonal line corresponds to the small hole in the lower left that is not

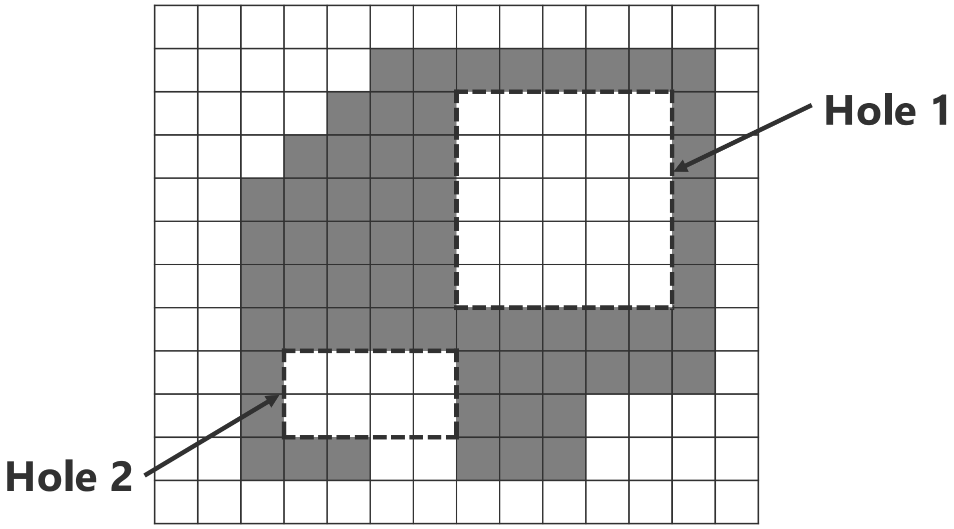


Fig. 5.3: A sample image with two holes.

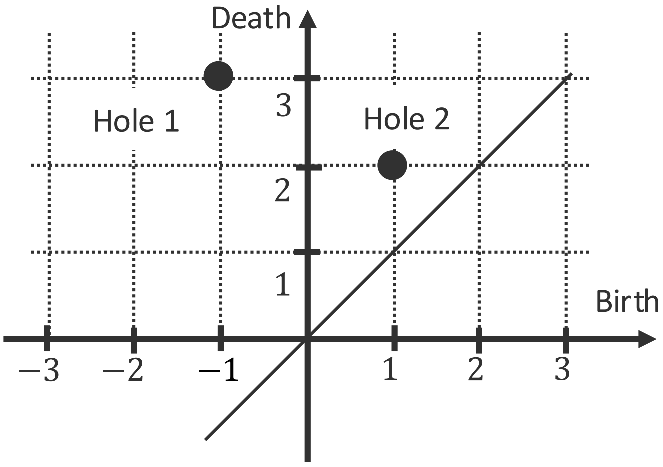


Fig. 5.4: Persistent diagram of the two-hole image (Fig. 5.3) .

completely surrounded by black in Fig. 5.3.

In the case of TDA of a binary image, as in the case of TDA for a point cloud, the hole tends to become larger as the distance from the diagonal of the persistent diagram, i.e., the persistence, becomes larger. Therefore, persistence is also considered helpful in the analysis of binary images.

Note that white or black is taken as the standard depending on the part of images when applying TDA. If black is taken as the standard, TDA is performed as if the black pixels are expanding, and white regions are regarded as holes, such as Hole 1 and Hole 2 in Fig. 5.3.

5.2 Segmentation of binary images using TDA

In this section, I perform segmentation using TDA on binary images obtained by quantizing gray-scale images. To this end, I compare two methods:

- Applying TDA to binary images obtained by Otsu's method, one of the thresholding methods
- Applying TDA to binary images obtained by random dithering, one of the halftoning methods

and confirm what information can be obtained by TDA in each case. I finally show that random dithering is more suitable for segmentation.

The flow of the proposed method is as follows. First, I perform quantization processing on the gray-scale image to obtain a binary image (Fig. 5.5(b), 5.8(a)). In this study, I use random dithering as the quantization process. After this preprocessing, I apply TDA to the binary image and draw the persistent diagram.

Next, I perform inverse analysis from the persistent diagram. Here, inverse analysis refers to the following series of analyses:

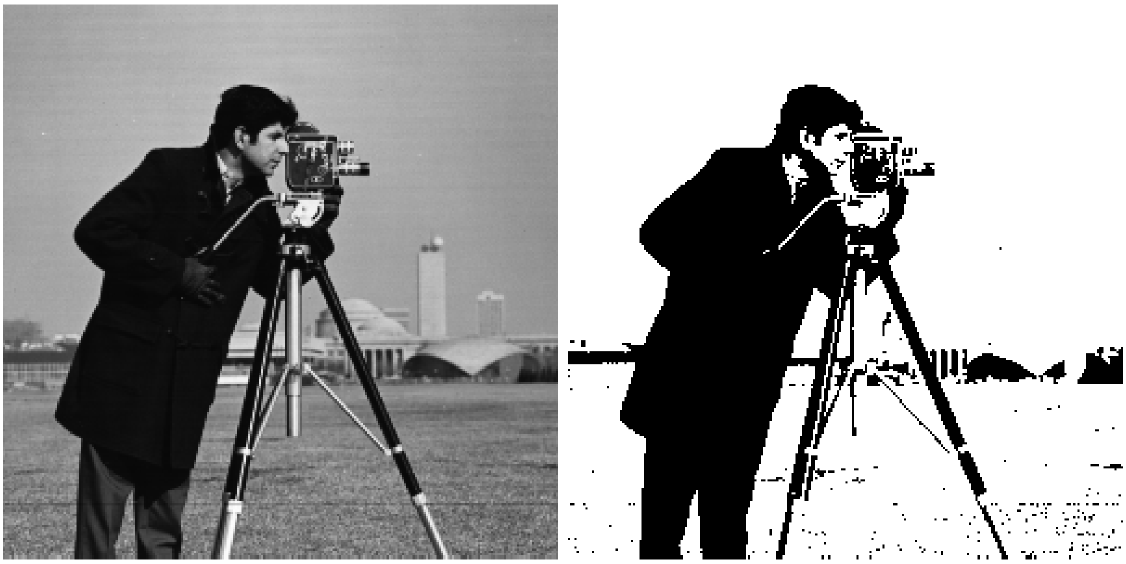
1. Select several points with large persistence, i.e., points far from the diagonal of the persistent diagram, to extract a large region from the image.
2. Extract the holes corresponding to the selected points.
3. Select the hole that fits the part of the image to be segmented from the extracted holes.

Actually, I specify the radii of birth and death and display the holes corresponding to the selected points as a red region. In the inverse analysis, I determine the region with the smallest volume among the candidates corresponding to the holes with the specified radii using optimization calculations[99].

5.2.1 Segmentation of binary images generated by Otsu's method

In this subsection, I apply segmentation to binary images generated by Otsu's method, one of the representative thresholding methods[87] to compare to the proposed method. Otsu's method can automatically determine the threshold and binarize gray-scale images.

In numerical experiments, I apply TDA to the binary image in Fig. 5.5(b) obtained by binarizing the standard image (cameraman) with 256×256 pixels shown in



(a) Grayscale image (original) .

(b) Case of Otsu's method .

Fig. 5.5: Original image and binary image by Otsu's method with the threshold 88.

Fig. 5.5(a) using Otsu's method.

For the binary image in Fig. 5.5(b), when I apply TDA, I obtain the persistent diagram shown in Fig. 5.6. The figure also shows some of the results of the inverse analysis. In the persistent diagram, points near the diagonal line represent small holes, while points far from the diagonal line represent large holes. The results can be interpreted as points near the diagonal line representing are high spatial frequency textures (e.g., noise), and points away from the diagonal line are low spatial frequency textures.

The results can be confirmed in Fig. 5.6, where it can be seen that the high spatial frequency part is concentrated near the diagonal line. In this segmentation, I select the point (10, 38) farthest from the diagonal line with Persistence = 28 to extract a large region with low spatial frequency. The region corresponding to this point is shown in red in Fig. 5.7, corresponding to the cameraman's body part. However, the segmentation of the cameraman's body parts is insufficient.

5.2.2 Segmentation of binary images generated by random dithering

In this subsection, I apply segmentation to halftone images generated by random dithering, one of the representative halftoning methods[100]. In random dithering, the threshold is changed for each pixel when binarizing the gray-scale image. Specifically, an integer value between 0 and 255 is randomly selected as the threshold.

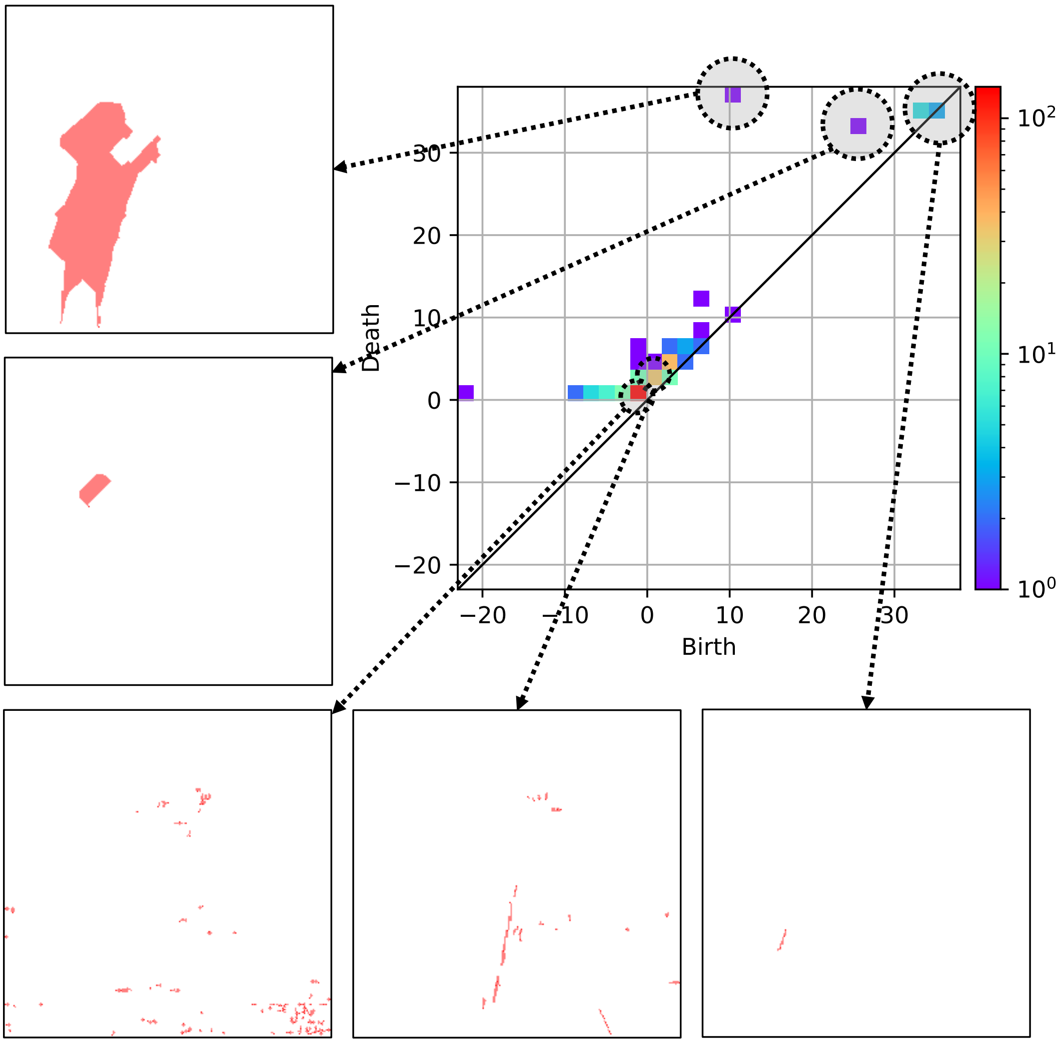


Fig. 5.6: Persistent diagram of Fig. 5.5(b) and segmented images which correspond to points on it.

When I binarize the gray-scale image in Fig. 5.5(a) using random dithering, I obtain the image shown in Fig. 5.8(a). Applying TDA with white as the standard to this image, I obtain the persistent diagram shown in Fig. 5.9. Compared to the results of Otsu's method, the points in the persistent diagram in Fig. 5.9 are scattered in the case of random dithering, while points in the persistent diagram shown in Fig. 5.6 are dense in the case of Otsu's method.

In the case of Otsu's method, when I perform inverse analysis of the hole corresponding to the point with Persistency = 28 on the persistent diagram in Fig. 5.6, I obtain the result shown in Fig. 5.7. In contrast, in the case of binarization by random dithering, when I perform inverse analysis of the hole corresponding to the point (1,

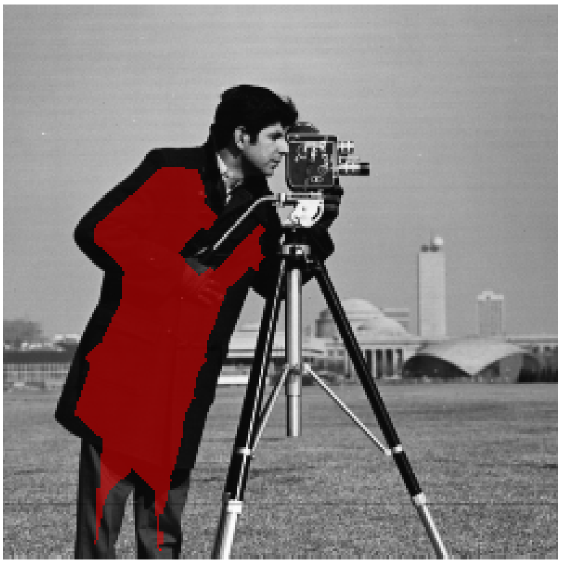
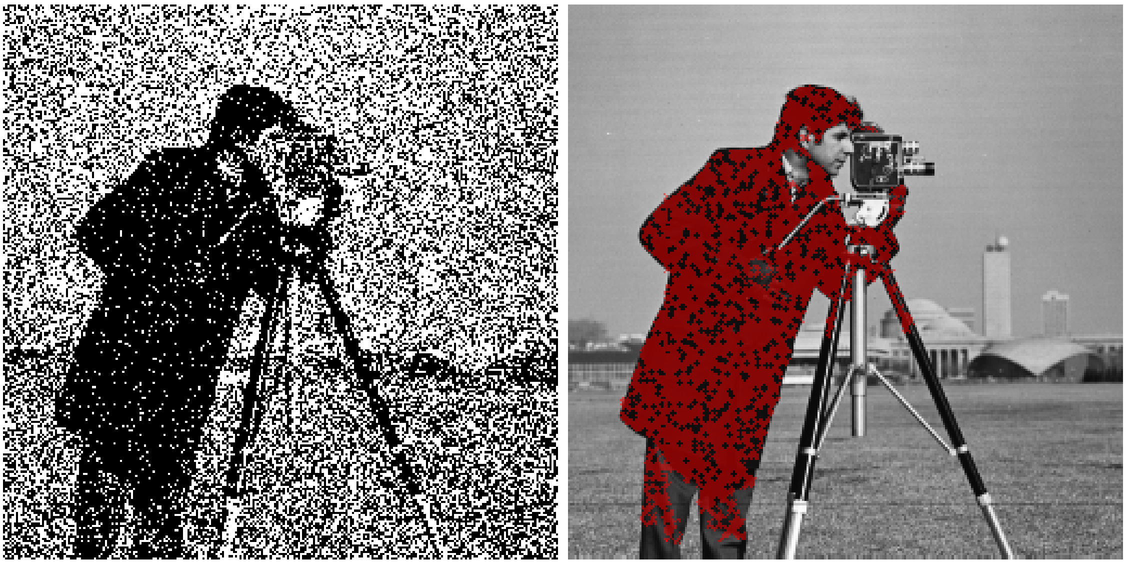


Fig. 5.7: Segmentation result for the Fig. 5.5(b). The red area is the area extracted by segmentation method.



