Kinesthetic Data Reduction Techniques in Bilateral Teleoperation Systems

Deng, Qifang

Awarding institution:
King's College London

Kinesthetic Data Reduction Techniques in Bilateral Teleoperation Systems

Qifang Deng

Supervisor: Prof. A. Hamid Aghvami
Prof. Toktam Mahmoodi

Department of Engineering
King's College London

This thesis is submitted for the degree of
Doctor of Philosophy

June 2023

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Qifang Deng
June 2023

Acknowledgements

I would like to express my heartfelt gratitude to the following people for their invaluable contribution to my journey towards the completion of this thesis:

First and foremost, I would like to thank my supervisors, Prof. Hamid and Prof. Toktam for their invaluable advice, continuous supports, and patience during my PhD study. Thanks Hamid for providing me with life advice and sharing your successful experience. Also, thanks for your patient guidance in my papers writing from when I was a new PhD student to now. Thanks Toktam for your comments on my papers. You are always timely to provide comments on my papers and make revisions. It is your unwavering commitment to academic excellence that inspired me to strive for the highest standards of scholarship. I am indebted to you for your generosity, kindness, and patience throughout this challenging journey.

I would really appreciate my parents, grandparents, for their unwavering support, encouragement, and belief in my abilities. You are so enlightened that I can choose my majors in college and my future careers. Your love and understanding have been a source of strength and inspiration throughout my journey. I am grateful for your sacrifices, understanding, and constant encouragement. Without their support, I would not have been able to complete this journey.

I would also like to acknowledge the faculty and staff of the Engineering Department in King's College London, for providing me with the resources and opportunities to pursue my PhD. Their dedication to academic excellence has been a constant source of inspiration and motivation for me. I am grateful to them for their invaluable support, advice, and encouragement.

I would also like to express my gratitude to my girlfriend Yiki, who keeps encouraging me to graduate smoothly. Thank you for the wonderful memories you give me in this country. Every time I felt lost or got stuck in researches, you could always find something interesting or some places to help me relax. It is your encouragements and belief to me that I can finish my PhD thesis. It is you that make my PhD life bright and unforgettable.

My job advisor, Leilei Zhu, have been an integral part of my journey.Thanks for my career guidances during these three years. Thank you for appreciating my programming and data analysis skills, and I can be a part-time NHS employee. You took me to meet

numerous excellent people, and helped me with my spoken English in English-speaking work environment. I am grateful for your camaraderie, intellectual stimulation, and willingness to share your knowledge and expertise. Your feedback and suggestions have been immensely helpful in refining my research.

Thanks Wendy for your English guidances on this thesis. You are the best writer who I know. Thanks for helping me with proofreading this thesis. You always give me useful advice, such as shortening a long sentence to several shorter sentences, replacing words to a more elegant version. It is your encouragement I can this impossible task.

Last but not least, I would like to thank the study participants who generously gave their time and shared their experiences with me. Yunchen Song, Qiutong He, Dian Jing, Annie, Zhenhao Li, Xiang Li, Zhuoyue Sun, Zheng Ma. Thanks for contacting with me when I was under pressure. Also, thanks for creating emotional value in my spare time. Their contributions were invaluable in making this research possible, and I am honored to have had the opportunity to learn from them.

In conclusion, I am grateful to everyone who has contributed to my journey towards the completion of this thesis. Thank you all for being a part of my journey.

Abstract

The sense of touch, acting as a connection between humans and surroundings, is a stretch of audio and video information. People have been striving to pursue higher productivity and quality in the telecommunication system by proposing schemes combining with voice, capture, and sensory feedback since last decades. To achieve the transmission of force feedback from remote environment is thus the objective and research concentrates on bilateral teleoperation systems.

Traditional bilateral teleoperation systems are not only requiring a large amount of network resources, but sensitive to the transmission delay. From the previous research, the possible communication delay of a bilateral teleoperation system ranges from one to several hundred milliseconds. However, even a small communication delay or packet loss in the communication channel can affect the system's stability and transparency. Therefore, kinesthetic data reduction techniques are required in bilateral teleoperation systems. The current scheme to reduce the high-rate haptic data transmission employs a mathematical threshold to transmit data selectively based on human perceptual limitations, which is called perceptual deadband (PD)-based codecs. It describes the perceptual thresholds by pairwise comparison. However, the current perceptual threshold is not sufficiently accurate to describe some types of stimuli in practice, including kinesthetic perception, and human time perception. Moreover, pairwise comparison is required to be made in each collected kinesthetic data; this wastes memory and time.

Due to various aforementioned limitations of deploying PD-based codecs in real bilateral teleoperation systems, I believe that novel mathematical models controlling kinesthetic data transmission should be proposed, so that the transmission status of the newly collected data will be determined by the model directly, without any comparison. In this thesis, three different machine learning algorithms are used for kinesthetic data reduction over the haptic communication network. By comparing with conventional PD-based codecs for kinesthetic data reduction, proposed techniques perform better in different aspects. Such a system with kinesthetic data reduction techniques is shown in this thesis to reduce the kinesthetic data transmission effectively.

The first idea for reducing kinesthetic data transmission is, by deploying long-short term memory (LSTM)-based data reduction modules, to control the transmission status of

each data. Current PD-based codecs is not practical in dealing with time series data, as pairwise comparison is required before each transmission. Therefore, a novel mathematical model for deriving the the transmission status of each collected data is proposed based on LSTM networks. This model is trained from Weber's law of just noticeable difference (JND), in which explains humans' perceptual limitation.

The second idea is reducing the size of kinesthetic data in each transmission. Dimensionality reduction techniques (DRTs) are introduced to map original kinesthetic data in high dimensions to corresponding embeddings in low dimensions. This is novel, unprecedented in abandoning the concept of selective transmission (reducing the amount of data transmission), and applying DRTs on each collected kinesthetic data for reducing the network offload respectively. More specific, three different dimensionality reduction techniques, including principal component analysis (PCA), stacked auto-encoder(SAE) and uniform manifold approximation and projection (UMAP), are stated and compared with each other. Moreover, for reconstructing dimensions of kinesthetic data from low to original, three different data reconstruction techniques are used in terms of three dimensionality reduction techniques.

The third idea is clustering kinesthetic data with unsupervised learning techniques, by which realizing selectively transmitting kinesthetic data. Even though LSTM-based mathematical models can reduce the transmission of kinesthetic data effectively, labels of each data are deriving from existing PD-based codecs. In order to deal with original unlabelled kinesthetic data, unsupervised clustering techniques are introduced to classify each sample into different clusters in terms of intra-cluster similarities and inter-cluster distances. Unsupervised clusterings can find internal features of the dataset which may be ignored by humans. What's more, we select kinesthetic data in a part of clusterings to transmit over the network, and compare it with PD-based and LSTM-based kinethetic data reduction techniques.

We also improve the accuracy of prediction models in bilateral teleoperation systems with the assistance of gradient boosting decision tree (GBDT) algorithm. Prediction models are required since operator needs to estimate the force feedback when no data is received. Current PD-based predictive scheme assumes the future value is related to last one or two received data, which is too simple to describe this whole prediction. Therefore, GBDT, which is acknowledged as the most accurate and commonly used predictive algorithm is introduced to improve the whole accuracy and transparency of the system.

Table of contents

List of figures

List of tables

Symbols

Roman Symbols

σ	sigmoid function
F	force
K	optimal number of clusters
k	Weber Parameter
v	velocity
x	position

Greek Symbols

δ	viscosity
μ	coefficient of friction
Φ	selected curve function
Ψ	preferred curve function

Other Symbols

$\dot{x}$	derivative of x
$\dot{\vec{x}}$	derivative of vector x
A^T	transpose matrix of A
F^a	attractive force
F^r	repulsive force
$X_{m \times n}$	matrix with m rows and n columns

$\|\vec{x}\|$ norm of $\vec{x}$

$\mathbf{R}_N$ N dimensional space

$\sum$ summation

$\vec{x}$ vector x

Acronyms / Abbreviations

3D Three Dimensions

ACC Accuracy

AR Autoregressive

CHAI3D Computer Haptics and Active Interface

ChatGPT Chat Generative Pre-trained Transformer

CHI Calinski-Harabasz Index

CON Continuity

DBI Davies-Bouldin Index

DCT Discrete Cosine Transform

DoF Digree of Freedom

DRT Dimensionality Reduction Technique

FF Feedforward

FOLP First-Order Linear Predictor

GBDT Gradient Boosting Decision Tree

HPW-PSNR Haptic Perceptually Weighted Peak Signal-To-Noise Ratio

HPW Haptic Perceptually Weight

JND Just Noticeable Difference

k-NNs k-Nearest Neighbours

LSTM Long Short-Term Memory

MSE Mean Squared Error

NN Neural Network

OP Operator

PCA Principle Component Analysis

PD-based Perceptual Deadband-based

PSNR Peak Signal-To-Noise Ratio

RNN Recurrent Neural Network

RRE Relative Rank Error

SAE Stacked Auto-encoder

SC Silhouette Coefficient

SGD Stochastic Gradient Descent

TOP Teleoperator

TRU Trustworthiness

UMAP Uniform Manifold Approximation and Projection

WPT Wavelet Packet Transform

ZOH Zero-Order Hold

Chapter 1

Overview

1.1 Introduction

There is an endless attitude for people to pursue higher productivity and quality of experience in remote communication. In the past decades, humans have made great success in transmitting audio-visual signals. However, the sense of touch, acting as a connection between humans and their surroundings, is achievable beyond audio and video information. Therefore, there exists a prospect of building a communication system which proposes schemes combined with voice, capture, and sensory feedback. In reality, many haptic-related industries have emerged in various fields, including medical teleoperation, remote exploration, dive detection, etc., which enables human users to fully immerse into the remote environment and perform complicated tasks in a distance.

Thanks to these ambitions, achieving immersive experience becomes possible, it can provide users with comprehensive sensory feedback —— auditory, visual, and haptic. With increasing quality, the presence of users and feeling of togetherness are realized at the level of voice or video conferences. For instance, the prevalence of working from home and on video conference platforms during the pandemic, which offers users attend the meeting in a distance from workplace. While the feeling of being present in a remote environment is clearly available with these systems, a complete immersion cannot be realized without the possibility of physical interaction with the remote environment.

Attempting to fulfil the wish of transmitting the sense of feeling to the distance, a real-time teleoperation system for exchanging the action and reaction information between the master and the slave have been developed. In other words, a bilateral teleoperation system which supports simultaneous data transmission from master to slave side and slave to master side has been proposed. Unlike a bidirectional system which transmits data in both directions (send and receive) but not at the same time, a bilateral system allow data to be transmitted simultaneously. These bilateral teleoperation systems provide users with

multimodal sensor information about the remote environment, while allowing the remote robotics to interact with the environment by following users' movements. Furthermore, haptic data is normally divided into two categories, kinesthetic and tactile information. All the haptic data including position, velocity, force, torque, texture can be included in kinesthetic data. Simultaneously, data recording vibrations produced when touching on different surfaces, is termed tactile data. In this thesis, we use the word haptic to refer to the kinesthetic components of position/velocity and force.

Using a bilateral teleoperation system with haptic feedback, a complete immersion into a distant environment is possible for users. Users can execute tasks without being physically present but with the feeling of being there. From previous research, a bilateral teleoperation system consists normally of three different components, including the human operator side/master system (OP), the teleoperator side/slave system (TOP), and the communication link which connects the OP with the TOP [1]. A bilateral multimodal sensor information exchange between the master and slave system occurs during the teleoperation process.

Nevertheless, communication of haptic information for teleoperation systems imposes strong demands on the communication network. The sampling rate of haptic devices is normally 1 kHz or may be higher to maintain the transparency of the system. The collected data is packetized and transmitted at the same rate [2–4]. Therefore, for communicating kinesthetic information in teleoperation system, thousands of haptic data packets per second must be transmitted between the master and the slave devices.

This means that due to the consumption of a large amount of network resources, the possible communication delay of a bilateral teleoperation system ranges from one to several hundred milliseconds [2]. In practice, the communication delay depends on the geographic distance and the communication facility. The delay may reach several seconds in a long-distance communication system, such as space communication system [93]. The implications of this delay are not restricted to user frustration. As many studies have shown, even a small communication delay or packet loss in the communication channel can affect the system's stability and transparency [5]. As a result, techniques to reduce the transmission of haptic data or packets should be proposed in bilateral teleoperation systems.

The theoretical basis for data or packet reduction for teleoperation systems is Weber's Law of Just Noticeable Differences (JND). Weber's Law is based on human perceptual limitations. The so-called perceptual deadband based (PD-based) approach was first introduced in [6–9]. This law states that the JND between two stimuli, which is the minimum change in the magnitude of a stimulus that the human can be aware of, is ordinarily proportional to its magnitude. The JND model currently used describes the perceptual thresholds for pairwise comparison, which is resource-consuming in practice.

This thesis will present three novel haptic data reduction schemes which reduces the potential transmission rate more effectively relative to traditional PD-based codecs.

The first scheme proposes a novel mathematical model to control the transmission of time series of kinesthetic data. Recurrent neural networks (RNNs) with long short-term memory (LSTM) architecture are utilized to derive the mathematical model by labelling the JND. The second scheme reduces the packet size of haptic data during the transmission. Nearly all of the previous data reduction techniques strive to selectively transmit haptic data with super high dimensions. This thesis proposes a novel pipeline to solve the compression-recovery problem of haptic signals, and specifically compares three typical dimensionality reduction techniques (DRTs) to reduce the size of each transmitted data. The third scheme aims at clustering original unlabelled kinesthetic data in terms of unsupervised learning algorithms. Two different unsupervised clustering techniques, k-means clustering and hierarchical clustering, are introduced to classify kinesthetic data into different optimal clusters. Moreover, only data in a part of clusters will be transmitted over the network, which reduces the amount of data transmission effectively.

Three solutions to reduce the haptic data transmission by means of either deriving new mathematical models or DRTs are introduced and analysed in this thesis.

Furthermore, in order to fill in the non-transmitted sample on the TOP, predictive coding is proposed to estimate future haptic data from that previously received. The current predictive model is both simple and inefficient [27–29]. This thesis deploys the gradient boosting decision tree (GBDT) to predict the haptic data when the difference between newly collected sample and the one previously transmitted is imperceptible to the operator. In other words, a predictive sample is produced on the slave side once the newly collected sample on the master side is within the deadzone of the previous one. Comparisons between GBDT-based predictive and traditional schemes are discussed in this thesis.

Up to now, I have published one conference paper [95]. Meanwhile, one paper has been accepted by IEEE Internet of Things Journal with minor revisions [**?**], and the other one is still under review [97].

1.2 Bilateral Teleoperation Systems

This thesis presents a novel and feasible scheme with the deployment of two different haptic data reduction techniques to cope with the stability and transparency issue resulting from the transmission delay of the network. Bilateral systems with haptic data reduction modules are explained briefly below.

In tradition, there is a bilateral multimodal sensor information exchange between the master and slave system during the teleoperation process. Initialy, the local operator receives the command from the human, which produces a signal consisting of the velocity and position of the robot. Those signals are usually sampled in 1ms [2–4]. Later, when the signal transmits over the communication link, the slave robot follows the received position or velocity commands. Correspondingly, there will be some haptic feedback from the TOP side when the remote robot interacts with the environment. The haptic, visual, and audio signals captured from the slave side are sent back and displayed to the master. This type of position-force or velocity-force teleoperation system is proposed and widely used in [10–12]. Another architecture called the position-position teleoperation system is first proposed and used in [13]. As the name implies, the master sends its position signals to the slave, and it also receives the slave's position signals. In this type of teleoperation system, the main haptic feedback is the slave motion signals, which are rendered on the master side. In this thesis, we mainly consider the force feedback during the teleoperation process.

Normally, a number of studies over the past 2 decades have found that the optimal vibration frequency for haptics ranges from 100 to 300 Hz [14, 15]. For more complex haptic signals that include different types of tactile sensations, such as pressure or texture, the maximum frequency content can be in the range of 200 ro 400 Hz [14]. A 1 kHz sampling frequency is typically sufficient to capture the maximum frequency content of haptic signals. The Nyquist theorem states that the sampling rate should be at least twice the maximum frequency content of the signal to avoid aliasing. In the case of haptic signals, the Nyquist frequency would be half of the maximum frequency content, which is usually in the range of a few hundred Hz. Therefore, a sampling frequency of 1 kHz provides a sufficient margin for the Nyquist frequency to be well above the maximum frequency content of haptic signals. Besides, using a 1 kHz sampling frequency allows for the use of anti-aliasing filters. An anti-aliasing filter is a low-pass filter that removes any frequency content above the Nyquist frequency before the signal is sampled. This ensures that any frequency content above the Nyquist frequency does not create spurious signals in the digital signal. For a 1 kHz sampling frequency, the Nyquist frequency is 500 Hz. Therefore, an anti-aliasing filter can be used to remove any frequency content above 500 Hz.

The underlying principle of traditional bilateral teleoperation systems is depicted in Fig. 1.1. The OP applies a force of F_i on the master manipulator and moves it by X_i at the velocity of $\dot{X}_i$. These position and velocity measurements are transmitted to the TOP over a communication channel. The manipulator on the slave side follows the received position and velocity signals and interacts with the environment. The reaction force measurements produced during the interaction F_{xi} are fed back to the master.

The traditional bilateral teleoperation system not only requires a large amount of network resources, but is sensitive to transmission delay. Therefore, haptic data reduction, or packet rate reduction, is required in teleoperation systems. As argued above, the current scheme to reduce the high-rate haptic data transmission employs a mathematical threshold to transmit data selectively based on human perceptual limitations. As we have seen, Weber's Law states that the JND between two stimulus, which is the minimum change in the magnitude of a stimulus that the human can be aware of, is ordinarily proportional to its magnitude. The current JND model, also termed PD-based codecs, describes the perceptual thresholds by pairwise comparison. However, in practice, Weber's law cannot be applied when dealing with a time series of force or velocity values. The current JND perceptual threshold is not sufficiently accurate to describe some types of stimuli in practice, including kinesthetic perception, and human time perception [16–19]. Moreover, pairwise comparison is required to be made in each collected kinesthetic data; this wastes memory and time. In addition, there is no fault tolerance of current JND perceptual threshold and therefore once the comparison is wrong, the fault will remain.

Due to various aforementioned limitations of deploying PD-based codecs in real bilateral teleoperation systems, I believe that novel mathematical models controlling kinesthetic data transmission should be proposed, so that the transmission status of the newly collected data will be determined by the model directly, without any comparison. As long as the mathematical model is derived, the new haptic data produced on both master and slave side can swiftly determine whether or notit should be transmitted.

Since humans are living in a 3-dimensional world, each visible or sensible physical measurement can be represented by a 3-dimensional vector. Simultaneously, as is shown in 1.1, both position/velocity and force data is in high dimensions. Human operators control the master robot, and move it with a velocity of $\dot{x}_i$ at position x_i. The position/velocity data is transmitted to the slave side over the communication channel, and thus the received position and velocity data is x_{i_R} and $\dot{x}_{i_R}$. The slave follows received instructions, and interacts with the remote environment. Furthermore, corresponding force feedback data f_{xi} is transmitted back to human operators on the master side. Received force feedback can be represented as f_{xi_R}. In a position-force teleoperation architecture, the position/velocity data is typically represented with six dimensions, while the force data is three. However, the force data is a vector along three different axes (x, y, z). The magnitude of the frictional force in the tangential direction (along x&y axes) of a moving object is always proportional to the pressure it experiences in the vertical direction (along z axis). Inspired by this phenomenon, force data along z axis needs not to be transmitted as long as the frictional force in the tangential direction is given, and vice versa. Moreover, it allows us to extend this idea to position/velocity data. As is stated above, haptic data with full dimensions is collected, packetized, and transmitted at a rate of 1 kHz or higher. This is novel,

unprecedented in abandoning the concept of selective transmission (reducing the amount of data transmission), and applying DRTs on each collected kinesthetic data for reducing the network offload respectively. This DRT-based kinesthetic data transmission scenario will be analysed in this thesis. The simulation performance and feasibility will also be discussed. In addition, unsupervised clustering techniques are introduced to classify original kinesthetic data without labeling data before training process. Kinesthetic data is segregated into different clusterings, while data belonging to certain groups is transmitted over the network.

To accomplish a logical thread in deriving the fractional allocation strategies, the thesis has been organised as follows.

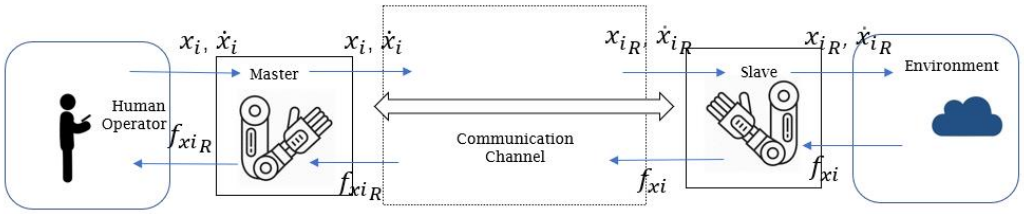


Fig. 1.1 Overview of a Conventional Bilateral Teleoperation System

1.3 Background & State of the Art

The bilateral teleoperation system was first realized by Goertz and his team in the 1940s, when they developed a mechanical pantograph introducing the concept of bilateral teleoperation. Since then, mechanical manipulators were built to manage the nuclear material of a nuclear reactor [20]. The objective of the bilateral teleoperation system is to provide a better haptic perception to the operator, while they perform a remote operation task. The generalisation of the concept of haptic communications which extended the audiovisual communications was first proposed in Feb. 2012 [8]. Apart from this, the 1ms-challenge in haptic communications was first given in [8], which prospected the overall time delay should be less than 1 ms in haptic communications.

Following this, numerous schemes have emerged aiming to accomplish the 1ms-challenge. The original scheme refers to haptic data reduction techniques. Kinesthetic data reduction techniques are mainly based on two approaches: statistical, and perceptual schemes [21].

The statistical scheme aims at using the statistics of the haptic signals to compress the packet size. For instance, for kinesthetic data of 1 degree of freedom (DoF), lossy data

compression and decompression methods have been achieved by using discrete cosine transform (DCT) or wavelet packet transform (WPT) [22, 23]. These methods made contributions by using transformation basis to compress the data, concentrating the energy of the original signal in data packets of a smaller size.

The perceptual scheme focuses on reducing the packet over the communication network. The first packet rate reduction for networked control system was believed to be proposed in [24]. This work supposes a fixed threshold, of which only samples containing changes higher than it are transmitted. The receiver reacts to missing samples by holding the value of the most recently received sample. However, the threshold given by [24] has strong limitations as a fixed value instead of a dynamic one fluctuating with the human perception. With the development of exploiting Weber's law of JND in [25], [26] proposed a dynamic threshold which changes with the previous transmitted sample. Since then, the so-called PD-based approach was put forward and introduced in [6–9]. In order to fill in the non-transmitted sample on the TOP, predictive coding is proposed to estimate future haptic data from previously received one. The simplest but also the least efficient prediction method is called the zero-order hold (ZOH) predictor. As long as the receiver does not receive any data at each sampling time, the receiver will hold the value of the last sample it received. In addition, in [27] and [28], a first-order linear predictor (FOLP) was given. During the non-transmission period, the receiver generates linearly increasing/decreasing samplesz with the slope of the last two samples, till receiving the next one. This simple predictor can lead to a significantly decreased packet rate, up to 90-95%, without deteriorating in the system. Furthermore, [29] deployed a third-order autoregressive (AR) model for three-dimensional position and force data. This model computed the adaptive coefficients from initialization and training processes, with the algorithm then deciding whether the training values needed to be updated either from the predicted data or the current real data by taking into account the JND threshold.