(a) Halftone image (binarized) .

(b) Segmented image .

Fig. 5.8: Halftone image by random dither method and its segmented image.

9) farthest from the diagonal line on the persistent diagram in Fig. 5.9, I obtain the result shown in Fig. 5.8(b). As can be seen from these results, random dithering is more effective than Otsu's method in segmenting near the boundary.

Finally, I quantitatively evaluate the segmentation results using the Dice coefficient (F-score)[101, 102]. The Dice coefficient can be calculated using the following

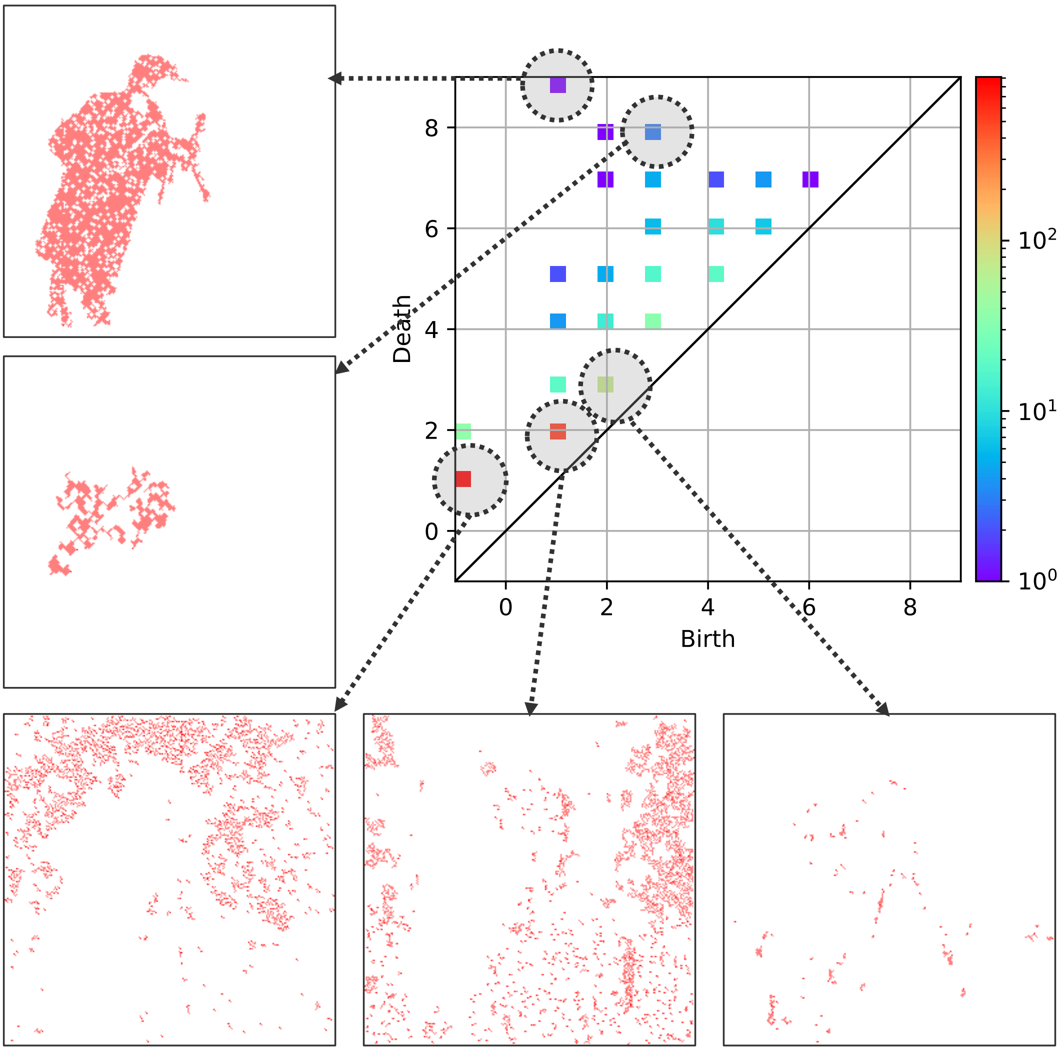


Fig. 5.9: Persistent diagram of Fig. 5.8(a) and segmented images which correspond to points on it.

equations:

$$\text{Precision} := \frac{TP}{TP+FP}, \tag{5.1}$$

$$\text{Recall} := \frac{TP}{TP+FN}, \tag{5.2}$$

$$\text{Dice coefficient} := \frac{2}{1/\text{Precision}+1/\text{Recall}}, \tag{5.3}$$

where TP, FP, and FN stand for “True Positive,“, “False Positive,“ and “False Negative,“ respectively, representing the number of correct pixels recognized as correct, the number of incorrect pixels recognized as correct, and the number of correct pixels not recognized as correct, respectively. Precision and Recall correspond to ”lack of overflow" and "lack of omission," respectively.

Here, I calculated the Dice coefficient using Fig. 5.10 as the reference image and Fig. 5.7 and Fig. 5.8 (b) as the segmented images. The Dice coefficients were 0.662 and 0.784, respectively, indicating that the result of Fig. 5.8 (b) was better.

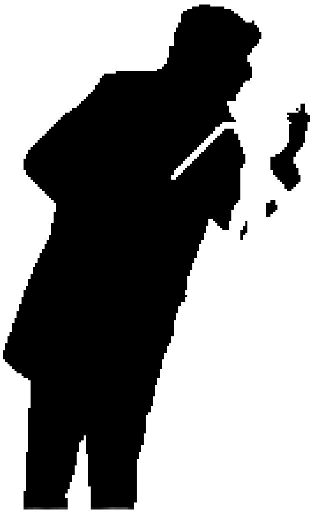


Fig. 5.10: Reference image.

In the proposed method, I use random dithering, which allows me to avoid the problem of determining the threshold that becomes a bottleneck when binarizing the gray-scale image. Therefore, it is considered useful to combine random dithering and TDA.

In summary, the contents of this chapter are as follows:

- Halftone images generally contain high spatial frequency noise, often near the diagonal line on the persistent diagram. Therefore, by combining random dithering, one of the halftoning methods, with TDA, I can segment images with a relatively simple algorithm.
- Comparing the method combining thresholding with TDA and the proposed method, the proposed method can segment images near the boundary. Furthermore, by using random dithering in the proposed method, I can avoid the problem of determining the threshold that becomes a bottleneck in thresholding.

5.3 Validation of the proposed method

In this section, I describe the advantages of halftoning in the proposed method. Specifically, I verify the effectiveness of the proposed method from three perspectives:

- Results of applying the proposed method to blurred images
- Results of applying the proposed method to other standard images and CT images
- Results of applying other halftoning methods

5.3.1 Results for blurred images

In this subsection, I describe the results of applying the proposed method to blurred images. I use the gray-scale image in Fig. 5.11. The image is obtained by applying a 5×5 average filter to the image in Fig. 5.5(a). Fig. 5.12(a) shows the result of applying the proposed method to the blurred image. Fig. 5.12(b) shows the result of applying Otsu's method to the blurred image and performing TDA. Comparing Fig. 5.12(a) and (b), I can see that segmentation is also better when random dithering is used than when Otsu's method is used.

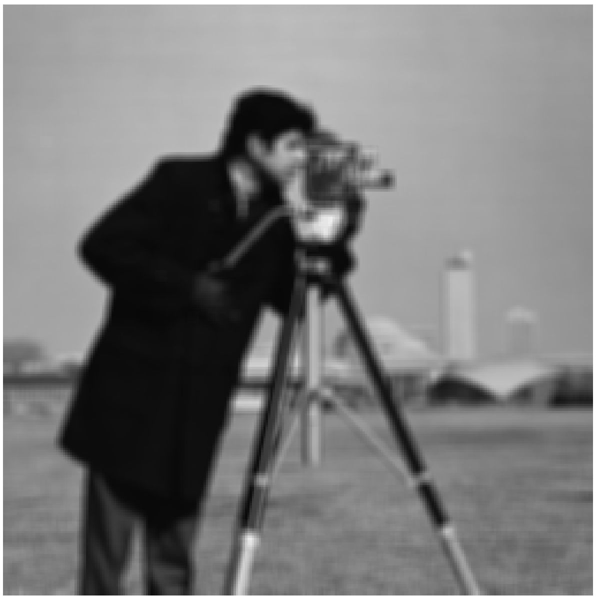
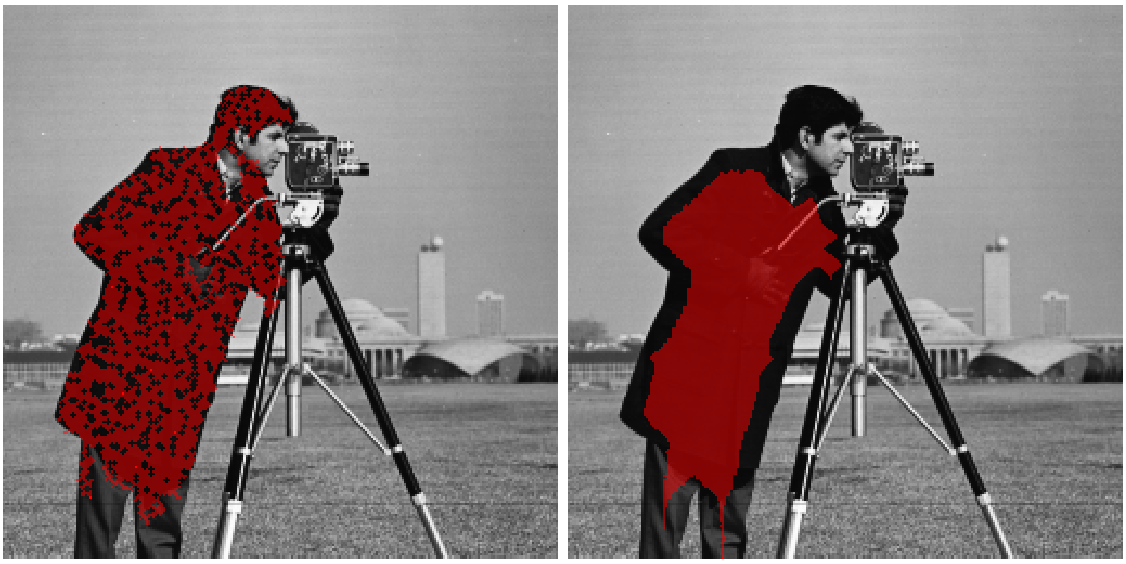


Fig. 5.11: Blurred image.



(a) Case of halftone method. (b) Case of Otsu's method.

Fig. 5.12: Comparison of segmentation results between different binarizing algorithms.

5.3.2 Results for other standard images and CT images

In this subsection, I describe the results of applying the proposed method to other standard images with low spatial frequency textures. Figure 5.13 shows the results of quantizing the gray-scale images of standard images and segmenting them using random dithering. For the images in Fig. 5.13(a), I obtained the images in Fig. 5.13(b) by extracting the parts with Persistency ≥ 4 in the persistent diagram. Similarly, for the images in Fig. 5.13(c), I obtained the images in Fig. 5.13(d) by extracting the parts with Persistency ≥ 4 in the persistent diagram. Thus, it can be confirmed that segmentation is possible by drawing the persistent diagram of the binary image by random dithering using TDA and selecting appropriate points.

In addition, I applied the proposed method to the brain CT images of a cricket in Fig. 5.14(a), (b)[103]. The results are shown in Fig. 5.14(c), (d). In Fig. 5.14(c), the centrosome part (inside the red circle in the center of Fig. 5.14(c)) is segmented because it does not come into contact with the surrounding white area. However, it was impossible to segment the centrosome when it comes into contact with the surrounding structure, as in Fig. 5.14(d).

[Remark] As mentioned in Subsection 5.2.1, applying TDA to the image in Fig. 5.5(b) results in Fig. 5.7. However, when I add the pre-processing of "adding a white frame of one pixel around the image" to the image in Fig. 5.5(b) and then apply TDA, I obtain good results as shown in Fig. 5.15(a).

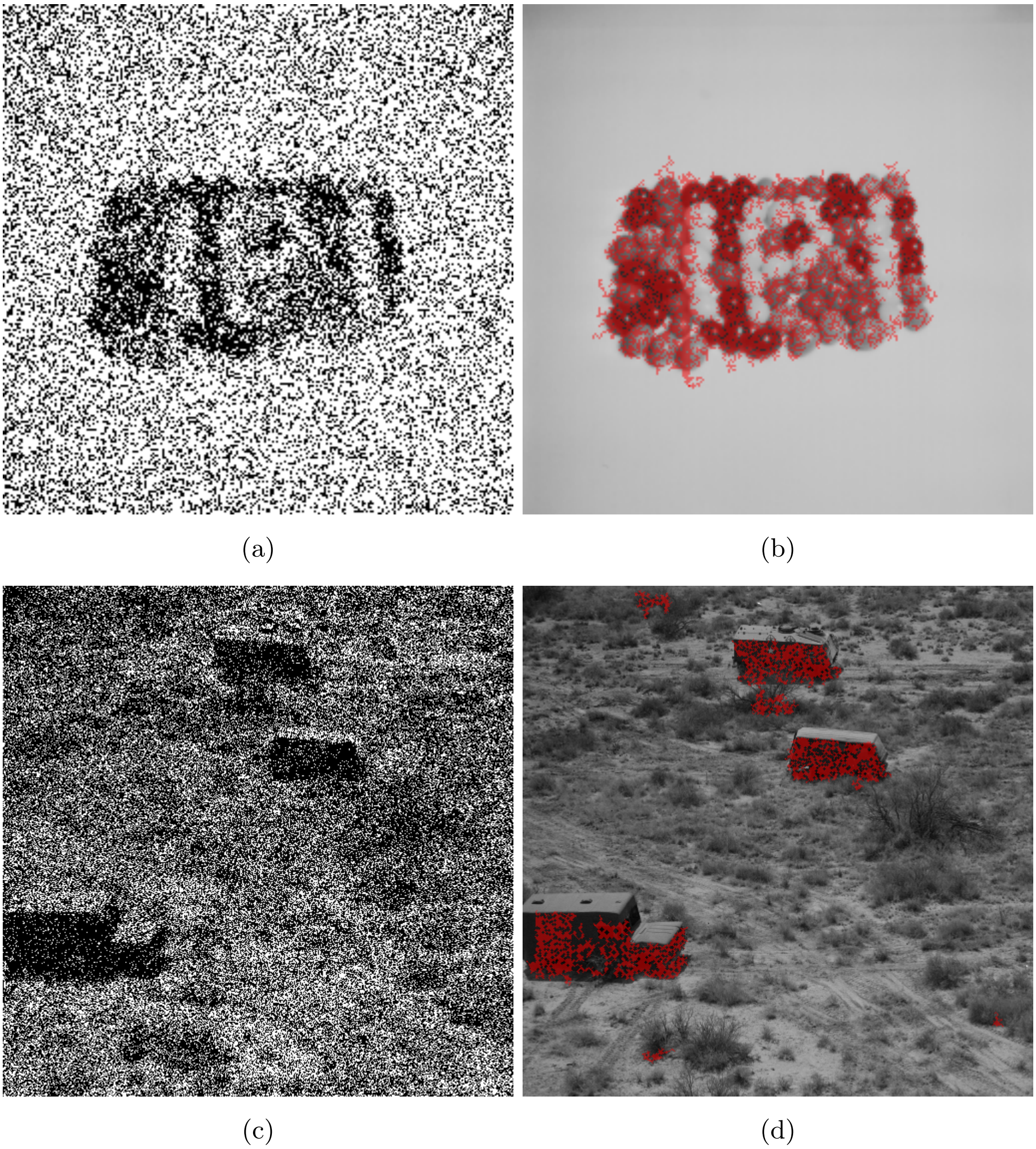
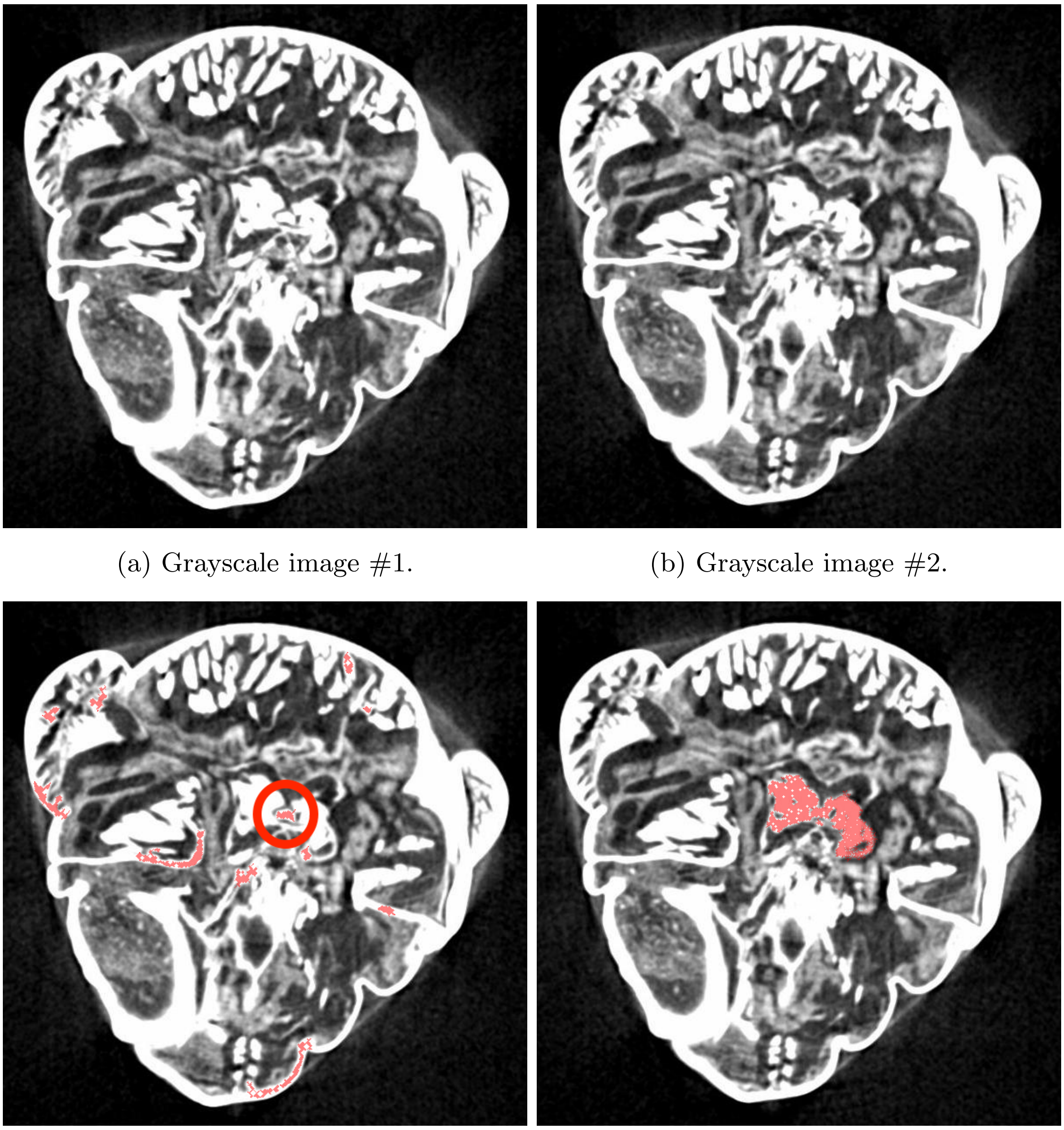

(a) (b) (c) (d)

Fig. 5.13: Halftone images and its segmented images.

In contrast, I applied Otsu's method to the original standard images in Fig. 5.13(a), (c), binarized them, and then applied TDA after adding the above pre-processing. The results are shown in Fig. 5.15(b), (c), and the expected results were not obtained. The results are considered to be due to the inability of Otsu's method to determine the threshold appropriately. From these results, it can be considered that the proposed method combining halftoning and TDA is simple in terms of:

- Avoiding the difficulty of setting the threshold during quantization



(a) Grayscale image #1.

(b) Grayscale image #2.

(c) Segmented image #1.

(d) Segmented image #2.

Fig. 5.14: Experiment result with brain CT images of cricket.

- Being able to segment any image moderately without post-processing

compared to the method with the above pre-processing, although the proposed method may be inferior in some cases.

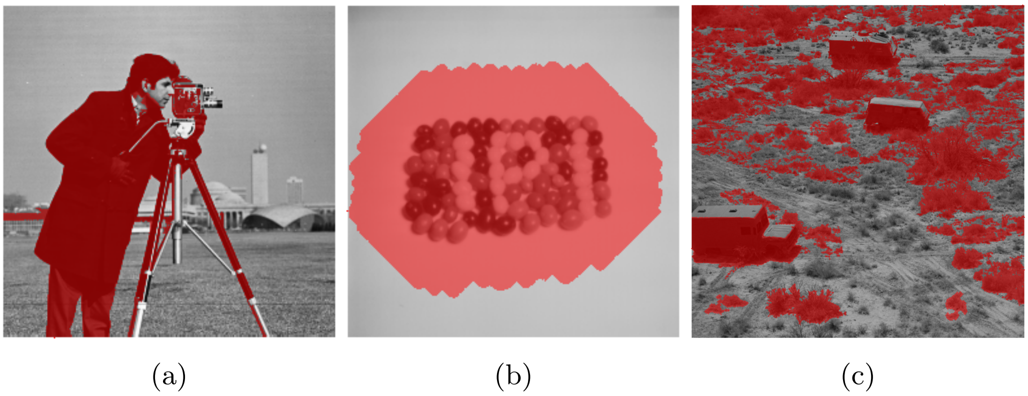


Fig. 5.15: Regions of interest segmented from binary images with white margin.

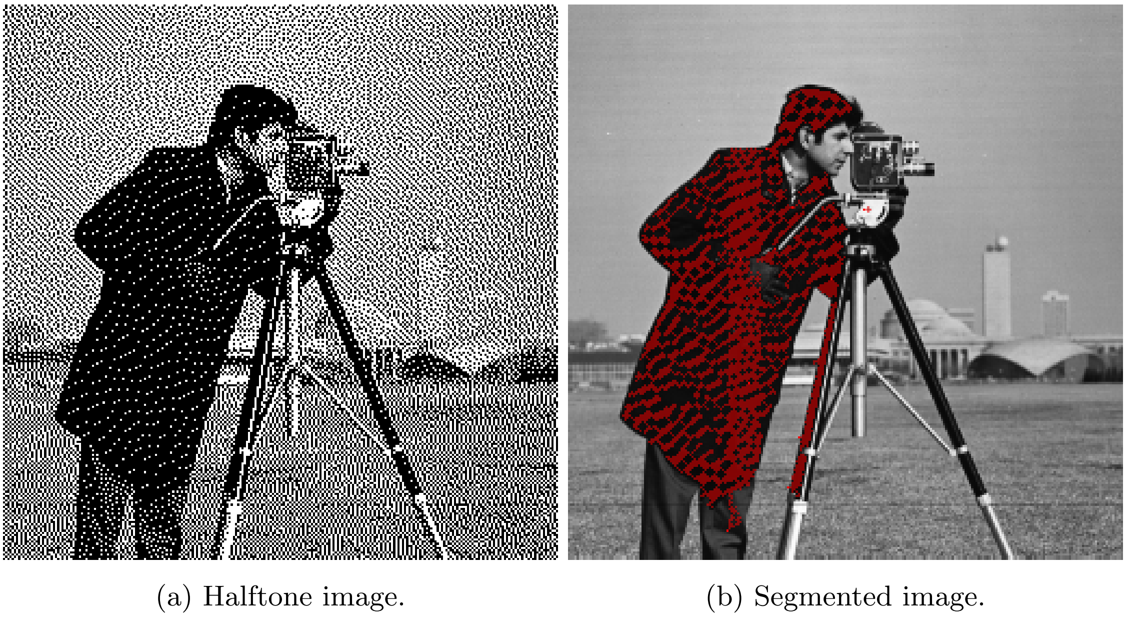

(a) Halftone image. (b) Segmented image.

Fig. 5.16: Halftone image by error diffusion method and its segmented image.

5.3.3 Comparison with other halftoning methods

I have used random dithering as the halftoning method, but in this subsection, I consider the case of the error diffusion method (Floyd & Steinberg filter)[104], another representative halftoning method.

The error diffusion method and TDA can also segment the image, as shown in Fig. 5.16. These results confirm that the proposed method yields similar results regardless of the halftoning method used. However, when comparing Fig. 5.8(b) and Fig. 5.16(b), I can see that they are slightly different. In other words, changing the halftoning method affects the features extracted by TDA and, thus, the results of the proposed method. In particular, when performing inverse analysis from the persistent diagram, it was sufficient to select one point in the case of random dithering. In contrast, in the case of error diffusion, it was necessary to choose points that satisfy Persistency ≥ 3. This point is a challenge when advancing the automation of the method, but on the other hand, the following can be considered: I confirmed that segmentation is possible by applying TDA to halftone images and drawing the persistent diagram. However, the results were influenced by the choice of halftoning method. Therefore, the development of a customized segmentation method tailored to the image by customizing the halftoning process can be expected. I would like to continue my research in this direction.

5.4 Summary

In this chapter, I confirm the effectiveness of the segmentation method that combines halftoning and TDA. As a result, I can segment standard images roughly with a simple algorithm. Future tasks include improving the accuracy of segmentation and designing halftone processing filters tailored to images. In the future, I would like to research to improve these aspects. As existing methods[86], there are segmentation methods, such as the method using TDA combined with deep learning[88], that use snakes[105], and that using level sets[106]. It is important to compare the proposed method with these methods for performance evaluation. Furthermore, I perform a quantitative evaluation in this chapter using manually cropped images and the Dice coefficient. However, the reference images and evaluation criteria vary depending on users and problem settings. Therefore, it is also a task to establish evaluation criteria tailored to users and problem settings.

Finally, the approach of combining halftoning with TDA is the first of its kind in this paper, and discussing the effectiveness of this combination is valuable for expanding the application range of TDA.

Chapter 6

Model-tuning approach to quantization process in TDA-based QNN evaluation system

In this chapter, I apply our proposed method to the system, which is composed of TDA and quantization process. The whole system is the evaluation process of quantized NNs, Σ_1 is the TDA process for quantized NNs, and Σ_2 is the quantizer Q, which quantize the weights of a original real-valued NNs. I fix the design procedure $\mathcal{M}$ of quantization process of weights and design the quantization process Σ_2 by tuning the error-diffusion filter P.

Recently, many Neural Network (NN) models and learning methods have been proposed and used in various scientific and technological fields. However, if the NN model becomes large, the number of parameters, such as weights and biases, increases. Consequently, it is difficult to build NNs on computers with limited computing resources, such as microcomputers. Thus, compression methods is needed for memory saving and computational efficiency[107].

One way to compress NN models is to quantize the weights for connections of the NN nodes. In quantization methods, weights with a large number of bits are converted into weights with a smaller number of bits. Existing quantization methods for NNs can be categorized into two approaches: Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ) [108, 109]. QAT Incorporates quantization into the learning process to obtain low-bit weight coefficients[110]. In the QAT framework, NN models can be quantized while their performance is evaluated. On the other hand, PTQ degrades the performance of a QNN compared with the original NN because of quantization of the weight coefficients of learned NNs[111, 112, 113, 114, 115, 116, 117]. In the PTQ framework, the performance degradation of QNNs is not known in advance and must be evaluated later.

One of the evaluation criteria is the inference accuracy of NNs. However, experiments with large amounts of data are needed to verify the performance of QNNs in detail. Besides, in some studies, NNs are used as a component of a system, such as a controller, e.g., model predictive control[118, 119]. When an NN is incorporated into a control system, their performance must be evaluated through simulations or experiments to confirm that they satisfy the control specifications. However, establishing a simulation environment, a dataset, and appropriate experimental conditions for large systems can be time-consuming and computationally expensive. Moreover, in the experiments, the control system may be damaged if significant performance degradation of the NN occurs, such as instability. Hence, instead of conducting experiments using NNs, an evaluation method based on their structure is needed to assess the difference in performance between the original and QNNs.

This paper proposes a method for evaluating the performance of QNNs using Topological Data Analysis (TDA)[90, 91, 92, 93, 94, 95, 96, 97]. TDA is a topology-based method of analyzing data and examines the number and size of holes in the data. The method allowed us to visually demonstrate the performance of the QNN models as a diagram without experiments or simulations. The critical point of TDA is to infer the spatial information and features of Big Data from the size and number of holes in the data. TDA compresses the information on the weights of the NN model into that on the holes of the NN model, and it is expected that the properties of the NN model can be captured. Moreover, TDA has features such as a theoretical foundation, practical computability, and robustness with small perturbations, as described in a previous paper[91], which are helpful for the evaluation of not only original NNs but also QNNs. The previous study of the TDA for NNs[95] is not directly applicable to QNNs, because of their many zero-valued weights; thus, I modified the TDA for QNNs. In addition, I demonstrated this method for NN models through numerical examples using the MNIST dataset. In the examples, I applied the proposed TDA to QNNs generated by static and dynamic quantizers [115].

In addition to evaluating QNNs, this study also aims to analyze the structure of QNNs. Various quantization methods have been developed in previous studies of NNs, and it is known that dynamic quantization outperforms static quantization[117]. However, the internal structure of QNNs has not yet been thoroughly investigated. If the internal structure of QNNs can be linked to their performance using topological indices, topology-based optimality of quantization may be defined. This study can be considered the beginning of the connection between the internal structure and topology of QNNs.