Nevertheless, the compression of tactile signals is beyond the scope of this thesis. The interested reader is therefore referred to [30, 31]

RNN & LSTM. For a bilateral teleoperation system, the haptic data reduction can be regarded as building a transmission threshold for time-varying sequences. Recurrent neural networks are a class of neural networks (NNs) dealing with time-series sequential data inputs or outputs. By introducing feedback to feedforward (FF) neural network, the relationship between data input and output is captured by RNNs. Despite of that, there exists a significant limitation of RNN models, called vanishing gradient effect. The state information of each cell integrates over time. In practice, the ability to backpropagate an error through a long range time interval becomes extremely difficult. Therefore, LSTM units, recurrent modules which enable long-range learning were first proposed in [60].

LSTM units have hidden state augmented with nonlinear mechanisms to allow state to propagate without modification, be updated, or be reset, using simple learned gating functions. LSTM has had successful application in many fields, such as speech recognition [32], language translation [33], and human activity recognition [34].

The advantages of LSTMs for modeling sequential data are twofold. First, LSTM models are straightforward to fine-tune end-to-end. The produced model is convincing even though no empirical models have been derived before. For example, [36] eliminates the need for complex multi-step pipelines in speech recognition by training a deep bidirectional LSTM that maps spectrogram input to text. The model produces convincing text translations even without a language model or pronunciation dictionary.

Second, LSTMs are flexible enough to deal with inputs or outputs of varying lengths, such as text or video. We next describe a unified framework to combine recurrent models such as LSTMs with deep convolutional networksin order to form end-to-end trainable networks capable of complex visual and sequence prediction tasks. For example, [33] and [37] use a multi-layer LSTM encoder and decoder to translate sentences from English to French.

DRT. Dimensionality reduction technique is a method that can map the data from the original high dimensional data to low dimensional data during the measurement, which aims to decrease the required storage space, achieve a faster transmission and use less transmission time [42]. The basis of dimensionality reduction techniques is to use the data with reduced dimensionality to represent the original data as much as possible. According to previous research, dimensionality reduction techniques have been applied in numerous fields, including recognition system [43], wireless network security [44], and disease prediction [45].

This thesis proposes a pipeline to solve the compression-recovery problem of haptic signals, and specifically compares three typical dimensionality reduction algorithms to reduce the size of each transmitted data. The first is principle component analysis (PCA). This a common approach to data analysis, is often used to reduce the dimensionality of high-dimensional data, and can extract the main feature components of the data [47]. PCA is attractive as it is a linear method that uses the mathematical method to transfer high-dimensional datasets to low-dimensional datasets whilst keeping the principal components. The principal component of data is extracted by calculating the eigenvalue and eigenvector of the covariance matrix, which will be explained in 2.4. The previous research in [46] has proved that principal components could cover 70-80% of data variation. Moreover, PCA is a reversible technique which allows data with reduced dimensions to recover its original dimension via a simple computation.

The second algorithm is stacked auto-encoder (SAE). This neural network consists of several layers of sparse autoencoders, where output of each hidden layer is connected to the input of the successive hidden layer. The benefit of SAE is that it provides a version of raw data with much detailed and promising feature information, which is used to train a classifier with a specific context, and achieve better accuracy than training with raw data [48].

The third algorithm is uniform manifold approximation and projection (UMAP), which is a novel manifold learning technique for dimension reduction. UMAP is constructed from a theoretical framework based in Riemannian geometry and algebraic topology. Due to its good performance on data visualisation, UMAP has been applied in different fields, including hyper-spectral data [50], page management [51], as well as voice separation [52]. Generally, UMAP could be described as adopting the local manifold approximation principles and generating the local fuzzy structure to create high-dimensional topology; it then should consider building a similar topology in low-dimensionality. Additionally, UMAP optimizes the difference between the original and new topology by using the cross-entropy function [49]. The progress of fuzzy topology creation could be divided into two phrases. One is the manifold of the approaching data; another is the representation of simple fuzzy sets for approaching manifold building. The result is a practical scalable algorithm that applies to real world data. Furthermore, UMAP has no computational restrictions on embedding dimension, making it viable as a general-purpose dimension reduction technique for machine learning [49].

Unsupervised Clustering Techniques. Unsupervised clustering technique is an approach which works for datasets that have no outcome (target) variable, nor any information about the relationship between observations, that is unlabeled data. In real life, it is arduous to produce a labelled data set if the volume of data is large or the research is difficult. Furthermore, even if the labelling is done manually, the speed of labelling is much slower than the speed of data production. As for kinesthetic data, the collected data is originally unlabelled, although we can label them by transmission status. In order to reduce the workloads of creating labels for each data, 2 unsupervised clustering techniques are introduced.

K-means is one of the most common-used unsupervised learning algorithms for solving the well-known clustering problem [87]. K-means has now been improved and extended by many scholars and applied to a wide range of fields. Kapil, Chawla and Ansari clarified that clustering algorithms including K-means, have been used in software engineering, statistics, data mining, image analysis, machine learning, web cluster engines and text mining in various disciplines to infer groups in large amounts of data, such algorithms group objects into clusters, making the objects in one cluster more homogeneous with

other clusters [88]. Moreover, K-means algorithm saves time and increases the speed of clustering. As mentioned in [89], on image data classification using K-means, clustering or grouping of data is the key initial process in image processing. In this case, there is a dramatic increase in the size of enterprise databases which contain large amounts of text and images. These huge databases need to be mined and accurate decisions made in a short period of time. Therefore, the concept of image segmentation plays a useful role in clustering as it is time-saving and efficient.

Hierarchical clustering is another algorithm for clustering data. Ward showed that it is often used in large-scale studies to form hierarchical groups of mutually exclusive datasets, where the individual data within each dataset have the most similarity in terms of specified characteristics [90]. Therefore, in contrast to K-means, which clusters the data itself, the Hierarchical clustering algorithm is often used in studies that require analysis of the links and correlations between clusters [81]. There are many more examples of hierarchical clustering. Tomasini and Van Wassenhove point out the differences between commercial and humanitarian supply chains and state that humanitarian supply chains must be able to short periods for responding to multiple emergencies and provide feedback [91]. In addition, hierarchical clustering and routing procedures for coordinating vehicle routes in large-scale post-disaster distribution and evacuation activities are stated in [92].

GBDT. Gradient boosting decision tree is a powerful machine learning tool for achieving good predictive performances in many data mining and machine learning problems. It has been widely used in many applications. For instance, online advertising [38], search ranking [39], and instance transfer [40]. This algorithm can be applied to deal with two types of problems, classification and regression problems. The base learners of GBDT are decision trees, to which leaf nodes a real value is assigned [41].

As an iterative decision tree algorithm, GBDT consists of multiple decision trees, and the conclusions of all trees are accumulated to derive the final answer. Therefore, the prediction model in the form of an ensemble of weak prediction models, in other words, a group of weaker decision trees, is derived after the training process.

In this thesis, GBDT algorithm is introduced to predict the haptic feedback since the current ZOH, and FOLP models simply simulate the motion during 1 sampling interval as either static or uniform. As illustrated above, the predictive values from both ZOH and FOLP models are dependent on either 1 or 2 samples previously received [27, 28]. The GBDT algorithm is allowed to train the model with the input and output data in a variety of ranges of time, from beginning to the end, which is more realistic. The motion of human OP is normally decided by a long-range time, instead of 1 or 2 ms. Although ZOH and FOLP models are reasonable as the sampling frequency is normally greater than 1 kHz, this thesis aims at delivering an more accurate model with diminished loss of transparency.

1.4 Contributions

Kinesthetic data reduction techniques and predictive schemes are currently the most flourishing areas in bilateral teleoperation systems. However, the combination of machine learning and kinesthetic data analysis has been incorporated into this thesis, whereas the majority of the exposed idea and analysis is the contribution of the author. The contributions to the research community can be summarised as follows:

1. A bilateral teleoperation system with LSTM-based models for kinesthetic data reduction has been proposed, as well as an algorithm for generating output labels for the training process.

2. A bilateral teleoperation system with dimensionality reduction and data reconstruction techniques has been proposed. Three different dimensionality reduction techniques and corresponding data reconstruction techniques are used to deal with kinesthetic data.

3. A bilateral teleoperation system with unsupervised clustering algorithms has been proposed. Two different clustering algorithms are used to separate data into several clusters.

4. A bilateral teleoperation system with GBDT-based predictive schemes has been proposed.

5. A kinesthetic data collection experiment has been set up.

1.5 Organisation of the Thesis

To accomplish a logical thread in reducing kinesthetic data transmission in bilateral teleoperation systems by using different machine learning algorithms, the thesis has been organised as follows. In Chapter 2, various kinesthetic data reduction techniques are proposed, where the potential kinesthetic data transmission under each data reduction technique is derived. Also, a novel predictive scheme is also given. These are then utilised in Chapter 3 to 5 to simulate kinesthetic data reduction; again, many metrics are used to measure the performance of each data reduction technique. Additionally, the performance analysis of proposed predictive scheme is discussed in Chapter 5 as well. Conclusions to the entire thesis are drawn in Chapter 6, which are accompanied by suggestions on future research taking the result of the whole thesis into account.

In more details, major contributions of Chapter 2 can be summarised as follows. First, a brief introduction of Weber's law of JND is given, which is the basis of the idea

of kinesthetic data reduction. Then, a novel kinesthetic data reduction technique with LSTM-based model is proposed. The labelling algorithm is evolved from Weber's law of JND, which helps LSTM network derive the relationship between kinesthetic data and its transmission status. In addition, a bilateral teleoperation system with DRTs are also proposed. Three different DRTs and data reconstruction techniques are given. This is a completely new thought that abandons the traditional idea of selectively transmitting kinesthetic data over the network, and instead reduces the size of each transmission. Also, unsupervised clustering algorithms are used for comparison with supervised learning, and two clustering algorithms are given. This is realised by separate a set of kinesthetic data into several clusters, and only selected clusters of data is transmitted over the network. Also, a novel predictive scheme based on GBDT is given. However, traditional ZOH and FOLP schemes are both based on Weber's law of JND.

The effort of Chapter 3 concentrates on setting up a kinesthetic data collection experiment, and comparing the proposed LSTM-based mathematical model with PD-based codecs. HPW-PSNR and transmission rate are two metrics to measure the performance of models. Furthermore, in order to make the result more intuitive, figures of velocity and force signals along three axes are given.

Chapter 4 aims at analysing the performance of three different DRTs, and compare them with proposed LSTM-based model and PD-based codecs. First, for comparing the performance of three DRTs, five metrics are used to test the embeddings under four textures. Then, for testing the loss of transparency due to dimensionality reduction, HPW-PSNR is used to measure the data recovered from three data reconstruction techniques. Additionally, further comparison of DRTs with LSTM-based models and PD-based codecs is also discussed.

In Chapter 5, we make efforts on simulations of unsupervised clustering and predictive schemes. As an extension of supervised learning proposed in Chapter 3, labels are not required when training kinesthetic data. Instead, several clusters can be derived when dealing with a set of kinesthetic data. The first part of this chapter concentrates on finding the optimal number of clusters for two clustering techniques. Then, transmission bits and HPW-PSNR of unsupervised clustering techniques are used to compare with LSTM-based models and PD-based codecs. Another part of this chapter is the analysis of proposed GBDT-based predictive scheme. A further discussion of comparison with traditional ZOH and FOLP schemes is also given.

Chapters 2 to 5 contain almost exclusively novel material with background information kept to a minimum.

Chapter 2

Various Kinesthetic Data Reduction Techniques Proposed in a Bilateral Teleoperation System

2.1 Introduction

This chapter proposes bilateral teleoperation systems with kinesthetic data reduction and predictive modules for achieving the 1-ms challenge in haptic communications. Three different kinesthetic data reduction techniques, and one predictive algorithm are introduced in this chapter. Additionally, current bilateral teleoperation systems with PD-based codecs and predictive scheme are also explained. The current kinesthetic data reduction technique based on Weber's law of JND is given in Section 2.2. Contributions of the author can be found in Section 2.3 to Section 2.6, which are the proposal of a bilateral teleoperation system with various machine learning techniques, including LSTM-based kinesthetic data reduction models, dimensionality reduction and data reconstruction techniques, unsupervised algorithms, and GBDT-based predictive schemes.

This chapter is structured as follows. The concept of Weber's law of JND and its application to kinesthetic data redctuction are given in Section 2.2. This law describes the limitation of humans' perception when distinguishing 2 similar stimulus, of which 1 stimuli is within the deadzone of the other. It shows that the range of this deadzone is normally proportional to the magnitude of the reference stimuli. Therefore, PD-based codecs based on Weber's law of JND was proposed in bilateral teleoperation systems. It was achieved by introducing some tolerable errors within the JND threshold which are imperceptible to human OP. This codecs can selectively transmit kinesthetic data by doing pairwise comparison, between the newly collected and last transmitted sample. Additionally, prediction models were proposed to estimate the future samples from previous

data. It can predict samples when no data is received on both OP and TOP sides. Two current predictive schemes called ZOH and FOLP are also introduced.

In Section 2.3, bilateral teleoperation systems with LSTM-based models for kinesthetic data reduction are proposed. It first analyses the drawback of current JND perceptual threshold used in PD-based codecs, and concludes that a new mathematical threshold should be proposed by avoiding pairwise comparison. LSTM netowrks have exhibited many successful experiences when dealing with time-series of data [32–34]. Therefore, Section 2.3 introduces kinesthetic data reduction in proposed teleoperation systems with LSTM-based models. Later, the design of LSTM-based data reduction modules is given. Mathematical expressions and functionalities of each gate are introduced in details. Moreover, the training process shows how LSTM networks inherit Weber's law of JND by labelling transmission status. An LSTM network with haptic data input and transmission status output is thus proposed. We also describes effects of timesteps in label prediction; that is the label in 1 ms can be affected by previous 1 to 100 samples.

When it comes to Section 2.4, a novel idea of reducing the size of kinesthtic data is given, unprecedented in abandoning the concept of selective transmission (reducing the amount of data transmission). This DRT-based kinesthetic data transmission scenario is analysed in this section. This idea is inspired by the phenomenon that a physical law that friction an object suffers is proportional to the pressure. Therefore, original force data in $\mathbf{R^3}$ can be represented by a 2D vector. Therefore, bilateral systems with 3 different DRTs (PCA, SAE and UMAP) for kinesthetic data reduction are given. However, data reconstruction modules are proposed simultaneously, with which receivers can recover dimensions from the reduced to the original. Three data reconstruction techniques are given corresponding to three DRTs.

In Section 2.5, unsupervised clustering techniques are used to classify kinesthtic data into different clusters; this idea allows data in a part of clusters to be transmitted over the network. By abandoning the derivation of transmission status from Weber's law of JND, it extends the labelled training (supervised learning) in LSTM networks to unsupervised learning. In this case, clusters are segregated by means of similarities of internal points and the distances amongst external clusters. Bilateral teleoperation systems with 2 different unsupervised techniques for kinsesthetic data reduction are given, including k-means and agglomerative hierarchical clusterings. By selecting data in a portion of clusterings to be transmitted, both k-means and hierarchical clustering techniques can effectively reduce the kinesthetic data transmission over the network.

In Section 2.6, GBDT-based predictive scheme is proposed to increase th whole accuracy of the whole bilateral teleoperation systems. In fact, GBDT is viewed as a powerful and accurate machine learning tool to predict data with the help of several weak learners. The predictive scheme and model update control strategy are also given in this

section. Update control modules are required since it determines whether or not the current GBDT model is accurate enough to predict force feedback. Also, the design of GBDT modules is also described.

2.2 Basics in Weber's Law of JND

2.2.1 The Concept of Weber's Law of JND

Just noticeable Difference is the minimum amount of the stimulation that must be changed to produce a significant variation in sensory experience. Ernst Weber, an experimental psychologist in 19th century, found that the size of this difference which can be noticed by a person appeared to be lawfully related to initial stimulus magnitude [25]. The relationship can be expressed as:

$$JND: \quad \frac{\|\Delta\vec{x}\|}{\|\vec{x}\|} = k \tag{2.1}$$

Here, the stimulus is represented by vectors. $\vec{x}$ represents a reference stimuli, and $\Delta\vec{x}$ represents the maximum increment or decrement of stimuli $\vec{x}$ which is still considered an imperceptibly change (i.e., the JND) and k is a constant called the Weber parameter. The interval $[\vec{x}-\Delta\vec{x},\vec{x}+\Delta\vec{x}]$ is often referred to as the perceptual deadband. As for the two slightly different stimulus, which are within in the deadband, it shows that the human haptic sensory system gives the same feedback. Mostly, suppose users are asked to differ the reference stimuli $\vec{x}$ from other stimulus contained in the respective deadband thresholds defined by $\vec{x}$ and the constant k. In that case, they are unable to do so and would just consider both as the same stimuli. This fact can be exploited to selectively skip packet transmissions whenever similar enough stimulus can be estimated and displayed instead. Broadly speaking, Weber's law can be applied in various stimulus, including force, velocity, inertia, and texture. Besides, the Weber parameter k is adjustable with the stimuli. For example, as is given in [53], the JND when a human operator perceives force feedback to the index finger is approximately 10%. As is given in the Table 2.1, values of k under different physical properties can be found there.

Meanwhile, those physical properties are both vectors, which indicates that Weber's Law can be extended to the three-dimensional (3D) space in reality. Disparate from a linear deadband in one-dimensionality, the perceptual deadzone can be assumed as a sphere in 3D case. As is shown in Fig. 2.1, the red line $x(\vec{t})$ refers to the compared haptic sample, and black line $x(\vec{t'})$ represents the reference haptic sample. The red sphere represents the deadzone of $x(\vec{t'})$. The radius of deadzone is normally proportional to the magnitude of the reference sample $x(\vec{t'})$. Additionally, the Weber parameter k can be found in Table

2.1. Once the magnitude of the reference stimuli and the weber parameter are determined, the deadzone can be derived respectively. If the compared value lies in the deadzone of reference value, then humans cannot feel any difference between $x(\vec{t})$ and $x(\vec{t'})$. Otherwise, humans can distinguish between two stimuli.

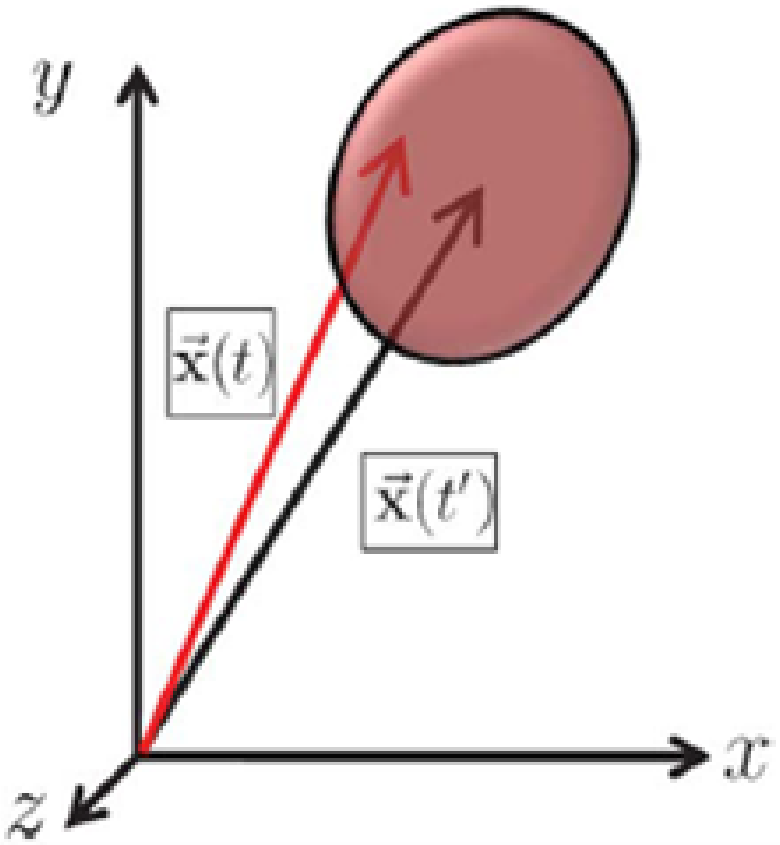


Fig. 2.1 An Explanation of Deadzone [8]

Table 2.1 JND of human perceptual discrimination for haptic signals [8]

Physical property	JND	Experimental conditions
Force	10%	arm/forearm
Movement	8%±4%	arm/forearm
Stiffness	23%±3%	arm/forearm
Viscosity	34%±5%	arm/forearm
Inertia	21%±3.5%	pinch-fingers, at 12 kg

2.2.2 PD-based Codecs

The approach in [9, 54–57] termed PD-based codecs is based on Weber's Law of JND, which is the widest-spread technique to reduce the kinesthetic data transmission. In general, PD-based codecs are achieved by introducing some tolerable errors within the JND threshold which are imperceptible to human OP. This scheme can be expressed as:

$$\begin{array}{lll} \text{If} & \dfrac{\|x(\vec{t}) - x(\vec{t'})\|}{\|x(\vec{t'})\|} \leq k_p & \text{Do not Transmit} \\ \text{Else} & & \text{Transmit a New Sample} \end{array} \tag{2.2}$$

where $x(\vec{t})$ is the data currently collected, $x(\vec{t'})$ is the last transmitted sample, $\|x(\vec{t'})\|$ represents the magnitude of $x(\vec{t'})$, $\|x(\vec{t}) - x(\vec{t'})\|$ measures the difference between newly collected and last transmitted data, and k_p is the perceptual threshold parameter. In the fact of PD-based codecs, the parameter k_p can either be greater or smaller than the Weber parameter k of the transmitted physical measurement. According to this scheme, the newly collected data will only be transmitted if the relative difference between it and the data last transmitted exceeds perceptual threshold. As long as k_p is given, some unnecessary kinesthetic data transmission can be selectively skipped.

Consequently, pairwise comparison can be found between the new and previously transmitted sample. However, PD-based codecs is not suitable when dealing with a time series of haptic data. Under the assumption of applying PD-base in practice, the comparison time interval is less than 1 ms since the sampling rate is 1 kHz or higher. Within 1ms, the memory should determine whether ot not the newly collected data is in the deadzone of previously transmitted data. If it is out of the deadzone, new data should be overwritten in memory.

It is thus clear that PD-based codecs not only increases the complexity of the whole system, but wastes the limited memory resources. In order to avoid this pairwise comparison, a mathematical model controlling the kinesthetic data transmission is urgent to be proposed, so that the transmission status of the newly collected sample can determined as long as it is collected.

2.2.3 PD-based Schemes with Predictive Coding

Prediction models for kinesthetic data are required to estimate the future samples from previous data. As is shown in Fig. 2.2, an example of a PD-based kinesthetic data reduction with predictive modules, two same predictive modules are deployed on the both master and slave sides. When triggering the model, newly collected samples come to the predictive module, and make a pairwise comparison with the data last transmitted in the JND verifier,

which has been explained in (2.2). Normally, the predictor generates the predicted haptic signal at every sample instant on the slave side. Once the transmission happens, the newly transmitted/received data triggers the prediction model updates, and overwrites the data in the memory. Also, the received data is used as the instruction from the OP.