6.1 Quantization of Neural Networks

An NN is a type of machine-learning model inspired by the structure and function of the human brain. It consists of interconnected nodes organized into layers, as shown in Fig. 6.1. Each neuron is depicted in Fig. 6.2, and the forward propagation of layer

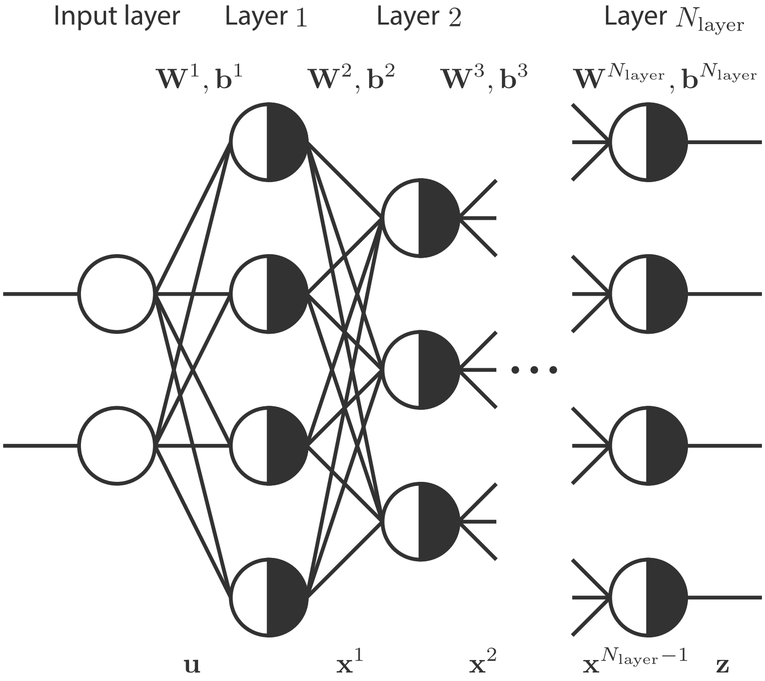

Input layer
Layer 1
Layer 2
Layer N_{layer}
$\mathbf{W}^1, \mathbf{b}^1$
$\mathbf{W}^2, \mathbf{b}^2$
$\mathbf{W}^3, \mathbf{b}^3$
$\mathbf{W}^{N_{\text{layer}}}, \mathbf{b}^{N_{\text{layer}}}$
$\mathbf{u}$
$\mathbf{x}^1$
$\mathbf{x}^2$
$\mathbf{x}^{N_{\text{layer}}-1}$
$\mathbf{z}$

Fig. 6.1: NN model and the input $\mathbf{u}$, the state $\mathbf{x}^l$, the output $\mathbf{z}$ and parameters, such as the weights $\mathbf{W}^l$, and $\mathbf{b}^l$.

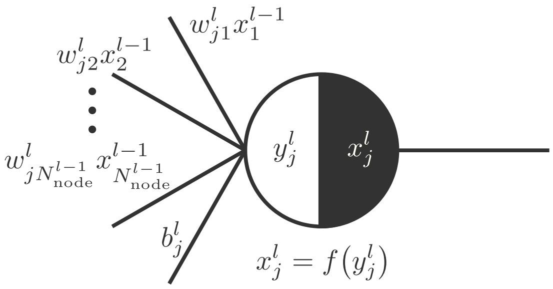


Fig. 6.2: Weighted linear summation and activation in the jth node in layer l.

l in the NN is expressed as

$$\begin{cases} \mathbf{y}^l = \mathbf{W}^l \mathbf{x}^{l-1} + \mathbf{b}^l, \\ \mathbf{x}^l = f\left(\mathbf{y}^l\right), \end{cases} \tag{6.1}$$

where $\mathbf{W}^l = \left[w_{ji}^l\right] \in \mathbb{R}^{N^l \times N^{l-1}}$ is the weight matrix, $\mathbf{b}^l = \left[b_j^l\right] \in \mathbb{R}^{N^l}$ is the bias vector, $\mathbf{x}^l = \left[x_j^l\right] \in \mathbb{R}^{N^l}$ is the state vector of layer l, and $\mathbf{y}^l = \left[y_j^l\right] \in \mathbb{R}^{N^l}$ is the output vector of layer l. Note that N^l represents the number of nodes in layer l and the activation function f is applied element-wise to the output vector $\mathbf{y}^l$ in Equation (6.1). In layer l, the $j \in \left\{1, 2, \ldots, N^l\right\}$th neuron has state $x_j^l \in \mathbb{R}$. The state x_j^l is obtained by substituting the parameter $y_j^l \in \mathbb{R}$, which is the weighted linear summation of the states of the neurons in layer $l-1$, into the activation function $f : \mathbb{R} \to \mathbb{R}$, as shown in Fig. 6.2.

I explain the quantization methods of an NN. The weight coefficients w_{ji}^l are converted from high- to low-bit values, $v_{ji}^l \in \mathbb{V}$. Note that the bias coefficients b_j^l are not quantized in this chapter. The equation for each layer of the NN with quantized weights is denoted as

$$\begin{cases} \mathbf{y}^l = \mathbf{V}^l \mathbf{x}^{l-1} + \mathbf{b}^l, \\ \mathbf{x}^l = f\left(\mathbf{y}^l\right), \end{cases} \tag{6.2}$$

where $\mathbf{V}^l = \left[v_{ji}^l\right] \in \mathbb{V}^{N^l \times N^{l-1}}$ denotes the quantized weight matrix.

There are two types of quantization methods in PTQ: static and dynamic. Static quantization is a simple method for quantizing the NN. In this chapter, I use the function of static quantization given by Equation (2) in the previous study[115] as

$$q : v_{ji}^l = \begin{cases} \operatorname{sgn}\left(w_{ji}^l\right) \left\lfloor \dfrac{\left|w_{ji}^l\right|}{d} + \dfrac{1}{2} \right\rfloor d & \text{if } \left|w_{ji}^l\right| \leq N_{\mathrm{bit}} d, \\ \operatorname{sgn}\left(w_{ji}^l\right) N_{\mathrm{bit}} d & \text{otherwise}, \end{cases} \tag{6.3}$$

which converts the original weight w_{ji} into the closest weight of all the candidate values $0, \pm d, \pm 2d, \ldots, \pm N_{\mathrm{bit}} d$. In the case of static quantization, the quantized weights v_{ji}^l depend on the original weights w_{ji}^l, quantization width d, and number of quantization bits N_{bit}.

Tsubone et al. (2023) proposed a dynamic quantization method for discretizing NN weights[115]. In dynamic quantization, the filtered quantization error for each weight is propagated to other weights to reduce the output error of each layer. Therefore, the quantized weights v_{ji}^l depend on the quantization error as well as the original weights

w^l_{ji}, quantization width d, and number of quantization bits N_{bit}. An overview of the dynamic quantization process is described below.

1. One weight in each layer is quantized, and the quantization errors are propagated to the next node in the same layer to reduce the gap between the pre- and post-quantization outputs.
2. In each layer, the propagation route and value of the quantization error are determined according to the 2-norm distance of the feature vectors calculated by input data.

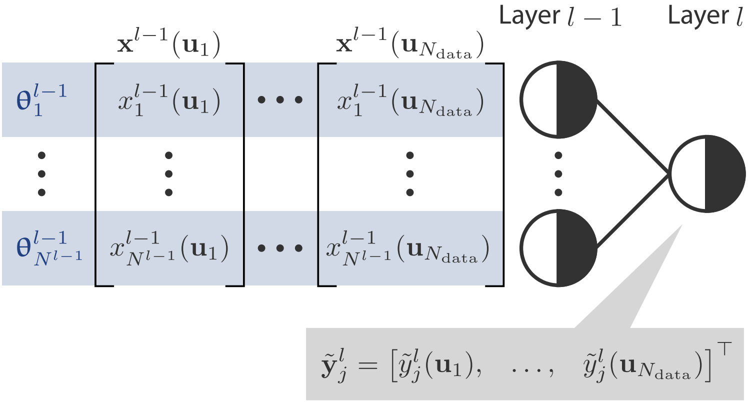

Layer $l-1$
Layer l
$\mathbf{x}^{l-1}(\mathbf{u}_1)$
$\mathbf{x}^{l-1}(\mathbf{u}_{N_{\mathrm{data}}})$
$\boldsymbol{\theta}^{l-1}_1$
$x^{l-1}_1(\mathbf{u}_1)$
$x^{l-1}_1(\mathbf{u}_{N_{\mathrm{data}}})$
$\boldsymbol{\theta}^{l-1}_{N^{l-1}}$
$x^{l-1}_{N^{l-1}}(\mathbf{u}_1)$
$x^{l-1}_{N^{l-1}}(\mathbf{u}_{N_{\mathrm{data}}})$
$\tilde{\mathbf{y}}^l_j = [\tilde{y}^l_j(\mathbf{u}_1), \ \ldots, \ \tilde{y}^l_j(\mathbf{u}_{N_{\mathrm{data}}})]^\top$

Fig. 6.3: Construction method for feature vectors $\boldsymbol{\theta}^{l-1}_i$ and output vectors $\tilde{\mathbf{y}}^l_j$ from N_{data} data.

Here, I explain the algorithm for dynamic quantization in detail. I assume that a trained NN and N_{data} sets of training data are given. Here, $\mathbf{u}_n \in \mathbb{R}^{N_{\mathrm{input}}}$ $(n = 1, 2, \ldots, N_{\mathrm{data}})$ are defined as the nth input data, and N_{input} represents the number of input nodes. I call the NN with high-bit weights the original NN and the NN with quantized weights from layer 1 to layer l the lth QNN. The output vector of the jth node of layer l in the original NN is $\hat{\mathbf{y}}^l_j = [\hat{y}^l_j(\mathbf{u}_1), \hat{y}^l_j(\mathbf{u}_2), \ \ldots, \hat{y}^l_j(\mathbf{u}_{N_{\mathrm{data}}})]^\top \in \mathbb{R}^{N_{\mathrm{data}}}$, and that of the jth node of layer l in the $l-1$th QNN is $\tilde{\mathbf{y}}^l_j = [\tilde{y}^l_j(\mathbf{u}_1), \tilde{y}^l_j(\mathbf{u}_2), \ldots, \tilde{y}^l_j(\mathbf{u}_{N_{\mathrm{data}}})]^\top \in \mathbb{R}^{N_{\mathrm{data}}}$. Note that $\hat{y}^l_j(\mathbf{u}_n), \tilde{y}^l_j(\mathbf{u}_n)$ mean the output vectors $\hat{y}^l_j, \tilde{y}^l_j$ in the cases where the input data $\mathbf{u}_n$ are fed to the original NN and the $l-1$ QNN, respectively. The feature vector $\boldsymbol{\theta}_i^{l-1} \in \mathbb{R}^{N_{\text{data}}}$ is defined as

$$\boldsymbol{\theta}_i^{l-1} = \begin{cases} \begin{bmatrix} x_i^{l-1}(\mathbf{u}_1) \\ x_i^{l-1}(\mathbf{u}_2) \\ \vdots \\ x_i^{l-1}(\mathbf{u}_{N_{\text{data}}}) \end{bmatrix} & \text{if } l \in \{2, 3, \dots, N_{\text{layer}}\}, \\ \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{iN_{\text{data}}} \end{bmatrix} & \text{otherwise,} \end{cases} \tag{6.4}$$

using the input data $\mathbf{u}_n \in \mathbb{R}^{N_{\text{input}}}$ $(n = 1, 2, \dots, N_{\text{data}})$, shown in Fig. 6.3. I denote the number of layers in the NN as N_{layer}. The feature vector $\boldsymbol{\theta}_i^{l-1} \in \mathbb{R}^{N_{\text{data}}}$ of the $l-1$ QNN is the vector input to the ith node in layer l for each input data $\mathbf{u}_n$.

Then, I define the dynamic quantizer for the NN, which is denoted as

$$Q^l : \begin{cases} \boldsymbol{\Xi}_{k+1}^l = \boldsymbol{\Xi}_k^l + \boldsymbol{\theta}_{\sigma^{l-1}(k)}^{l-1} \left(\mathbf{v}_{\sigma^{l-1}(k)}^l - \mathbf{w}_{\sigma^{l-1}(k)}^l \right)^\top, \\ \mathbf{v}_{\sigma^{l-1}(k)}^l = q\left(-\left(\boldsymbol{\theta}_{\sigma^{l-1}(k)}^{l-1} \right)^\dagger \boldsymbol{\Xi}_k^l + \mathbf{w}_{\sigma^{l-1}(k)}^l{}^\top \right)^\top \quad \left(k = 1, 2, \dots, N^{l-1}\right), \end{cases} \tag{6.5}$$

where $\sigma^{l-1}(\cdot)$ is the permutation of the nodes in layer $l-1$; $q(\cdot)$ is the quantization function defined in Equation (6.3); $\mathbf{v}_{\sigma^{l-1}(k)}^l, \mathbf{w}_{\sigma^{l-1}(k)}^l \in \mathbb{R}^{N^l}$ are the column vectors of $\mathbf{V}^l, \mathbf{W}^l$; and $\Xi_k \in \mathbb{R}^{N_{\text{data}} \times N^l}$ is the quantization error matrix. The jth column vector of the quantization error matrix $\boldsymbol{\Xi}_1^l$ is initialized as

$$\boldsymbol{\xi}_{1j}^l = \begin{cases} \tilde{\mathbf{y}}_j^l - \hat{\mathbf{y}}_j^l & \text{if } l \in \{2, 3, \dots, N_{\text{layer}}\}, \\ \mathbf{0} & \text{otherwise,} \end{cases} \tag{6.6}$$

where j is in the set $\{1, 2, \dots, N^l\}$.

Finally, I define the permutation $\{\sigma^{l-1}(1), \sigma^{l-1}(2), \dots, \sigma^{l-1}(N^{l-1})\}$, which expresses the propagation route of the quantization error. The permutation is calculated as follows: First, I construct the complete graph $\mathcal{G}^{l-1} = \left\{\mathcal{V}^{l-1}, \mathcal{E}^{l-1}\right\}$, where $\mathcal{V}^0 = \{1, 2, \dots, N_{\text{input}}\}$ and $\mathcal{V}^{l-1} = \{1, 2, \dots, N^{l-1}\}$ $(l = 2, 3, \dots, N_{\text{layer}})$ are the set of vertices and $\mathcal{E}^{l-1} = \{(i, j) \mid i, j \in \mathcal{V}^{l-1}\}$ is the set of edges, using the weight coefficients a_{ij}^{l-1} defined as

$$a_{ij}^{l-1} = \left\| \boldsymbol{\theta}_i^{l-1} - \boldsymbol{\theta}_j^{l-1} \right\|_2. \tag{6.7}$$

Then, I determine the permutation $\{\sigma^{l-1}(1), \sigma^{l-1}(2), \ldots, \sigma^{l-1}(N^{l-1})\}$ of the vertices $\mathcal{V}^{l-1}$ by solving the optimization problem formulated as Traveling Salesman Problem. The optimal permutation $\{\sigma^{l-1}(1), \sigma^{l-1}(2), \ldots, \sigma^{l-1}(N^{l-1})\}$ minimizes the evaluation function J, which is given by the sum of the distances between adjacent nodes:

$$J = \sum_{k=1}^{N^{l-1}-1} a^{l-1}_{\sigma^{l-1}(k)\sigma^{l-1}(k+1)}. \tag{6.8}$$

Using the propagation route $\{\sigma^{l-1}(1), \sigma^{l-1}(2), \ldots, \sigma^{l-1}(N^{l-1})\}$, I convert the high-bit weights $\mathbf{W}^l$ of layer l into low-bit weights $\mathbf{V}^l$ using the dynamic quantizer (6.5) along the optimal permutation $\{\sigma^{l-1}(1), \sigma^{l-1}(2), \ldots, \sigma^{l-1}(N^{l-1})\}$. The first equation of the dynamic quantizer (6.5) is the error propagation equation, and the second equation of the dynamic quantizer (6.5) subtracts the error from the high-bit weights and statically quantizes them by Equation (6.3).