Typically, there are two types of predictive schemes, called ZOH and FOLP. For each scheme, we can represent is separately. For the ZOH scheme, it can be represented by:

$$OP: \quad \vec{x}_t(t) = \begin{cases} \vec{x}_t(t), & \text{Predictive Model Updates} \\ None, & \text{No Data Transmitted} \end{cases} \tag{2.3}$$

$$TOP: \quad \vec{x}_r(t) = \begin{cases} \vec{x}_r(t), & \text{Predictive Model Updates} \\ \vec{x}_r(t'), & \text{No Data Received} \end{cases} \tag{2.4}$$

where $\vec{x}_t(t)$ represents the current transmitted sample on the master side, and $\vec{x}_r(t)$ is the current sample on the slave side either retrieving from the receiver or generated by the predictor. Additionally, t' is the most recent time when the packets received by the slave side. Therefore, in the ZOH predictive scheme, receiver holds the last value of the sample it previously received till receiving a new sample.

For the FOLP scheme, the mathematical expression is as follows:

$$OP: \quad \vec{x}_t(t) = \begin{cases} \vec{x}_t(t), & \text{Predictive Model Updates} \\ None, & \text{No Data Transmitted} \end{cases} \tag{2.5}$$

$$TOP: \quad \vec{x}_r(t) = \begin{cases} \vec{x}_r(t), & \text{Predictive Model Updates} \\ \dfrac{\vec{x}_r(t') - \vec{x}_r(t'')}{t' - t''}(t - t') + \vec{x}_r(t'), & \text{No Data Received} \end{cases} \tag{2.6}$$

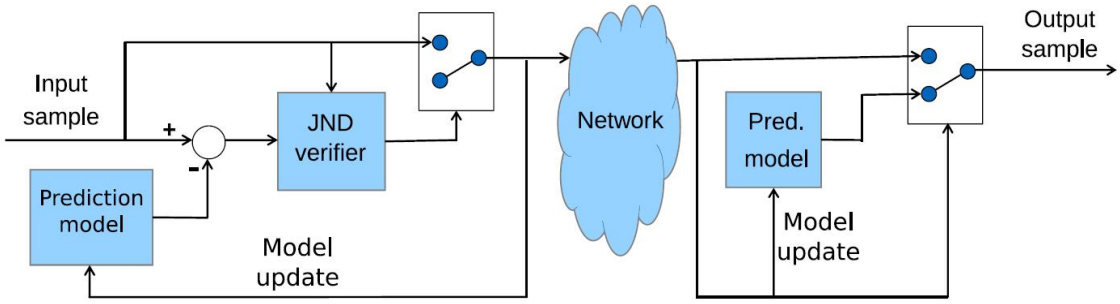


Fig. 2.2 Overview of the PD-based Kinesthetic Data Reduction with Predictive Coding

Different from ZOH predictors, the FOLP scheme predicts future samples by connecting last 2 points received at t' and t'' into a straight line. Therefore, FOLP is a linear predictive scheme that receiver predicts the motion of TOP is uniform when no data receives.

To visualize the actual and transmitted samples with predictive schemes, we assume that an object is moving along one axis, in which case velocity data can be represented by one-dimensional vectors. Therefore, an example of 1-DoF of actual and transmitted samples with ZOH and ZOLP predictive schemes are sketched in Fig. 2.3 and Fig. 2.4. In Fig. 2.3, 10 actual samples collected on the master side within 10 ms are given. However, as is shown in Fig. 2.4, only a part of the kinesthetic data is transmitted over the network. Also, the data reduction scheme is PD-based codecs. Therefore, as explained in (2.2), the transmission kinesthetic data happens when the difference between predicted and newly collected sample is perceptual to human operators. In this example, the samples above red dashed lines are not transmitting. It is thus obvious that the samples to be transmitted over the network under ZOP and FOLP predictive schemes are quite different.

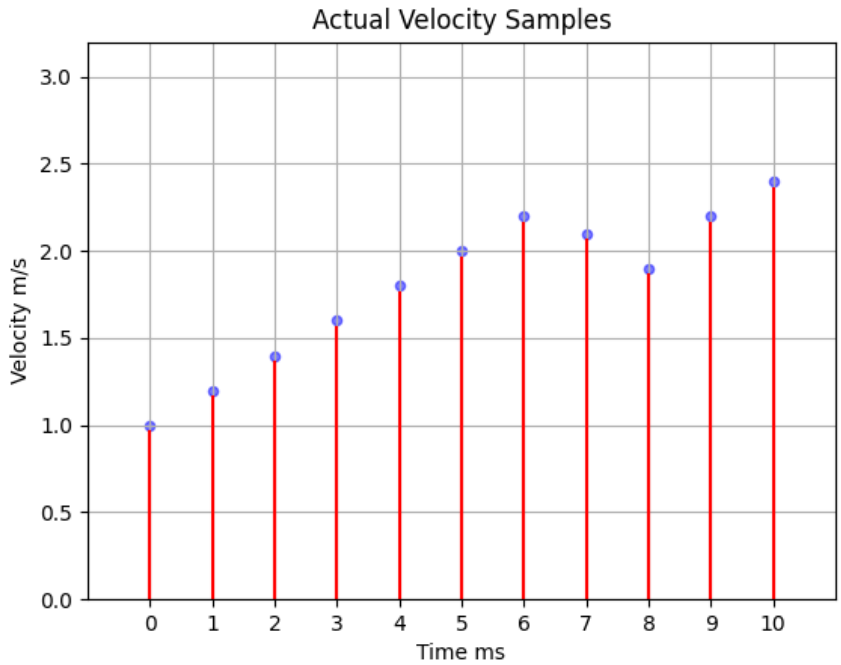
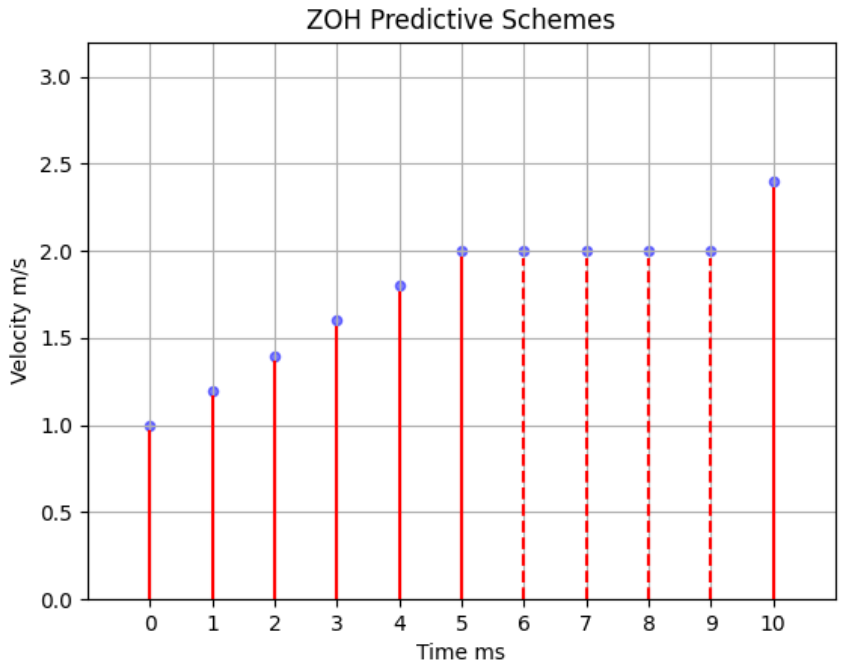
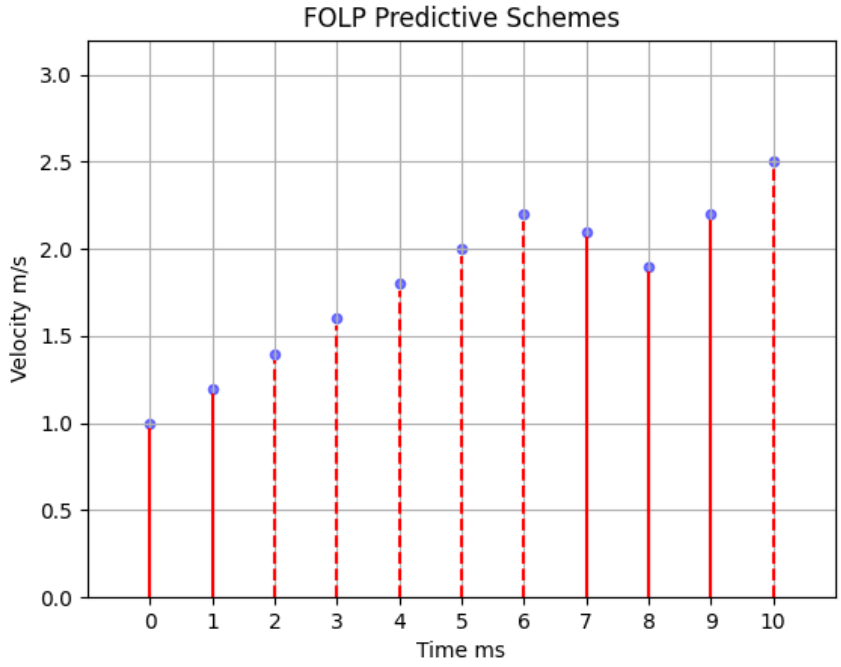


Fig. 2.3 An Example of Collected Velocity Data When an Object Moving Along x Axis



(a) An Example of Transmitted Velocity Data with ZOH Predictive Schemes When an Object Moving Along x axis



(b) An Example of Transmitted Velocity Data with FOLP Predictive Schemes When an Object Moving Along x axis

Fig. 2.4 An Example of Kinesthetic Data Reduction Scheme; In (a)&(b), the samples above the red dashed lines are not transmitting, otherwise, they are transmitting over the network

2.3 A Bilateral Teleoperation System with LSTM-Based Models for Kinesthetic Data Reduction

2.3.1 Motivation

As illustrated in Section 1.3, the current kinesthetic data reduction technique is based on Weber's Law [26, 6–9]. However, the current JND model is not accurate enough to deal with a time series of kinesthetic data from three aspects: i) the accuracy of Weber's law is not high enough; ii) the pairwise comparison in Weber's law is time-consuming; iii) the fault tolerance of current JND perceptual threshold is zero.

In recent research, many researchers found that Weber's law cannot represent humans' deadzone for some types of stimuli in practice. In human time perception, Haigh et al have found that the relationship sensitivity and duration on interval timing tasks is a reversed-J shape or a U-shaped function, rather than a linear relationship predicted by Weber's law [16, 58]. As for tactile data, Heath et al have found that JNDs increased linearly with increasing target object size when employing a manual estimation wherein the sizes were increasing, which adhered to Weber's law [17]. Moreover, Bhardwaj et al have found that the best expression of deadzone is level crossing classifier, instead of Weber's law, when they asked human to respond if he/she feels any change in the stimuli in a kinesthetic perception setup [18, 19].

From the angle of cost, the current JND perceptual threshold is a pairwise comparison which is applied in all of the collected kinesthetic data. Within 1ms, the mathematical model in memory should determine whether or not the newly collected data is in the deadzone of previously transmitted data. If it is out of the deadzone, the new data should overwrite the data in memory.

On top of that, the fault tolerance of current JND perceptual threshold is nearly zero. A small error in the pairwise comparison can affect outcomes. Assuming that the transmitted data is miswritten, or the comparison is wrong, this JND perceptual threshold will keep running incorrectly since the current wrong value will be used for the following comparisons.

In order to avoid this pairwise comparison, a mathematical model controlling the kinesthetic data transmission should be proposed. The transmission status of the newly collected data will be determined by the model directly without comparison, which reduces the time cost and increases the reliability of the system. As is mentioned in Section 1.3, multiple layers of LSTM networks can increase accuracy and maintain the stability of the entire model, which is suitable for dealing with time series of kinesthetic data and building a new data reduction threshold. The advantages of LSTM network can be concluded in three aspects: i) LSTM network is more reliable since the long-term memory function can

predict the value based on previous, sequential data; ii) LSTM network has been applied in many fields successfully when dealing with a time series of data; iii) The transmission status of newly collected kinesthetic data can be derived from the model directly.

A conventional LSTM network is given in Fig. 2.5, the cell can remember values over arbitrary time intervals, and three gates regulate the flow of information into and out of cells. Furthermore, the LSTM network can take inputs with different lengths. This feature can adjust the model with different situations by resetting the number of timestep. A timestep refers to the number which subjects the current sample to several previous samples. For example, we assume the current sample has a relationship with the previous 100, then the timestep in this assumption is 100 ms. Additionally, when dealing with the kinesthetic data with higher frequency, i.e., the force data during the surface exploration, we can increase the timestep to derive a model fitting this situation.

Over and above that, LSTM networks have exhibited many successful experiences when dealing with time-series of data [32–34]. This architecture utilizes model parameters more efficiently than other architectures, converges faster, and outperforms deep FF neural networks having an order of magnitude of more parameters. When it is applied in the reality of kinesthetic data transmission in a bilateral teleoperation system, the transmission status of currently collected data can be derived from the trained model immediately, and the outcome is based on current and previous data, which is more applicable and reliable.

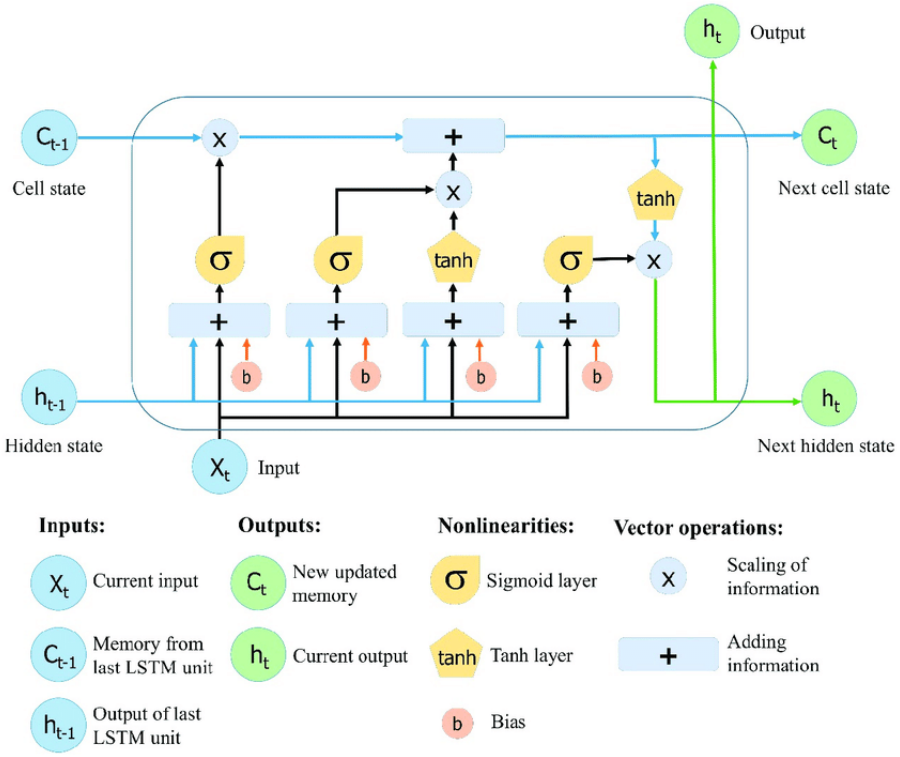


Fig. 2.5 A Conventional LSTM Network [59]

2.3.2 System Model

The proposed system model is given in Fig. 2.6, and the most significant difference between the proposed and conventional bilateral telecommunication system is data reduction modules for position/velocity data and force data. The data reduction module for position/velocity data aims at selectively transmitting collected data according to the mathematical model inside from master to slave side. Simultaneously, the force data reduction module is used to control the force transmission from the slave to master side.

During teleoperation, as soon as the human triggers the OP, a sensor begins to collect the position $(x_0, x_1 \ldots, x_i, \ldots, x_M)$ and velocity $(\dot{x}_0, \dot{x}_1 \ldots, \dot{x}_i, \ldots, \dot{x}_M)$ data of the moving cursor. Both x_i and $\dot{x}_i$ are vectors. As stated above, the more packets encapsulating kinesthetic data transmit over the haptic communication network, the less stable system will be. Even a small-time delay or packet loss can cause the jitter of the whole system. Therefore, data reduction modules are required in both OP and TOP side since the bi-directionality of the system. As is shown in Fig. 2.6, the position/velocity data is transmitted from the master to slave side, and the corresponding force feedback collected from the slave side is sent back to the master side via backward channel. The data reduction module reduces the amount of transmitted haptic data. Hence, the position/velocity data to be transmitted is $(x_0, \ldots, x_i, \ldots, x_j, \ldots, x_N)$ and $(\dot{x}_0, \ldots, \dot{x}_i, \ldots, \dot{x}_j, \ldots, \dot{x}_N)$, where $N \leq M$. Those data will pass through a radio channel and be received by the TOP. Since collected data on the TOP is discrete with variant time steps due to the data reduction process, the TOP should predict the motion when no kinesthetic data received. The data recovery module aims at modifying the received haptic data into a time series version. In this case, the FOLP is introduced in the TOP, assuming the OP is uniform during the receive interval. Therefore, the output of the data recovery module in the TOP can be interpreted as $(\tilde{x}'_0, x'_1, \ldots, \tilde{x}'_i, x'_{i+1}, \ldots, \tilde{x}'_j, x'_{j+1}, \ldots, x'_M)$ and $(\dot{\tilde{x}}'_0, \dot{x}'_1, \ldots, \dot{\tilde{x}}'_i, \dot{x}'_{i+1}, \ldots, \dot{\tilde{x}}'_j, \dot{x}'_{j+1}, \ldots, \dot{x}'_M)$,

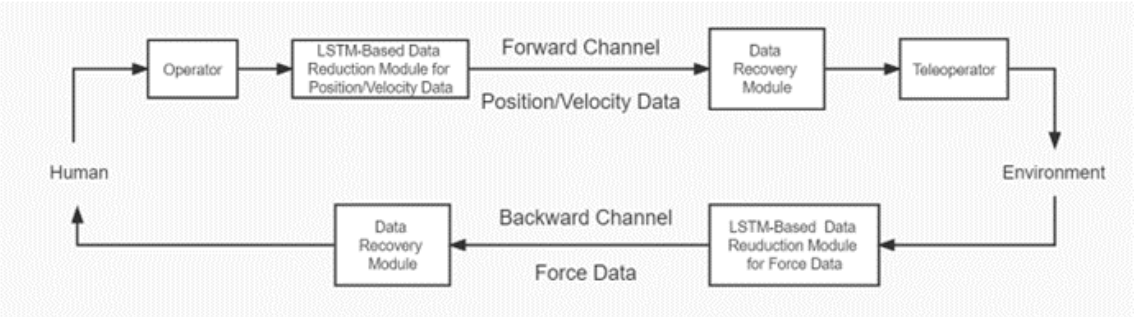


Fig. 2.6 A Bilateral Teleoperation System with LSTM-Based Data Reduction Modules

where $\tilde{x}'_i, \tilde{x}'_j$ and $\dot{\tilde{x}}'_i, \dot{\tilde{x}}'_j$ are the receiving position and velocity data at time i and j correspondingly. Between 2 data receiving intervals, the detailed relationship of output data from the data recovery module on the TOP side can be explained as:

$$\begin{cases} x'_{i+m} &= \tilde{x}'_i + \dot{\tilde{x}}'_i * mT_s \quad , \quad i+m<j \\ \dot{x}'_{i+m} &= \dot{\tilde{x}}'_i \quad , \quad i+m<j \end{cases} \tag{2.7}$$

where TOP received data at time i and j. Thenceforth, we can extend (2.7) to the whole transmission time, and increase the size of the received position/velocity data from N to M.

When the TOP interacts with the remote environment, force feedback is collected correspondingly, which can be interpreted as $(F_0, F_1 \ldots, F_i, \ldots, F_M)$. Each force data F_i is a vector as well. Similarly, only a part of the original force data needs to be transmitted back to the OP in a bilateral teleoperation system, which is $(F_0, F_{i'} \ldots, F_{j'}, \ldots, F_{N'}), N' \leq M$. A ZOH predictor is deployed on the OP to extend the size from N' to M. Therefore, the output of the data recovery module in the TOP can be interpreted as $(\tilde{F}'_0, F'_1, \ldots, \tilde{F}'_i, F'_{i+1}, \ldots, \tilde{F}'_j, F_{j+1}, \ldots, F_{N'})$. Similarly, the relationship of data between two force data receiving intervals can be explained as:

$$F'_{i+m} = \tilde{F}'_i \quad , \quad i+m<j \tag{2.8}$$

In this section, we concentrate on deriving a mathematical kinesthetic data reduction model to reduce the size of transmission either from M to N (OP sends to TOP) or M to N' (TOP sends to OP). The proposed mathematical model is based on LSTM networks, and assuming the data reduction module in OP is Θ, while Θ' is in TOP. As long as the threshold models are built, the relationship between the actual and transmitted velocity can be shown as follow:

$$\begin{cases} \Theta(\dot{x}_i, \Gamma(\dot{x}_a, \dot{x}_{a+1}, \ldots, \dot{x}_{a+n-1}, \ldots, \dot{x}_{a+N-1})) &> 0 \quad , \quad \text{Data transmits} \\ \Theta(\dot{x}_i, \Gamma(\dot{x}_a, \dot{x}_{a+1}, \ldots, \dot{x}_{a+n-1}, \ldots, \dot{x}_{a+N-1})) &\leq 0 \quad , \quad \text{Data remains} \end{cases} \tag{2.9}$$

where $\dot{x}_i$ is the actual velocity value, usually a 3D-vector, and Θ is a mathematical mapping function between the actual velocity value and the threshold model, Γ is the trained model derived from LSTM networks. Specifically, Θ is a mapping function generally built and trained between the training data within the step and its corresponding output label in the training process of LSTM networks. In order to retrieve the same type of output label from the mathematical model, Θ holds in the test process. Γ is derived from a set of training data with size N. It refers to the function of the data reduction module, affected by time step n, training data size N and training data $(\dot{x}_a, \dot{x}_{a+1}, \ldots, \dot{x}_{a+n-1}, \ldots, \dot{x}_{a+N-1})$. The (2.9) can be acknowledged as deriving the output label from the previously well-trained LSTM

model as long as new data is collected. Γ indicates the implication between data at one moment with previous data within a time step. Meanwhile, output labels of the training data represent transmission states of each data, which are derived from Weber's Law of JND. If the label is -0.5, the input data is not transmitted; otherwise, it is transmitted, when the label is 0.5. More details about the training process can be found in Section 2.3.3. During the training process, Θ derives a mathematical model to map the input data and output labels. Afterwards, in the test process, Θ gets a series of float numbers through the trained mathematical model to predict transmission states of each collected data. In this case, if the output label is greater than 0, the data should be transmitted, while not the other way.

Correspondingly, it is the same to express the force data reduction module in the OP as follow:

$$\begin{cases} \Theta'(F_i, \Gamma'(F_a, F_{a+1}, \ldots, F_{a+n-1}, \ldots, F_{a+N-1})) & > \quad 0 \quad , \quad \text{Data transmits} \\ \Theta'(F_i, \Gamma'(F_a, F_{a+1}, \ldots, F_{a+n-1}, \ldots, F_{a+N-1})) & \leq \quad 0 \quad , \quad \text{Data remains} \end{cases} \tag{2.10}$$

where F_i is the real 3D force feedback, Θ' is the mathematical mapping function from one data with all of its previous data within the time step to the output label, and $\Gamma'(F_i, F_{a+1}, \ldots, F_{a+n-1}, \ldots, F_{a+N-1}$ is the trained model to indicate the inside relationship of the training data and the time-step data.