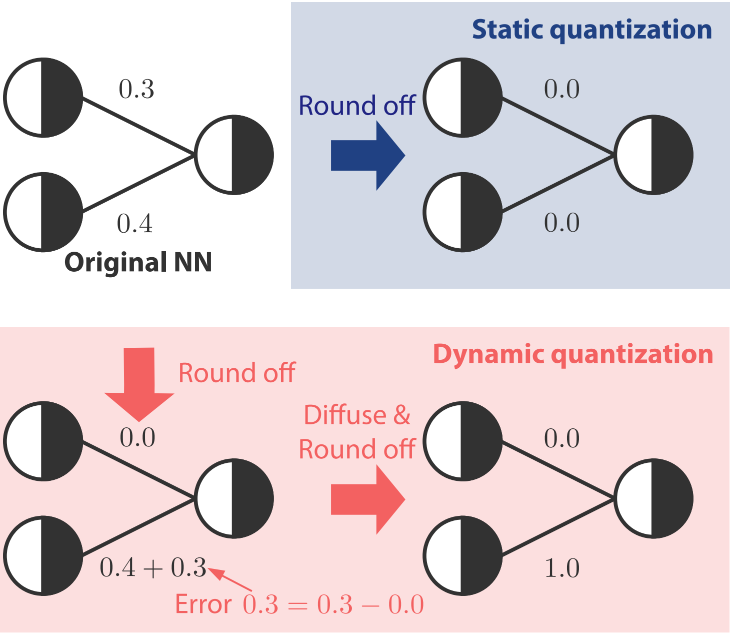


Fig. 6.4: Difference in the quantization process between static and dynamic quantization.

An illustrative example of the difference between static and dynamic quantization is shown in Fig. 6.4. If there are the original weights whose values are sufficiently smaller than the quantization width d, all the quantized weights become 0 through static quantization because static quantization rounds the weights independently. In contrast, dynamic quantization quantizes weights and diffuses the quantization error to other weights. Thus, even if the weights are sufficiently smaller than the quantization width d, they are not quantized to 0 by dynamic quantization, and input–output relationships of the NN are preserved to some extent.

6.2 TDA-based Performance Analysis

6.2.1 Basics of TDA

See the Subsection 5.1.1 for the basics of TDA.

6.2.2 TDA for NN

Watanabe and Yamana[95] applied the TDA to the performance analysis of NNs. I explain the definitions of birth and death as follows. Note that the mathematical terms such as simplex, complex, and filtration, which are required for the definition of TDA for NNs, were defined in Definitions 1–4 of [120].

First, I represent the NN as a finite directed-weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ with no self-loops or double edges. The set of nodes $\mathcal{V} = \{1, 2, \ldots, N\}$ corresponds to the set of all the neurons in the NN, the set of edges $\mathcal{E} = \{(i, j) \mid i, j \in \mathcal{V}\}$ corresponds to the set of connections between the neurons, and the weight matrix $\mathcal{W} = \{w_{ji}\}$ corresponds to the set of weights of the all connections. N denotes the number of all the nodes in the NN.

In [121], the clique complex $K(\mathcal{G})$ is defined from the graph $\mathcal{G}$ as

$$\begin{aligned} K(\mathcal{G})_0 &= \mathcal{V}, \\ K(\mathcal{G})_p &= \{(k_1, k_2, \ldots, k_p) \mid k_i \in \mathcal{V}, \\ &\qquad (k_i, k_j) \in \mathcal{E} \text{ for all } k_j > k_i\}, \end{aligned} \tag{6.9}$$

where the parameter p holds $p \geq 1$, and $K(\mathcal{G})_p$ denotes the set of p-simplexes on the graph $\mathcal{G}$. For example, 0-simplex is a connected component, 1-simplex is an edge, and 2-simplex is a triangle.

I define the relevance R_{ji} of edge (v_i, v_j) as

$$R_{ji} = \begin{cases} \dfrac{\max(w_{ji}, 0)}{\sum_{i, i \neq j} \max(w_{ji}, 0)} & i \neq j, \\ 1 & \text{otherwise.} \end{cases} \tag{6.10}$$

The relevance represents the normalized closeness between two adjacent nodes. I extend the definition of the relevance, and the relevance between nodes that are not directly connected is defined as follows:

$$\tilde{R}_{ji} = \max_{(i,m_1,m_2,\ldots,m_k,j)\in L_{ji}} R_{im_1}R_{m_1m_2}\cdots R_{m_kj}, \tag{6.11}$$

where L_{ji} denotes the set of all possible paths $(i, m_1, m_2, \ldots, m_k, j)$ from node i to node j. Equation (6.11) indicates that the relevance $\tilde{R}_{ji}$ is set as the maximum product of the relevance between two nodes on all paths from i to j.

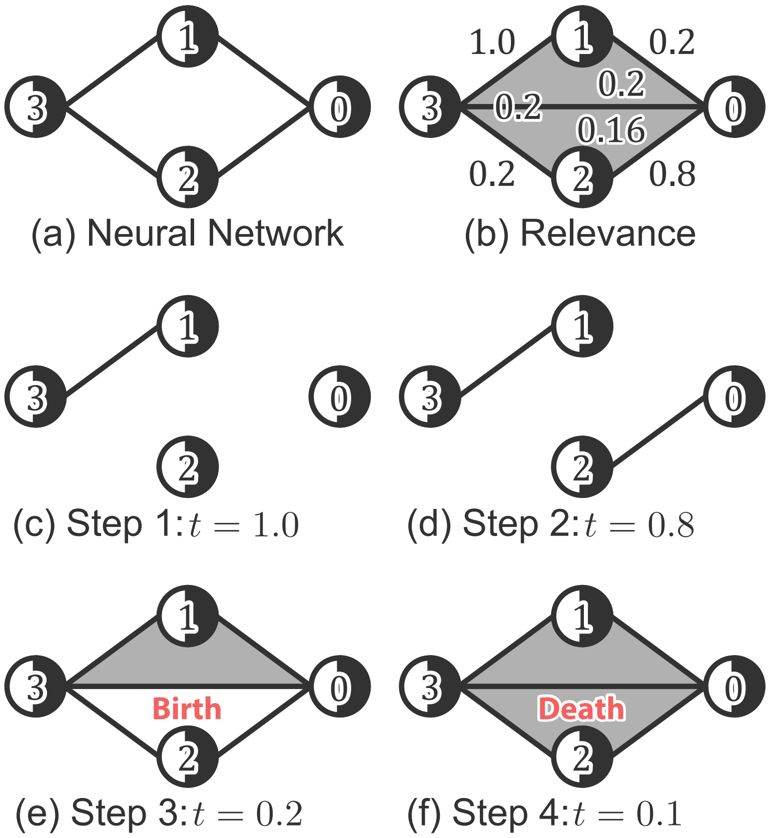
Similar to Equation (6.9), the clique complex with the threshold value and filtration were defined in previous research[95] using the relevance $\tilde{R}_{ji}$ and the threshold t. The clique complex with threshold value K_p^t is defined as follows:

$$K_p^t : \begin{cases} \mathcal{V} & \text{if } p = 0, \\ \{(k_1, k_2, \ldots, k_p) \mid k_i \in \mathcal{V}, & \text{if } p \geq 1, \\ \tilde{R}_{k_jk_i} \geq t \text{ for all } k_j > k_i\} & \end{cases} \tag{6.12}$$

where t is restricted to $0 \leq t \leq 1$ and the nodes are numbered in ascending order from the output layer to the input layer. Propositions 1 and 2 in [95] show that the set $K^t = \bigcup_{p=0}^{n} K_p^t$ is a simplicial complex and that the filtration of K^t can be formed. The birth and death of a hole can be calculated, and a persistent diagram can be drawn because filtration can be formed.

For example, I consider a part of the NN shown in Fig. 6.5(a). In Fig. 6.5(b), the relevance $\tilde{R}_{ji}$ is defined by Equations (6.10) and (6.11). When the threshold t is 1.0, only the edges between Nodes 1 and 3 exceed the threshold, as shown in Fig. 6.5(c). When the threshold t is 0.2, all edges including nodes 0, 2, and 3 exceed the threshold, and the hole is born in Fig. 6.5(e). When the threshold t is 0.1, the triangle including nodes 0, 2, and 3 exceeds the threshold, and the hole is dead, as shown in Fig. 6.5(f). The persistent diagram of an NN is shown in Fig. 6.6.

The intuitive process of TDA for NNs is as follows: The relevance R_{ji} is defined by Equation (6.10) from the weights $w_{ji} \in \mathbb{R}$. In addition, the relevance of three or more nodes is defined as the maximum product of the relevance of all paths by Equation (6.11). I then define the thresholds of the edges and triangles composed of nodes. The thresholds of the edges are defined as the relevance itself, and those of the triangles are defined as the maximum relevance of the path containing the three nodes of the triangle. When threshold t is varied from 1 to 0, birth is the radius at which the hole enclosed by the edges is formed and death is the radius at which the hole collapses.



(a) Neural Network (b) Relevance

(c) Step 1:$t = 1.0$ (d) Step 2:$t = 0.8$

(e) Step 3:$t = 0.2$ (f) Step 4:$t = 0.1$

Fig. 6.5: Relevance in the NN and the change in the shape of the NN based on the threshold t.

6.2.3 Simulation Example of Original NN: MNIST Classification

In this section, I use an NN to classify the MNIST dataset[122]. In this classification problem, I classify hand-drawn numbers from zero to nine. The input of the NN is 28×28 pixels of the images, and the output is the probability that corresponds to a number from 0 to 9. The NN model consists of the input layer, two hidden layers, and the output layer, and these layers have $28 \times 28 = 784$, 300, 100, and 10 nodes, respectively. The training data and test data are 60000 and 10000 images, respectively, and I set the batch size and number of epochs to 256 and 10, respectively. As a result, the prediction accuracy of the original NN is 89.7%.

A persistent diagram of the original NN is shown in Fig. 6.7. It is computed using Dionysus[96, 97]. Note that the color bar is on a logarithmic scale with a maximum value of $\log_{10}(20000)$. Each scale of the persistent diagrams corresponds to 64 steps of threshold t changes from 1 to 0, and these steps are as follows: $1.0 \times$

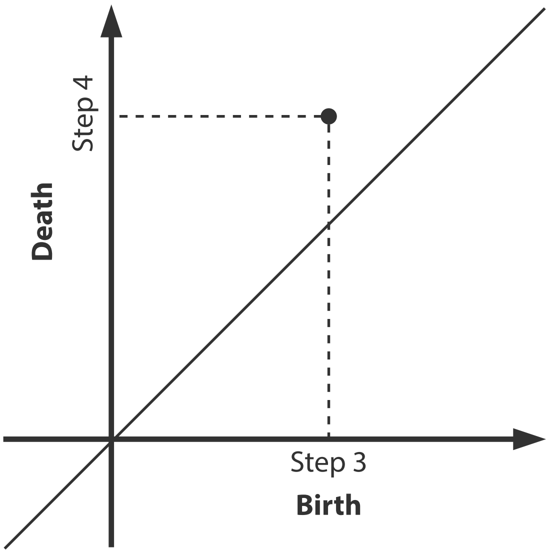


Fig. 6.6: Persistent diagram of the example NN in the Fig. 6.5, where the birth and death are defined as the number of steps.

$10^0,\ 0.9\times10^0,\ \ldots,\ 0.2\times10^0,\ 1.0\times10^{-1},\ 0.9\times10^{-1},\ \ldots,\ 0.2\times10^{-1},\ \ldots,\ 1.0\times10^{-6},\ 0.9\times10^{-6},\ \ldots,\ 0.2\times10^{-6},\ 1.0\times10^{-7}$. In Fig. 6.7, I observe many holes within approximately 10 steps in the gap between birth and death. Additionally, in Fig. 6.7, almost all the holes exist in the ranges of 15–55 for the birth radius and 30–65 for the death radius.

6.3 TDA-based Analysis of QNN

6.3.1 TDA for QNN

I explained the TDA for NNs in Section 6.2.2. However, it is difficult to apply TDA to a QNN, because most of the weights become 0, and many thresholds of the edges and the triangles are also set to 0. As a result, I lose the information of holes in the QNN.

To avoid the loss of hole information, I add a small value to each quantized weight and compute the relevance using the revised weights. I define the relevance R_{ji} as

$$R_{ji} = \begin{cases} \dfrac{\max\left(v_{ji}+\varepsilon,0\right)}{\sum_{i,i\neq j}\max\left(v_{ji}+\varepsilon,0\right)} & i\neq j, \\ 1 & \text{otherwise}, \end{cases} \tag{6.13}$$

instead of using Equation (6.10). Note that I use the positive value $\varepsilon = 0.01$ in the following example.

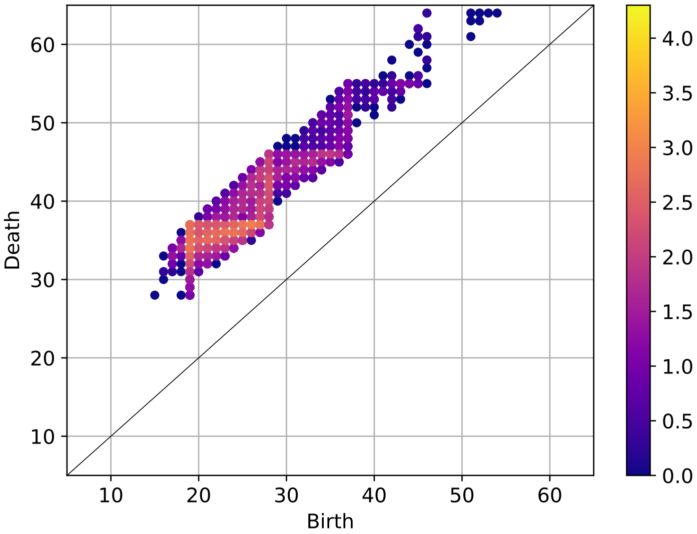
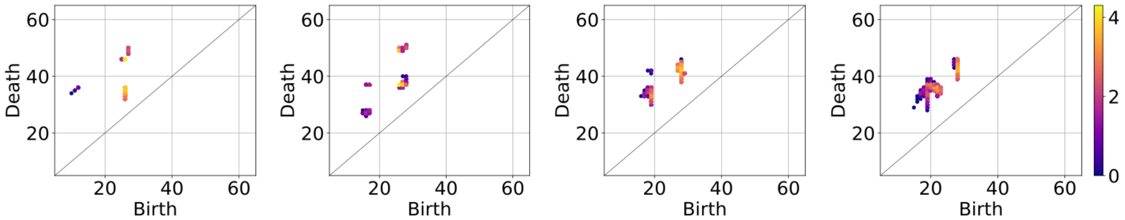


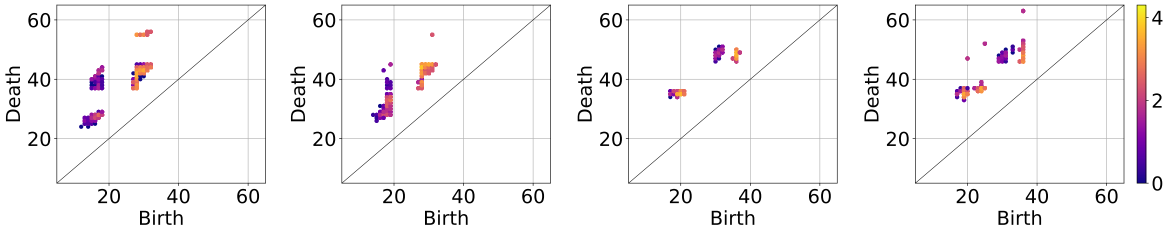
Fig. 6.7: Persistent diagram of the original NN for MNIST image classification

.



(a) $N_{\text{bit}} = 1$ (b) $N_{\text{bit}} = 2$ (c) $N_{\text{bit}} = 4$ (d) $N_{\text{bit}} = 8$

Fig. 6.8: Persistent diagrams of four static QNNs with different numbers of bits $N_{\text{bit}} = 1, 2, 4, 8$.



(a) $N_{\text{bit}} = 1$, dynamic (b) $N_{\text{bit}} = 2$, dynamic (c) $N_{\text{bit}} = 1$, random (d) $N_{\text{bit}} = 2$, random

Fig. 6.9: Persistent diagrams of two dynamically quantized and two randomly weighed NNs with different numbers of bits $N_{\text{bit}} = 1, 2$.

6.3.2 Evaluation of QNNs quantized by the static quantization method

In this section, I compare the performance and TDA results of QNNs quantized by the static quantization method (hereinafter referred to as the static QNNs) with various numbers of quantization bits N_{bit}. I use static QNNs with $N_{\text{bit}} = 1, 2, 4, 8$ quantization bits. The inference accuracy of these NNs is 8.92%, 68.1%, 88.5%, and 88.1%, and the settings and performance of the NNs are summarized in Table 6.1. In the case of low-bit static quantization, many weights are rounded to zero due to rough resolution quantization, resulting in low inference accuracy. In the case of high-bit static quantization, the resolution is higher and closer to the original NN, resulting in higher inference accuracy.

Table 6.1: Settings and performance of the original NN, static QNNs, dynamic QNN, and randomly weighted NNs.

Quantizer	Widnth d	Bits N_{bit}	Accuracy
Continuous	-	-	89.7%
Static	0.4	1	8.92%
Static	0.2	2	68.1%
Static	0.1	4	88.5%
Static	0.05	8	88.1%
Dynamic	0.4	1	85.7%
Dynamic	0.2	2	87.7%
Random	0.4	1	9.86%
Random	0.2	2	11.4%

I then apply the TDA-based method to the NNs. Persistent diagrams of the static QNNs are shown in Fig. 6.8. I mentioned earlier that the inference accuracy of static QNNs increases as the number of bits increases (Table 6.1). Focusing on the persistent diagrams of static QNNs in Fig. 6.8, I can see that as the number of bits increases, the number of holes increases in the regions where the birth radius is between 15 and 30 and the death radius is between 25 and 40. Focusing on the persistent diagram of the original NN in Fig. 6.7, I can see that holes are also concentrated in the birth radius (15 to 30) and death radius (25 to 40). This indicates that the resolution increases, and the static QNNs become similar to the original NN. In addition, holes in these regions may affect performance.

6.3.3 Evaluation of QNNs quantized by the dynamic quantization method

In this section, I compare the performance and TDA results of QNNs quantized by the dynamic quantization method (hereinafter referred to as the dynamic QNNs) and static QNNs with various numbers of quantization bits. I use dynamic QNNs with $N_{\text{bit}} = 1, 2$ quantization bits. The performance of the dynamic QNNs is 85.7% and 87.7%, and the settings and performance of the NNs are summarized in Table 6.1.

The persistent diagrams of the dynamic QNNs are shown in Fig. 6.9(a), (b). The performance of dynamic QNNs is high even when the number of bits N_{bit} is small (Table 6.1), and the holes of the dynamic QNNs exist in the regions around 15 to 30 for birth radius and 25 to 40 for death radius. Thus, the result suggests that the performance is maintained if NNs are quantized to preserve the holes in which holes in the persistent diagram of the original NN are concentrated. I can also explain why the original NN has a continuous distribution of holes, whereas the QNNs have a discrete distribution. In the original NN, the weights are conserved in 64 bits, so each relevance has almost continuous values. On the other hand, the quantized N N has only a limited number of weight coefficients, which limits the variety of values of relevance. Therefore, the birth and death computed from the relevance are also expected to be discrete.

The dynamic QNNs have higher performance, although the NNs are quantized by small number of bits ($N_{\text{bit}} = 1, 2$) compared with static QNNs. This is because dynamic quantization maintains the relationship between the input and output in each layer of NNs by minimizing quantization errors and preserving the weight values of essential nodes, as shown in Fig. 6.4.

In addition, I compare the dynamic QNNs and randomly weighed NNs. Randomly weighted NNs have randomly selected weights of $0, \pm 0.4$ and $0, \pm 0.2, \pm 0.4$ when the number of bits N_{bit} is 1 and 2, respectively. The performance of the randomly weighed NNs with $N_{\text{bit}} = 1, 2$ quantization bits is 9.86% and 11.4%, and the settings and performance of the NNs are presented in Table 6.1. The persistent diagrams of the randomly weighed NNs are shown in Fig. 6.9(c), (d). In the persistent diagrams of the randomly weighed NNs in Fig. 6.9(c), (d), the number of holes in the region round 15 to 30 for birth radius and 25 to 40 for death radius is reduced. Detailed and quantitative analysis will be conducted in the future.

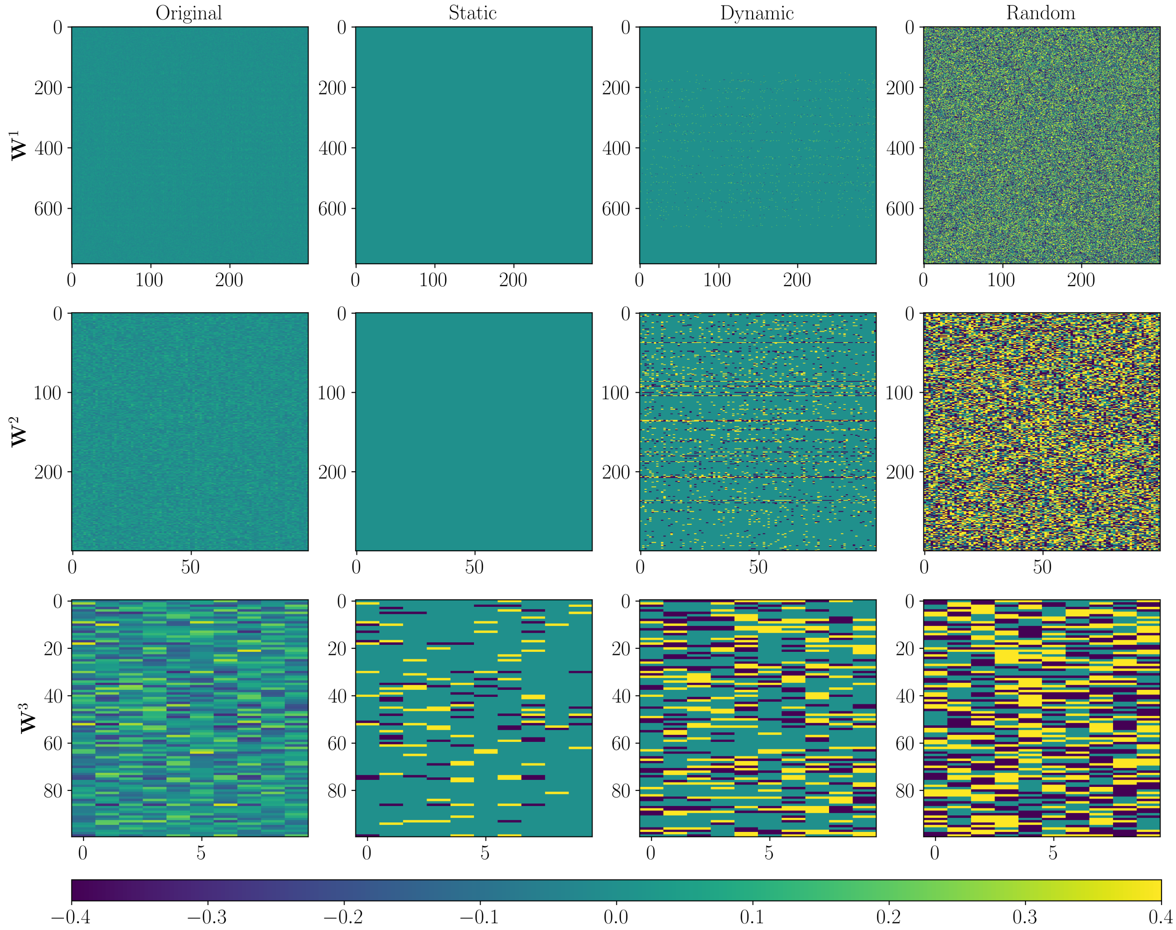


Fig. 6.10: Heatmaps of the weights of the original, statically quantized, dynamically quantized, and randomly weighed NNs with $N_{\text{bit}} = 1, 2$.

6.3.4 Comparison of Persistent Diagrams and Heatmaps

Finally, I compare the heatmaps of the original NN, the static QNN, dynamic QNN, and randomly weighed NN when the number of bits is $N_{\text{bit}} = 1$. Heatmaps are mainly used for the visualization of convolutional neural networks[123] and show the values of the weights of the NNs as colors. A comparison with a simple method using heatmaps confirmed the usefulness of focusing on the structure of NNs in the proposed TDA-based method.

The heatmaps are shown in Fig. 6.10, where the color bars indicate the weight values of the NNs. As indicated by the weights $\mathbf{W}_1$ and $\mathbf{W}_2$ in the heatmaps, the original NN has the most weights close to 0, all the weights of the static QNN become 0, and the dynamic QNN has some weights whose absolute values are close to 0.4.

These heatmaps are different, and it is difficult to evaluate the performance of NNs using heatmaps. However, in the persistent diagrams, when the performance of QNNs is higher, the distribution of holes is closer to that of the original NN. In addition, the randomly weighed NN exhibits worse performance similar to the static QNN than the original NN. However, the heatmap of the randomly weighed NN in the fourth column of Fig. 6.10 differs from that of the static QNN because the weights are assigned randomly. In contrast, the persistent diagram of a randomly weighed or static QNN has few holes in the region where the birth radius is between 15 and 30 and the death radius is between 25 and 40. Moreover, the heatmap shows a diagram for each layer, whereas the persistent diagram shows the overall characteristics in a single diagram. Thus, in contrast to heatmaps, persistent diagrams can intuitively express performance degradation.

6.4 Summary

I proposed a TDA-based evaluation method for QNNs. I evaluated the proposed method using QNNs for the classification of the MNIST dataset as an example, and I succeeded in visualizing the performance of the QNNs. As the performance degradation decreased, the persistent diagrams became more similar to the original persistent diagrams. In addition, persistent diagrams were found to be a more intuitive representation of performance than heatmaps.

In future work, I will modify this method to evaluate QNNs quantitatively. In this chapter, I used persistent diagrams to visualize the performance; however, the persistent diagrams have various parameters, such as the number and lifetime of holes. Therefore, if I can theoretically link these parameters to the performance of a QNN, the proposed method will allow quantitative evaluation of QNNs. Other future research direction is the design of a quantization filter or numerical optimization of the quantization method to maintain the topological features. Additionally, the applicability of the proposed method to Transformer models should be explored. Transformer models are widely used in natural language processing, and the quantization of Transformer models and the performance analysis of quantization are important issues. If a module is an NN model, the proposed method can be applied to the module; however, there are submodules in Transformer models. Thus, I must investigate the application of our method to multiple NN modules.

Chapter 7

Conclusion

In this thesis, I have proposed a new system design method based on model tuning. Specifically, the plant and the structure of the controller is given, and I find a model that reflects the designer's intention. The model is called as "Design Model" in this thesis. The main contributions of the thesis are summarized as follows:

- In Chapter 2, I first discuss two types of control system design, model-based and data-driven, as a background of conventional control system design. I show that my research is positioned between model-based and data-driven methods. Next, some previous studies that inspired the proposed design method are introduced. Finally, based on the above background and previous studies, I formulated the problem setting of the proposed method in a general way.
- In Chapter 3 and in Chapter 4, the effectiveness of the proposed method is demonstrated by applying it to an actual control system. I reformulated the problem as an optimization problem for the cases where the plant is linear or nonlinear, and the dynamic quantizer is a linear or switching-type in Chapter 3 and Chapter 4, respectively. Numerical experiments show that the proposed method can design a quantizer that satisfies the control performance and the possibility of faster optimization by the proposed method.
- In Chapter 5 and in Chapter 6, the proposed method was redefined from the framework of control engineering to a broader meaning and was applied to a system with a fixed structure, TDA. In Chapter 5, I designed a quantization process to convert a gray-scale image to a binary image when the TDA process for a binary image is fixed as the image segmentation. Numerical experiments confirmed that the quantization process gave better segmentation results than the simple thresholding method. In Chapter 6, I designed a quantization process to convert original NNs to quantized NNs when the TDA process is fixed

as an evaluation method of quantized NNs. Since I have only been able to discuss how to fix the TDA as a numerical experiment, the actual design of the quantization process is one of future works.

In this thesis, I have established a new system design method based on model tuning. The result gives us a important insight. It is the proposal of a new model called the design model. Some previous studies have considered models that correspond to design models for individual problems. Based on the studies, the design model is formulated as a general problem setting in this study. Unlike conventional models that reflect the dynamics and characteristics of the object, the design model reflects the designer's intention in terms of fixing the structure of the controller. The new idea is that what is needed in control is not a perfect model but only one that can be used for control. This concept opens up new horizons in control and is an essential idea for future research in control. In particular, large and complex systems are known to be difficult to model, but the design model is expected to be helpful in the design of control systems for such systems.