As is shown above, an accurate LSTM-based data reduction model depends heavily on the haptic training data.

2.3.3 Design of LSTM-Based Data Reduction Modules

Data reduction modules aim at acquiring the transmission status of a time series of kinesthetic data without pairwise comparison. Therefore, the mathematical model in the module should be trained at first. As is shown in Fig. 2.7, a typical LSTM network is in a chain structure, which is made up of a sequence of repeating neural networks and numerous memory blocks called cells.

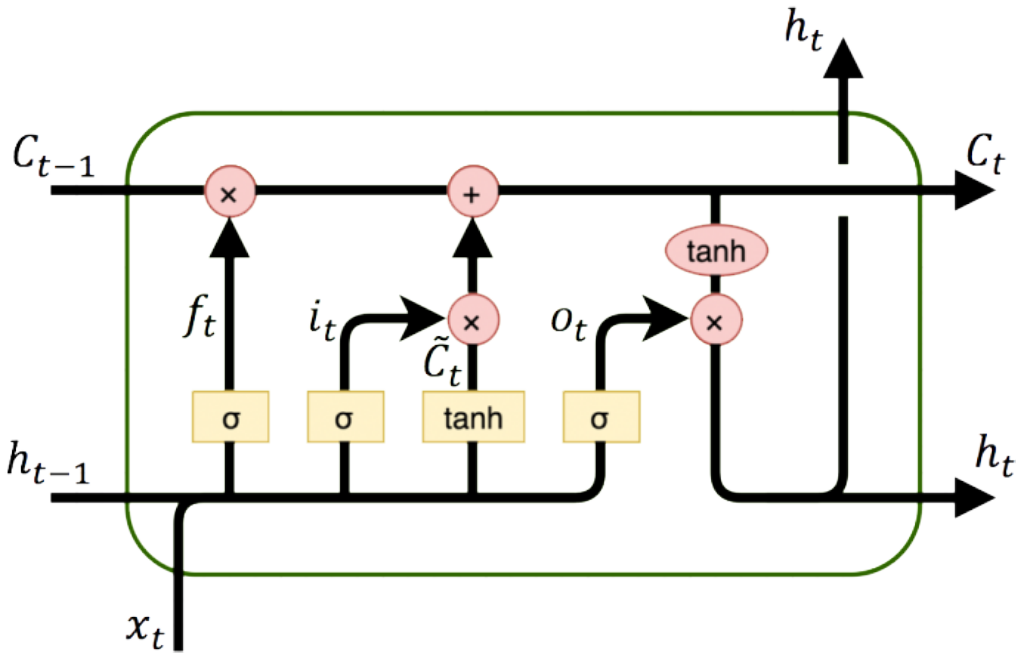


Fig. 2.7 An Overview of an LSTM Unit [59]

Normally, an LSTM unit is consisted of a a cell, an input gate, an output gate, and a forget gate. Three gates control into and out of the information flow of each cell, and the cell remembers values over arbitrary time intervals. In other words, the gates handles memory, whereas the cells store information. As is shown in Fig. 2.7, for a cell c and at time t, we have the following input-output relationships:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.11}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.12}$$

$$\tilde{C}_t = \tanh(W_u[h_{t-1}, x_t] + b_u) \tag{2.13}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.14}$$

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.15}$$

$$h_t = O_t \odot \tanh(C_t) \tag{2.16}$$

where $x_t \in \mathbf{R}^n$ is the input vector, $\tanh(.)$ is the hyperbolic tangent function, and $\sigma(.)$ stands for the sigmoid function. x_t can be collected velocity or force samples. Furthermore, $\odot$ represents the element-wise product. h_t and $h_{t-1} \in \mathbf{R}^l$ are the output values of the hidden layers at time t and $t-1$, respectively. h_t can be viewed as the filtered version of cell state C_t as well. Traditionally, both $\sigma(.)$ and $\tanh(.)$ are activation functions which define how the weighted sum of the input is transformed into an output from nodes in different gates of an LSTM network. In particular, an LSTM network uses the $\tanh(.)$ function for the activation of the cell state, and the $\sigma(.)$ function for the node output. $\sigma(.)$ specifically, it is the gating function of three gates (forget, write, output) which outputs non-negative values in the LSTM network. It determines whether the current flow of information can throughout the gates or not. Similarly, the output of $\tanh(.)$ can be positive or negative, allowing both increments and decrements of the states. On the other hand, to overcome the vanishing gradient problem, the second derivative of $\tanh(.)$ can sustain for a long range before going to zero. Apart from these two activation functions, the rectified linear unit (ReLU) activation function can also output values between 0 and 1, which can be used to replace $\sigma(.)$ function. In the meanwhile, the softsign function can be viewed as an replacement of $\tanh(.)$. However, the training results have shown that the overall loss (MSE) of the LSTM model with $\sigma(.)$ and $\tanh(.)$ activation functions is the smallest for two types of kinesthetic data(less than 10^{-4} for velocity signals, and 10^{-2} for force signals). Besides, more activation functions can be found in [94].

Besides, the forget gate (the leftmost gate in Fig. 2.7) determines how many activations of the previous cell state C_{t-1} is preserved at C_t. It is decided by a sigmoid function. For each number in the cell state C_{t-1}, it depends on the hidden state h_{t-1} and the current input value x_t and derives a number between 0 (forget this) and 1 (keep this). The write gate (the middle gate in Fig. 2.7) determines how many activations from the current input x_t is preserved at cell state C_t. It is decided by a sigmoid function and a tanh function. The sigmoid function determines whether to allow the current value changing the memory, 0 for no, and 1 for yes. The tanh function assigns weight to the data provided, determining their importance on a scale of -1 to 1. The output gate (the rightmost gate in Fig. 2.7) determines how many activations from the current cell state C_t is used as the output h_t.

The sigmoid function plays the same role as other gates, determine whether the current input values allow to change the output data, 0 for no, and 1 for yes. And the tanh function assigns weight to the current cell state, determining its relevance with h_t on a scale of -1 to 1. $[h_{t-1}, x_t]$ represents cascading two vectors into a longer vector. The weight matrices $W_k \in \mathbf{R}^{n+l}; k \in \{f,i,o,u\}$ and bias factors $b_k \in \mathbf{R}^l; k \in \{f,i,o,u\}$ are optimization parameters shared among all cells. There are four sets of parameters which need to be trained in an LSTM network, which are $W_f \& b_f$, $W_i \& b_i$, $W_u \& b_u$ and $W_o \& b_o$. The detailed formula derivation of training process can be viewed in [60].

Assuming a set of kinesthetic data with length N is used to train the model. The velocity data $(\dot{x}_a, \dot{x}_{a+1}, \ldots, \dot{x}_{a+N-1})$ is used to train position/velocity data reduction module, while force data $(F_a, F_{a+1}, \ldots, F_{a+N-1})$ is used for training the force data reduction module. Therefore, the input data is in shape of $X \in \mathbf{R}^{N\times 3}$. According to PD-based codecs, only a small amount of will be transmitted over the network. Consequently, the velocity data can be rewritten as $(\tilde{\dot{x}}_a, \dot{x}_{a+1}, \ldots, \tilde{\dot{x}}_b, \dot{x}_{b+1}, \ldots, \dot{x}_{a+N-1})$, where $\tilde{\dot{x}}_a$, $\tilde{\dot{x}}_b$ represents the velocity data that is supposed to be transmitted at time a and b. Simultaneously, the force data can be expressed as $(\tilde{F}_a, F_{a+1}, \ldots, \tilde{F}_b, F_{b+1}, \ldots, F_{a+N-1})$. On top of that, labels of transmission status should be given to each collected kinesthetic sample. The method to derive the label H_t inherits Weber's Law of JND, which can be derived as follows:

$$H_t = \begin{cases} -0.5, & \text{if } \dfrac{\|\vec{X}_t - \vec{\tilde{X}}_{t-m}\|}{\|\vec{X}_t\|} \leq \alpha \\ 0.5, & \text{if } \dfrac{\|\vec{X}_t - \vec{\tilde{X}}_{t-m}\|}{\|\vec{X}_t\|} > \alpha \end{cases} \tag{2.17}$$

where $H_t = -0.5$ stands for the sample collected at time t is not transmitted, and $H_t = 0.5$ means the sample needs to be transmitted. $\vec{X}_t$ is the 3D-kinesthetic data at t, and $\vec{\tilde{X}}_{t-m}$ is the previously transmitted data in the manner with PD-based codecs. α represents the JND of perceptual discrimination for different kinesthetic signals, and normally $\alpha = 0.1$. In practice, all of the training and testing data needs to be normalized to the interval $[-0.5, 0.5]$. Moreover, the output label $H_t \in \mathbf{R}^{N\times 1}$ is a series of -0.5 and 0.5.

The training algorithms of two kinesthetic data reduction modules are described in Table 2.2. The input of training data at t_0 is $X_{train}(t_0) \in \mathbf{R}^{\text{timestep}\times 3}$, which includes kinesthetic data with the size of timestep from $t = t_0 - \text{timestep} + 1$ to $t = t_0$. In the meanwhile, the output label H_{t0} in this case is the transmission status derived from Weber's law of JND. For example, when timestep is 100, t_0 is 1000, the corresponding output label $H[1000]$ is derived from $X[901]$ to $X[1000]$.

To summarize, the tasks of (2.17) is to generate labels for training data based on Weber's Law. As long as the training process goes smoothly, the trained model will converge to PD-based approach. In addition, when each new data is generated, the

transmission flag can be derived directly through the model without pairwise comparison, which makes it more practical in real transmission. Moreover, the introduction of timestep is more realistic; it claims the transmission flag of data in 1 ms can be affected by a portion of previous data. In the traditional PD-based codecs, the transmission flag is only related to the current and previously transmitted data, which does not take the series of haptic data into account.

Algorithm 1: Training algorithms for two data reduction modules

Initialize: Let input vectors X is a time series of collected kinesthetic data of length N; $X[0]$ is transmitted, the corresponding output label $H[0] = 0.5$; The reference value in memory $V = X[0]$; $\alpha = 0.1$; timestep $=$ 100ms; LSTM training data: $X_{train} = \{\}$, $Y_{train} = \{\}$

1. Normalise the input data: $\bar{X} \leftarrow \text{normalised}(X)$
2. **for** $K \leftarrow 1$ **to** N **do**
   - $a \leftarrow \|X[k] - V\| / \|X[k]\|$
   - **if** $a \leq \alpha$ **then**
     - $H[K] \leftarrow -0.5$;
   - **else**
     - $H[K] \leftarrow 0.5$
     - $V \leftarrow X[k]$;
3. **for** $i \leftarrow$ timestep **to** N **do**
   - X_{train}.insert($\bar{X}[i - \text{timestep} : i]$)
   - Y_{train}.insert($H[i-1]$)
4. //Building the LSTM Network//
5. .....
6. //Training Process//
7. .....
8. //Getting the Well-trained Mathematical Model//

Table 2.2 Training algorithms for two data reduction modules

2.4 A Bilateral Teleoperation System with Dimensionality Reduction and Data Reconstruction Techniques

2.4.1 Motivation

When observing the kinesthetic data carefully, we can find that all of the collected samples are in high dimensions. For instance, the position/velocity data is consisted of 6-DoF (3-DoF position + 3-DoF velocity), and the force data is normally represented by a 3-DoF vector. These dimensional disasters can lead to difficulties such as sparse data during transmission in the communication link or even affecting the prediction. In addition, linear correlation between features can easily occur in high-dimensional features, which means that some features are redundant. In the vast majority of cases, each type of data is a vector along three different axes (x, y, z). According to physical law, the magnitude of the friction of a moving object in the tangential direction (along x&y axes) is always proportional to the pressure it experiences in the vertical direction (along z axis). This phenomenon inspires us to reduce the size of force data transmission, even it allows us to extend the idea to position/velocity data.