I conclude this thesis by indicating some open problems as follows: The first is to propose a method for analytically constructing a design model. In Chapter 3 and in Chapter 4, I formulated optimization problems for constructing design models and obtained solutions by numerical optimization, but I did not show how to construct them analytically. Therefore, showing the exact optimality and convergence of these solutions is an essential issue for future work. By showing these analytical results, the effectiveness of the proposed method can be more quantitatively demonstrated, and the validity of the solutions can be guaranteed. The second is the application to fields other than control engineering. In this study, the proposed method is presented using control system design as an example, but it could also be applied to other fields. For example, in meta-heuristics optimization problems, an efficient search is possible by including a certain structure in the parameter update law. In addition, Chapter 5 and in Chapter 6 show applications to systems involving TDA, but these applications are still elementary. Applying the design model concept to general systems is also an important issue.

Bibliography

[1] Toshiyuki Kitamori. A method of control system design based upon partial knowledge about controlled processes. *Transactions of the Society of Instrument and Control Engineers*, Vol. 15, No. 4, pp. 549–555, 1979 (In Japanese).

[2] Kana Shikada, Noboru Sebe, and Masayuki Sato. Selection of the estimation model for robust observer-based controller against plant uncertainties. *IFAC-PapersOnLine*, Vol. 55, No. 25, pp. 193–198, 2022.

[3] Kana Shikada, Noboru Sebe, and Masayuki Sato. Robust observer-based controller against plant uncertainties. In *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1196–1201. IEEE, 2021.

[4] Hiroshi Okajima. Analysis of robust performance of model error compensator for polytopic-type uncertain continuous-time linear time invariant systems. *Transactions of the Society of Instrument and Control Engineers*, Vol. 55, No. 12, pp. 800–807, 2019 (In Japanese).

[5] Nobutaka Wada and Seiya Tsurushima. Constrained mpc to track time-varying reference signals: Online optimization of virtual reference signals and controller states. *IEEJ Transactions on Electrical and Electronic Engineering*, Vol. 11, pp. S65–S74, 2016.

[6] Nobutaka Wada. On the tracking controller synthesis in model predictive control. *SYSTEMS, CONTROL AND INFORMATION*, Vol. 61, No. 2, pp. 57–62, 2017 (In Japanese).

[7] Y Minami and K Kashima. Dynamic quantizer design based on serial system decomposition. In *Proceedings of the 22nd International Symposium on Mathematical Theory of Networks and Systems*, pp. 577–579, 2016.

[8] Y Minami and K Kashima. Dynamic quantizer design based on serial system decomposition. *Transactions of the Society of Instrument and Control Engineers*, Vol. 52, No. 1, pp. 46–51, 2016 (In Japanese).

[9] Taiki Kusui, Yuki Minami, and Masato Ishikawa. Stable dynamic quantizer

design for mimo non-minimum phase systems based on serial system decomposition. In *2019 18th European Control Conference (ECC)*, pp. 3704–3709. IEEE, 2019.

[10] Karl Johan Åström and Richard Murray. *Feedback systems: an introduction for scientists and engineers.* Princeton university press, 2021.

[11] Jan M Maclejowski. *Predictive control with constraints.* Prentice Hall, 2001.

[12] Max Schwenzer, Muzaffer Ay, Thomas Bergs, and Dirk Abel. Review on model predictive control: An engineering perspective. *The International Journal of Advanced Manufacturing Technology*, Vol. 117, No. 5, pp. 1327–1349, 2021.

[13] Kemin Zhou, John C Doyle, and Keith Glover. *Robust and optimal control.* Prentice Hall, 1995.

[14] Peter Dorato. A historical review of robust control. *IEEE Control Systems Magazine*, Vol. 7, No. 2, pp. 44–47, 1987.

[15] Lennart Ljung. *System identification: theory for the user.* Prentice Hall, 1999.

[16] Karl Johan Åström and Peter Eykhoff. System identification—a survey. *Automatica*, Vol. 7, No. 2, pp. 123–162, 1971.

[17] Li Fu and Pengfei Li. The research survey of system identification method. In *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Vol. 2, pp. 397–401. IEEE, 2013.

[18] Emre Sariyildiz and Kouhei Ohnishi. A guide to design disturbance observer. *Journal of Dynamic Systems, Measurement, and Control*, Vol. 136, No. 2, p. 021011, 2014.

[19] Kiyoshi Ohnishi, Kouhei Ohnishi, and Kunio Miyachi. Torque-speed regulation of dc motor based on load torque estimation method. In *JIEE/1983 International Power Electronics Conference*, 1983.

[20] Wen-Hua Chen, Jun Yang, Lei Guo, and Shihua Li. Disturbance-observer-based control and related methods—an overview. *IEEE Transactions on industrial electronics*, Vol. 63, No. 2, pp. 1083–1095, 2015.

[21] Arsalan Alavi, Mohammad Dolatabadi, Javad Mashhadi, and Ehsan Noroozinejad Farsangi. Simultaneous optimization approach for combined control–structural design versus the conventional sequential optimization method. *Structural and Multidisciplinary Optimization*, Vol. 63, No. 3, pp. 1367–1383, 2021.

[22] Ui-Jin Jung and Gyung-Jin Park. A new method for simultaneous optimum design of structural and control systems. *Computers & Structures*, Vol. 160, pp. 90–99, 2015.

[23] Christos S Patilas and Ioannis K Kookos. A novel approach to the simultaneous

design & control problem. *Chemical Engineering Science*, Vol. 240, p. 116637, 2021.

[24] Hideyuki Tanaka and Toshiharu Sugie. Integrated design of structure and control systems. *Journal of The Society of Instrument and Control Engineers*, Vol. 40, No. 6, pp. 448–453, 2001 (In Japanese).

[25] John G Ziegler and Nathaniel B Nichols. Optimum settings for automatic controllers. *Transactions of the American society of mechanical engineers*, Vol. 64, No. 8, pp. 759–765, 1942.

[26] Karl Johan Åström and Tore Hägglund. Revisiting the ziegler–nichols step response method for pid control. *Journal of process control*, Vol. 14, No. 6, pp. 635–650, 2004.

[27] Rakesh P Borase, DK Maghade, SY Sondkar, and SN Pawar. A review of pid control, tuning methods and applications. *International Journal of Dynamics and Control*, Vol. 9, pp. 818–827, 2021.

[28] Osamu Kaneko. Data-driven controller tuning: FRIT approach. *IFAC Proceedings Volumes*, Vol. 46, No. 11, pp. 326–336, 2013.

[29] Shotaro Soma, Osamu Kaneko, and Takao Fujii. A new method of controller parameter tuning based on input-output data–fictitious reference iterative tuning (FRIT)–. *IFAC Proceedings Volumes*, Vol. 37, No. 12, pp. 789–794, 2004.

[30] Osamu Kaneko. Parameter tuning of a controller based on the direct use of the data. *Journal of The Society of Instrument and Control Engineers*, Vol. 47, No. 11, pp. 903–908, 2008 (In Japanese).

[31] Osamu Kaneko. Direct data-driven controller tuning: FRIT approach. *Journal of The Society of Instrument and Control Engineers*, Vol. 52, No. 10, pp. 853–859, 2013 (In Japanese).

[32] Zhong-Ping Jiang, Tao Bian, Weinan Gao, et al. Learning-based control: A tutorial and some recent results. *Foundations and Trends® in Systems and Control*, Vol. 8, No. 3, pp. 176–284, 2020.

[33] James Ferlez and Yasser Shoukry. Aren: assured relu nn architecture for model predictive control of lti systems. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, pp. 1–11, 2020.

[34] Qikun Shen, Peng Shi, Junwu Zhu, Shuoyu Wang, and Yan Shi. Neural networks-based distributed adaptive control of nonlinear multiagent systems. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 3, pp. 1010–1021, 2019.

[35] Sangjae Bae, Dhruv Saxena, Alireza Nakhaei, Chiho Choi, Kikuo Fujimura, and

Scott Moura. Cooperation-aware lane change maneuver in dense traffic based on model predictive control with recurrent neural network. In *2020 American Control Conference (ACC)*, pp. 1209–1216. IEEE, 2020.

[36] Shankar Sastry and Marc Bodson. *Adaptive control: stability, convergence and robustness*. Courier Corporation, 2011.

[37] Gang Tao. Multivariable adaptive control: A survey. *Automatica*, Vol. 50, No. 11, pp. 2737–2764, 2014.

[38] Hiromitsu Ohmori. Structures and several methods of adaptive control systems. *Journal of The Society of Instrument and Control Engineers*, Vol. 48, No. 8, pp. 591–598, 2009 (In Japanese).

[39] Taichi Ikezaki and Osamu Kaneko. A new approach of data-driven controller tuning method by using virtual imc structure—virtual internal model tuning—. *IFAC-PapersOnLine*, Vol. 52, No. 29, pp. 344–349, 2019.

[40] Taichi Ikezaki and Osamu Kaneko. Virtual internal model tuning for cascade control systems. *SICE Journal of Control, Measurement, and System Integration*, Vol. 16, No. 1, pp. 55–62, 2023.

[41] Taichi Ikezaki and Osamu Kaneko. A new approach to parameter tuning of controllers by using output data of closed loop system. *IEEJ Transactions on Electronics, Information and Systems*, Vol. 139, No. 7, pp. 780–785, 2019 (In Japanese).

[42] Yusuke Fujimoto and Yuki Minami. Design of dynamic quantizer directly from input-output data. In *2024 SICE Annual Conference*, pp. 258–262, 2024.

[43] Shun-ichi Azuma and Toshiharu Sugie. Synthesis of optimal dynamic quantizers for discrete-valued input control. *IEEE Transactions on Automatic Control*, Vol. 53, No. 9, pp. 2064–2075, 2008.

[44] Masakazu Koike, Yuichi Chida, and Yuichi Ikeda. Control of a pneumatic isolation table including non-linear quantizer. *Transactions of the Society of Instrument and Control Engineers*, Vol. 49, No. 4, pp. 488–496, 2013 (In Japanese).

[45] Yuichi Chida, Nijihiko Ishihara, and Masaya Tanemura. Multirate and model predictive control of a pneumatic isolation table with a discrete actuator. *IFAC-PapersOnLine*, Vol. 52, No. 15, pp. 442–447, 2019.

[46] Yasuhiro Sugimoto, Keisuke Naniwa, Daisuke Nakanishi, and Koichi Osuka. Tension control of a mckibben pneumatic actuator using a dynamic quantizer. *Journal of Robotics and Mechatronics*, Vol. 35, No. 4, pp. 1038–1046, 2023.

[47] Hiroshi Okajima, Kenji Sawada, and Nobutomo Matsunaga. Dynamic quantizer design under communication rate constraints. *IEEE Transactions on Automatic*

Control, Vol. 61, No. 10, pp. 3190–3196, 2015.

[48] John Baillieul and Panos J Antsaklis. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, Vol. 95, No. 1, pp. 9–28, 2007.

[49] Graham C Goodwin, Eduardo I Silva, and Daniel E Quevedo. A brief introduction to the analysis and design of networked control systems. In *2008 Chinese Control and Decision Conference*, pp. 1–13. IEEE, 2008.

[50] Takao Nishiumi, Hiroshi Katoh, and Takayoshi Ichiyanagi. Application of the neural network for a hydraulic motor/load system. *Transactions of the Japan Society of Mechanical Engineers Series C*, Vol. 70, No. 695, pp. 2034–2041, 2004 (In Japanese).

[51] Jyunki Sato, Takahiro Kanno, Takanori Fukao, Ryohhei Takada, and Yasuyoshi Yokokohji. Trajectory control of hydraulic actuator systems using feedback modulator with unequally quantized inputs and nonlinear element model of the actuator. *Journal of the Robotics Society of Japan*, Vol. 31, No. 7, pp. 669–675, 2013 (In Japanese).

[52] Kodai Umeda, Tomoki Sakuma, Kenta Tsuda, Sho Sakaino, and Toshiaki Tsuji. Reaction force estimation of electro-hydrostatic actuator using reaction force observer. *IEEJ Journal of Industry Applications*, Vol. 7, No. 3, pp. 250–258, 2018.

[53] Shun-ichi Azuma and Toshiharu Sugie. Synthesis of optimal dynamic quantizers for symbolic input control. *Transactions of the Institute of Systems, Control and Information Engineers*, Vol. 20, No. 3, pp. 122–129, 2007 (In Japanese).

[54] Shun-ichi Azuma and Toshiharu Sugie. Optimal dynamic quantizers for discrete-valued input control. *Automatica*, Vol. 44, No. 2, pp. 396–406, 2008.

[55] Yuki Minami, Shun-ichi Azuma, and Toshiharu Sugie. Optimal dynamic quantizers in discrete-valued input feedback control systems. *Transactions of the Society of Instrument and Control Engineers*, Vol. 43, No. 3, pp. 227–233, 2007 (In Japanese).

[56] Shun-ichi Azuma, Yuki Minami, and Toshiharu Sugie. Optimal dynamic quantizers for feedback control with discrete-level actuators: unified solution and experimental evaluation. *Journal of Dynamic Systems, Measurement, and Control*, 2011.

[57] Shuichi Ohno, Yuma Ishihara, and Masaaki Nagahara. Min–max design of error feedback quantizers without overloading. *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 65, No. 4, pp. 1395–1405, 2017.

[58] Shuichi Ohno and M Rizwan Tariq. Optimization of noise shaping filter for quantizer with error feedback. *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 64, No. 4, pp. 918–930, 2016.

[59] Kenji Sawada and Seiichi Shin. Numerical optimization design of dynamic quantizer via matrix uncertainty approach. *Mathematical Problems in Engineering*, Vol. 2013, No. 1, p. 250683, 2013.

[60] Masato Ishikawa, Ichiro Maruta, and Toshiharu Sugie. Compensation of actuator nonlinearity using discrete-valued input control based on feedback modulator. *Transactions of the Society of Instrument and Control Engineers*, Vol. 44, No. 3, pp. 288–290, 2008 (In Japanese).

[61] Masato Ishikawa, Ichiro Maruta, and Toshiharu Sugie. Practical controller design for discrete-valued input systems using feedback modulators. In *2007 European Control Conference (ECC)*, pp. 3269–3274. IEEE, 2007.

[62] Shun-ishi Azuma and Toshiharu Sugie. Stability of optimal dynamic quantizers for discrete-valued input control. *Transactions of the Society of Instrument and Control Engineers*, Vol. 43, No. 12, pp. 1136–1143, 2007 (In Japanese).

[63] Shun-Ichi Azuma and Toshiharu Sugie. Stability analysis of optimally quantised lft-feedback systems. *International Journal of Control*, Vol. 83, No. 6, pp. 1125–1135, 2010.

[64] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, Vol. 9, No. 2, pp. 159–195, 2001.

[65] Nikolaus Hansen and Anne Auger. Principled design of continuous stochastic search: From theory to practice. In *Theory and principled methods for the design of metaheuristics*, pp. 145–180. Springer, 2013.

[66] Yugo Ogio, Yuki Minami, and Masato Ishikawa. A design method of nominal models for uncertain systems. *Proceedings of the Annual Conference of the Institute of Systems, Control and Information Engineers*, Vol. 65, pp. 781–784, 2021 (In Japanese).

[67] Yugo Ogio, Minami Yuki, and Ishikawa Masato. A design method of nominal models for uncertain systems. In *2021 SICE Annual Conference*, pp. 1190–1192, 2021.

[68] Girish N Nair, Fabio Fagnani, Sandro Zampieri, and Robin J Evans. Feedback control under data rate constraints: An overview. *Proceedings of the IEEE*, Vol. 95, No. 1, pp. 108–137, 2007.

[69] Mahmoud Abdelrahim, Victor Sebastiaan Dolk, and WPMH Heemels. Event-

triggered quantized control for input-to-state stabilization of linear systems with distributed output sensors. *IEEE Transactions on Automatic Control*, Vol. 64, No. 12, pp. 4952–4967, 2019.

[70] Yoav Sharon and Daniel Liberzon. Input to state stabilizing controller for systems with coarse quantization. *IEEE Transactions on Automatic Control*, Vol. 57, No. 4, pp. 830–844, 2011.

[71] Kenji Sawada and Seiichi Shin. Synthesis of continuous-time dynamic quantizers for lft type quantized feedback systems. *Artificial Life and Robotics*, Vol. 18, pp. 117–126, 2013.

[72] Shun-ichi Azuma and Toshiharu Sugie. Dynamic quantization of nonlinear control systems. *IEEE Transactions on Automatic Control*, Vol. 57, No. 4, pp. 875–888, 2011.

[73] Hai Lin and Panos J Antsaklis. Stability and stabilizability of switched linear systems: a survey of recent results. *IEEE Transactions on Automatic control*, Vol. 54, No. 2, pp. 308–322, 2009.

[74] Roger W Brockett, et al. Asymptotic stability and feedback stabilization. *Differential geometric control theory*, Vol. 27, No. 1, pp. 181–191, 1983.

[75] Yugo Ogio, Yuki Minami, and Masato Ishikawa. Design of switching-type dynamic quantizers for continuous-time nonlinear systems. *IFAC-PapersOnLine*, Vol. 56, No. 2, pp. 3904–3909, 2023.

[76] Yugo Ogio, Yuki Minami, and Masato Ishikawa. A model-tuning approach to sampled-data dynamic quantizer design. *Transactions of the Society of Instrument and Control Engineers*, Vol. 59, No. 12, pp. 542–549, 2023 (In Japanese).

[77] Karl Johan Åström and Katsuhisa Furuta. Swinging up a pendulum by energy control. *Automatica*, Vol. 36, No. 2, pp. 287–295, 2000.

[78] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, Vol. 4, pp. 1942–1948. ieee, 1995.

[79] Thu Ha Nguyen, Dang Quang Bui, Phuong Nam Dao, et al. An efficient min/max robust model predictive control for nonlinear discrete-time systems with dynamic disturbance. *Chaos, Solitons & Fractals*, Vol. 180, p. 114551, 2024.

[80] Marcelo Alves dos Santos, Antonio Ferramosca, and Guilherme Vianna Raffo. Set-point tracking mpc with avoidance features. *Automatica*, Vol. 159, p. 111390, 2024.

[81] Hoang Nguyen, Hoang Bach Dang, and Phuong Nam Dao. On-policy and

off-policy q-learning strategies for spacecraft systems: An approach for time-varying discrete-time without controllability assumption of augmented system. *Aerospace Science and Technology*, Vol. 146, p. 108972, 2024.

[82] Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, Vol. 113, No. 26, pp. 7035–7040, 2016.

[83] Takashi Ichinomiya, Ippei Obayashi, and Yasuaki Hiraoka. Persistent homology analysis of craze formation. *Physical Review E*, Vol. 95, No. 1, p. 012504, 2017.

[84] Joseph Minhow Chan, Gunnar Carlsson, and Raul Rabadan. Topology of viral evolution. *Proceedings of the National Academy of Sciences*, Vol. 110, No. 46, pp. 18566–18571, 2013.

[85] Yuhei Umeda. Time series classification via topological data analysis. *Information and Media Technologies*, Vol. 12, pp. 228–239, 2017.

[86] Atsushi Saito. Grayscale image processing and segmentation. *Medical Imaging Technology*, Vol. 35, No. 1, pp. 3–10, 2017 (In Japanese).

[87] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, pp. 62–66, 1979.

[88] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. *Advances in neural information processing systems*, Vol. 32, , 2019.

[89] James R Clough, Ilkay Oksuz, Nicholas Byrne, Julia A Schnabel, and Andrew P King. Explicit topological priors for deep-learning based image segmentation using persistent homology. In *International Conference on Information Processing in Medical Imaging*, pp. 16–28. Springer, 2019.

[90] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, Vol. 46, No. 2, pp. 255–308, 2009.

[91] Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, Vol. 5, No. 1, pp. 501–532, 2018.

[92] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, Vol. 4, p. 667963, 2021.

[93] Marcio Gameiro, Yasuaki Hiraoka, Shunsuke Izumi, Miroslav Kramar, Konstantin Mischaikow, and Vidit Nanda. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics*, Vol. 32, pp. 1–17, 2015.

[94] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

[95] Satoru Watanabe and Hayato Yamana. Topological measurement of deep neural networks using persistent homology. *Annals of Mathematics and Artificial Intelligence*, Vol. 90, No. 1, pp. 75–92, 2022.

[96] Herbert Edelsbrunner. Persistent homology: theory and practice. 2013.

[97] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & computational geometry*, Vol. 28, pp. 511–533, 2002.

[98] Ippei Obayashi, Yasuaki Hiraoka, and Masao Kimura. Persistence diagrams with linear machine learning models. *Journal of Applied and Computational Topology*, Vol. 1, pp. 421–449, 2018.

[99] Ippei Obayashi. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. *SIAM Journal on Applied Algebra and Geometry*, Vol. 2, No. 4, pp. 508–534, 2018.

[100] Ryosuke Morita. Signal digitization: Noise-shaping quantization—VI. *SYSTEMS, CONTROL AND INFORMATION*, Vol. 61, No. 12, pp. 512–517, 2017 (In Japanese).

[101] Yu-Jin Zhang. A survey on evaluation methods for image segmentation. *Pattern recognition*, Vol. 29, No. 8, pp. 1335–1346, 1996.

[102] Tom Eelbode, Jeroen Bertels, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops, and Matthew B Blaschko. Optimization for medical image segmentation: theory and practice when evaluating with dice score or jaccard index. *IEEE transactions on medical imaging*, Vol. 39, No. 11, pp. 3679–3690, 2020.

[103] Koichi Osuka. Source of various behaviors of living things that understands from zombification of insects. In *Proceedings of the Conference of Transdisciplinary Federation of Science and Technology*, pp. D–2. Transdisciplinary Federation of Science and Technology, 2017 (In Japanese).

[104] Yuki Minami. Signal digitization: Noise-shaping quantization − V. *SYSTEMS, CONTROL AND INFORMATION*, Vol. 61, No. 10, pp. 428–433, 2017 (In Japanese).

[105] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, Vol. 1, No. 4, pp. 321–331, 1988.

[106] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal*

of computational physics, Vol. 79, No. 1, pp. 12–49, 1988.

[107] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Algorithm compression and hardware acceleration for neural networks: A comprehensive survey.

[108] Babak Rokh, Ali Azarpeyvand, and Alireza Khanteymoori. A comprehensive survey on model quantization for deep neural networks in image classification. *ACM Transactions on Intelligent Systems and Technology*, Vol. 14, No. 6, pp. 1–50, 2023.

[109] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pp. 291–326. Chapman and Hall/CRC, 2022.

[110] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, Vol. 18, No. 187, pp. 1–30, 2018.

[111] Sangyun Oh, Hyeonuk Sim, Jounghyun Kim, and Jongeun Lee. Non-uniform step size quantization for accurate post-training quantization. In *European Conference on Computer Vision*, pp. 658–673. Springer, 2022.

[112] Jun Fang, Ali Shafiee, Hamzah Abdel-Aziz, David Thorsley, Georgios Georgiadis, and Joseph H Hassoun. Post-training piecewise linear quantization for deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 69–86. Springer, 2020.

[113] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, Vol. 110, No. 11, pp. 3245–3262, 2021.

[114] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems*, Vol. 32, , 2019.

[115] Naoki Tsubone, Minami Yuki, and Ishikawa Masato. Noise-shaping quantization utilizing training data for compaction of neural networks. In *2023 SICE Annual Conference*, pp. 1065–1067, 2023.

[116] Naoki Tsubone, Yuki Minami, and Masato Ishikawa. Noise-shaping quantization utilizing training data for compaction of neural networks. *Proceedings of the Annual Conference of the Institute of Systems, Control and Information Engineers*, Vol. 67, pp. 32–37, 2023 (In Japanese).

[117] Yuki Minami, Tomohiro Ikeda, and Masato Ishikawa. Design of noise shaping quantizers for data compaction of neural networks. *Transactions of the Society of Instrument and Control Engineers*, Vol. 56, No. 9, pp. 425–431, 2020 (In Japanese).

[118] Thomas Parisini and Riccardo Zoppoli. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, Vol. 31, No. 10, pp. 1443–1451, 1995.

[119] Benjamin Karg and Sergio Lucia. Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Transactions on Cybernetics*, Vol. 50, No. 9, pp. 3866–3878, 2020.

[120] Satoru Watanabe and Hayato Yamana. Overfitting measurement of convolutional neural networks using trained network weights. *International Journal of Data Science and Analytics*, Vol. 14, No. 3, pp. 261–278, 2022.

[121] Paolo Masulli and Alessandro EP Villa. The topology of the directed clique complex as a network invariant. *SpringerPlus*, Vol. 5, pp. 1–12, 2016.

[122] Yann LeCun, Corinna Cortes, and Christopher JC Burges. Mnist handwritten digit database. `http://yann.lecun.com/exdb/mnist/`.

[123] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision/arXiv*, Vol. 1311, 2014.

Related papers

The results of this thesis is published in the following articles.

Chapter 3

[1] 荻尾優吾，南裕樹，石川将人，「モデル探索アプローチによるサンプル値駆動型動的量子化器の設計」，計測自動制御学会論文集，Vol. 59，No. 12，pp. 542–549，2023

[2] Yugo Ogio, Yuki Minami, Masato Ishikawa, “A design method of nominal models for uncertain systems.” SICE Annual Conference (SICE AC 2021), Online, Japan, Sep., 2021

[3] Yugo Ogio, Yuki Minami, Masato Ishikawa, “A controller design method for continuous-time and discrete-valued systems.” SICE Annual Conference (SICE AC 2022), Kumamoto, Japan, Sep., 2022

[4] 荻尾優吾，南裕樹，石川将人，「コントローラ設計に忖度するモデル構築」，第 63 回自動制御連合講演会，オンライン，11 月，2020

[5] 荻尾優吾，南裕樹，石川将人，「不確かさを有するシステムに対する設計モデルの一構成法」，第 65 回システム制御情報学会研究発表講演会（SCI’21），オンライン，5 月，2021

[6] 荻尾優吾，南裕樹，石川将人，「離散値係数制御器の設計のための設計モデル」，第 66 回システム制御情報学会研究発表講演会（SCI’22），京都，5 月，2022

Chapter 4

[1] Yugo Ogio, Yuki Minami, Masato Ishikawa, “Design of switching-type dynamic quantizers for continuous-time nonlinear systems.” IFAC World Congress, Yokohama, Japan, Jul., 2023

[2] Yugo Ogio, Yuki Minami, Masato Ishikawa, “A model-tuning approach to switching-type dynamic quantizer design for nonlinear systems” International

Journal of Control, Automation and Systems, Submitted

Chapter 5

[1] 荻尾優吾，南裕樹，石川将人，「位相的データ解析に基づく濃淡画像のセグメンテーション」，システム制御情報学会論文誌，Vol. 34，No. 9，pp. 243–250，2021

[2] 荻尾優吾，南裕樹，石川将人，「位相的データ解析を用いたコオロギの脳 CT 画像のセグメンテーション」，第 62 回自動制御連合講演会，北海道，11 月，2019

Chapter 6

[1] Yugo Ogio, Naoki Tsubone, Yuki Minami, Masato Ishikawa, “A TDA-based performance analysis for neural networks with low-bit weights.” Artificial Life and Robotics, Accepted

[2] Yugo Ogio, Naoki Tsubone, Yuki Minami, Masato Ishikawa, “A TDA-based performance analysis for quantized neural networks.” AROB 29th 2024, Beppu, Japan, Jan., 2024

Publication list

Journal papers

[1] 荻尾優吾，南裕樹，石川将人，「位相的データ解析に基づく濃淡画像のセグメンテーション」，システム制御情報学会論文誌，Vol. 34，No. 9，pp. 243–250，2021

[2] 荻尾優吾，南裕樹，石川将人，「モデル探索アプローチによるサンプル値駆動型動的量子化器の設計」，計測自動制御学会論文集，Vol. 59，No. 12，pp. 542–549，2023

[3] Yugo Ogio, Naoki Tsubone, Yuki Minami, Masato Ishikawa, "A TDA-based performance analysis for neural networks with low-bit weights." Artificial Life and Robotics, Accepted

[4] Yugo Ogio, Yuki Minami, Masato Ishikawa, "A model-tuning approach to switching-type dynamic quantizer design for nonlinear systems" International Journal of Control, Automation and Systems, Submitted

Refereed international conference papers

[1] Yugo Ogio, Yuki Minami, Masato Ishikawa, "A design method of nominal models for uncertain systems." SICE Annual Conference (SICE AC 2021), Online, Japan, Sep., 2021

[2] Yugo Ogio, Yuki Minami, Masato Ishikawa, "A controller design method for continuous-time and discrete-valued systems." SICE Annual Conference (SICE AC 2022), Kumamoto, Japan, Sep., 2022

[3] Yugo Ogio, Yuki Minami, Masato Ishikawa, "Design of switching-type dynamic quantizers for continuous-time nonlinear systems." IFAC World Congress, Yokohama, Japan, Jul., 2023

[4] Yugo Ogio, Naoki Tsubone, Yuki Minami, Masato Ishikawa, "A TDA-based performance analysis for quantized neural networks." AROB 29th 2024, Beppu, Japan, Jan., 2024

Domestic conference papers

[1] 荻尾優吾，南裕樹，石川将人，「位相的データ解析を用いたコオロギの脳 CT 画像のセグメンテーション」，第 62 回自動制御連合講演会，北海道，11 月，2019

[2] 荻尾優吾，南裕樹，石川将人，「コントローラ設計に忖度するモデル構築」，第 63 回自動制御連合講演会，オンライン，11 月，2020

[3] 荻尾優吾，南裕樹，石川将人，「不確かさを有するシステムに対する設計モデルの一構成法」，第 65 回システム制御情報学会研究発表講演会（SCI’21），オンライン，5 月，2021

[4] 荻尾優吾，南裕樹，石川将人，「離散値係数制御器の設計のための設計モデル」，第 66 回システム制御情報学会研究発表講演会（SCI’22），京都，5 月，2022

Honors and awards

[1] 2019 年度大阪大学工学賞受賞

[2] 2021 年度日本設計工学会武藤栄次賞優秀学生賞受賞

[3] 第 66 回システム制御情報学会研究発表講演会 SCI 学生発表賞受賞

[4] 計測自動制御学会関西支部 2024 年度支部長賞受賞

Research grants

[1] 立石科学技術振興財団研究助成 (C) (2022/4 - 2025/3)

[2] 次世代挑戦的研究者育成プロジェクト (2022/4 - 2024/3)

[3] 日本学術振興会 特別研究員 (DC2) (2024/4 - 2025/3)