However, the current kinesthetic data size reduction techniques have only been tested and achieved in 1-DoF. The first lossy kinesthetic data compression and decompression is named DCT [22], similarly to the JPEG codec, it compresses force feedback with a ratio of 20%. Furthermore, another compression method termed WPT proved with similar performance to DCT [23]. Based on these problems, dimensionality reduction methods are used on master and slave sides in this thesis, aiming to hold great promise in terms of reduced transmission rates.

Dimensionality reduction aims to map the data from the original high dimensional data to a low dimensional data through some ways. The principle of dimensionality reduction is learning a mapping relation $f : x \rightarrow y$, in which x is the high points and y is the low points after mapping. There are five benefits of dimensionality reduction: i) the storage space is reduced; ii) less transmission time is required due to the smaller transmission size; iii) Decreasing mistakes caused by useless information; iv) the degree of recognition is elevating; v) the intrinsic anatomically features of the data can be found.

Generally, DRTs could be classified as linear and nonlinear dimensionality reduction. The most popular and important linear techniques, such as PCA, mainly use mathematical methods and geometric analysis to find the maximum variance direction from high-dimensional data. Then the projection to the low dimension could keep the important information (the principal component). However, non-linear methods, such as UMAP, could deal with complicated high-dimensional datasets. Related research has shown that non-linear dimensionality reduction techniques could have a relative optimized

performance on large datasets [61]. For instance, the non-linear method could find the embedded space on a group of 'Swiss Roll' data in a 2-dimensional manifold embedded in 3-dimensional space since the linear method could not do that.

The basics of non-linear DRTs are also explained in [61]. Assuming we have a $n \times D$ matrix which could also be denoted as $X_{n\times D}$, which has n data vectors $x_i (i \in \{1,2,3,\ldots,n\})$ of dimensionality D. Also,the intrinsic dimensionality of $X_{n\times D}$ is d. In general, the intrinsic dimensionality datasets d of $X_{n\times D}$ is embedded into dimensionality D space via manifolds. In fact, dimensionality reduction techniques aim to transfer the datasets in dimensionality D to its intrinsic dimensionality d, and we named the new datasets Y. The proposed methods to extract the manifold from the high dimensional space is non-linear DRTs. In addition to this, the datasets Y not only keeps original information of X, but geometric distribution.

Overall, DRT is a technology that could transfer high-dimensional data into low-dimension data, so that the low-dimensional data could keep the useful original information in a smaller space and avoid the dimension problems. Fig. 2.8 shows the basic dimensionality reduction techniques that the original item has threedimension contains x, y, z three axes, and the right part presented 2 two-dimensional images.

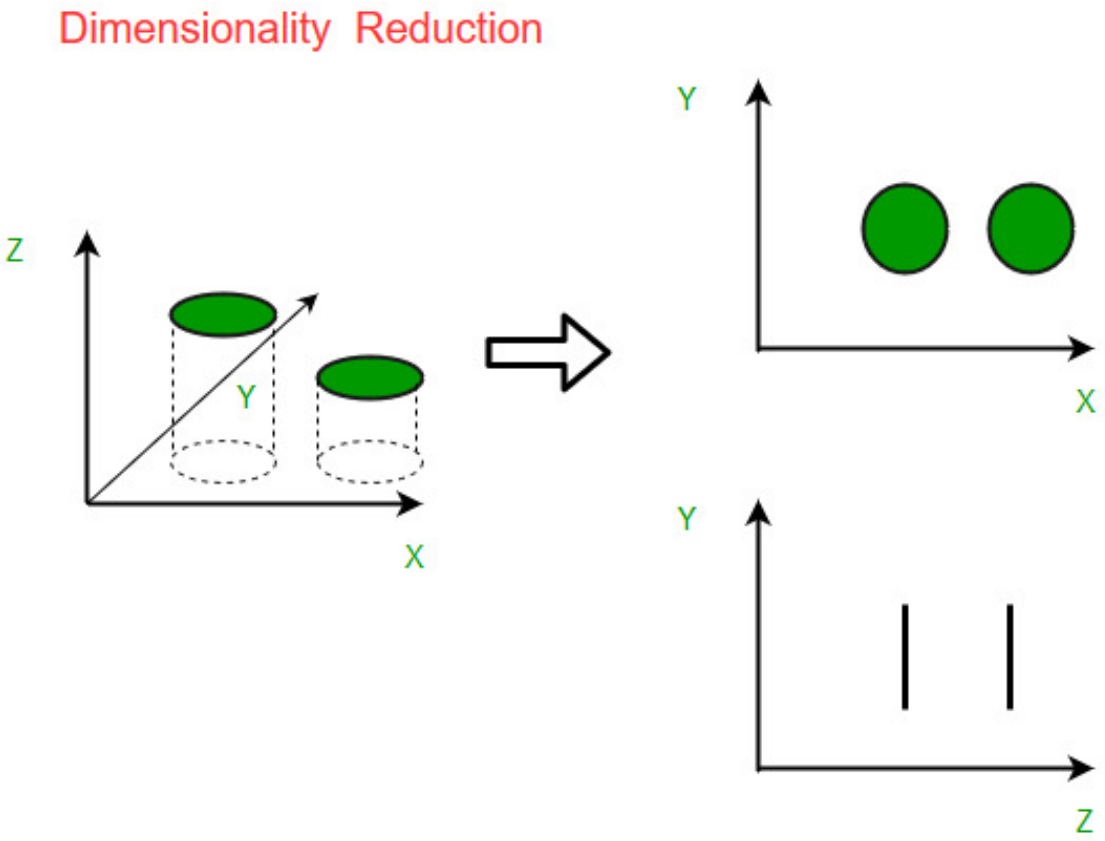


Fig. 2.8 An Overview of DRTs

2.4.2 System Model

The proposed system is given in Fig. 2.9. Two dimensionality reduction modules are deployed on both OP and TOP sides, aiming at reducing the size of each kinesthetic data packet. On top of that, there are two data reconstruction modules on the master and slave sides as well. The data reconstruction model targets on reconstructing the received data from reduced dimensions to the original one. In this thesis, three different dimensionality reduction techniques are applied, and each corresponding data reconstruction technique is proposed separately.

During the teleoperation process, the human OP moves the cursor on the master side, and a sensor starts collecting the position/velocity data of the cursor at the same time. Assuming the sampling rate is 1kHz, then in the following M ms, the collected kinesthetic data on the master side can be represented as:

$$X_{M\times 6} = \begin{bmatrix} x_1 & y_1 & z_1 & \dot{x}_1 & \dot{y}_1 & \dot{z}_1 \\ x_2 & y_2 & z_2 & \dot{x}_2 & \dot{y}_2 & \dot{z}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_M & y_M & z_M & \dot{x}_M & \dot{y}_M & \dot{z}_M \end{bmatrix} \tag{2.18}$$

where $\{x_i, y_i, z_i | i \in [1, M]\}$ is the i-th position data, $\{\dot{x}_i, \dot{y}_i, \dot{z}_i | i \in [1, M]\}$ represents the corresponding velocity data, and we assume M is 1000 in the simulation. As is mentioned in Section 2.4.1, the kinesthetic data transmitted from the OP to TOP is in a 6-dimensional space. In order to reduce the delay of the whole system, the dimensionality reduction module is deployed to reduce the transmitted data size. In this case, we assume the intrinsic dimensionality of the position/velocity data is 3. Hence, the transmitted data can be represented as:

$$Y_{M\times 3} = \begin{bmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ \vdots & \vdots & \vdots \\ u_M & v_M & w_M \end{bmatrix} \tag{2.19}$$

where $\{u_i, v_i, w_i | i \in [1, M]\}$ is the corresponding i-th transmitted data in a 3-dimensional space. Later, this block of data Y is transmitted over the network. On the TOP side, the received data can be represented as follows:

$$Y'_{M\times 3} = \begin{bmatrix} u'_1 & v'_1 & w'_1 \\ u'_2 & v'_2 & w'_2 \\ \vdots & \vdots & \vdots \\ u'_M & v'_M & w'_M \end{bmatrix} \tag{2.20}$$

where $\{u'_i, v'_i, w'_i | i \in [1, M]\}$ is the i-th received data of 3-DoF on the TOP side. At the current stage, the TOP cannot follow the command from the received 3-dimensional data due to the motion of the OP is 6-dimensional. For the purpose of the motion of the teleoperator synchronizes with the operator, a data reconstruction module is disposed on the TOP to recover the received data from $\mathbf{R}^3$ to its original dimension $\mathbf{R}^6$.

The reconstructed position/velocity data is given as:

$$X_{R_{M\times 6}} = \begin{bmatrix} x_{1_R} & y_{1_R} & z_{1_R} & \dot{x}_{1_R} & \dot{y}_{1_R} & \dot{z}_{1_R} \\ x_{2_R} & y_{2_R} & z_{2_R} & \dot{x}_{2_R} & \dot{y}_{2_R} & \dot{z}_{2_R} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{M_R} & y_{M_R} & z_{M_R} & \dot{x}_{M_R} & \dot{y}_{M_R} & \dot{z}_{M_R} \end{bmatrix} \tag{2.21}$$

Then the teleoperator follows the command of X_R, and starts the teleoperation with the remote environment. There is also a force sensor on the remote robot to collect the force data, which is normally in the 3-dimensional space $\mathbf{R}^3$. The collected force feedback on the TOP side is:

$$F_{M\times 3} = \begin{bmatrix} f_{x1} & f_{y1} & f_{z1} \\ f_{x2} & f_{y2} & f_{z2} \\ \vdots & \vdots & \vdots \\ f_{xM} & f_{yM} & f_{zM} \end{bmatrix} \tag{2.22}$$

where $\{f_{xi}, f_{yi}, f_{zi} | i \in [1, M]\}$ is the i-th collected force data on the TOP side. Due to the bidirectionality of bilateral teleoperation systems, there is also a dimensionality reduction module on the TOP to reduce the packet size of force data. In this case, we assume the 2-dimensional embedding concludes most of the force information. Subsequently, the dimensionality reduction module decreases the transmitted force data to a 2-dimensional space $\mathbf{R}^2$. And the transmitted force data from the slave side can be represented as:

$$E_{M\times 2} = \begin{bmatrix} f_{u1} & f_{v1} \\ f_{u2} & f_{v2} \\ \vdots & \vdots \\ f_{uM} & f_{vM} \end{bmatrix} \tag{2.23}$$

As long as E is transmitted over the network, the received force data with reduced dimensions on the OP side can be represented as:

$$E'_{M\times 2} = \begin{bmatrix} f'_{u1} & f'_{v1} \\ f'_{u2} & f'_{v2} \\ \vdots & \vdots \\ f'_{uM} & f'_{vM} \end{bmatrix} \tag{2.24}$$

Likewise, a data reconstruction module raises dimensions of the received force data from $\mathbf{R}^2$ to the original dimensions $\mathbf{R}^3$. The reconstructed force data on the OP side is as follows:

$$F_{R_{M\times 3}} = \begin{bmatrix} f_{x1_R} & f_{y1_R} & f_{z1_R} \\ f_{x2_R} & f_{y2_R} & f_{z2_R} \\ \vdots & \vdots & \vdots \\ f_{xM_R} & f_{yM_R} & f_{zM_R} \end{bmatrix} \tag{2.25}$$

Then when human feels the force feedback in (2.25), he/she will control the operator, and new position/velocity data of the moving cursor will be collected afterwards. As is shown above, the difference between the traditional and proposed bilateral teleoperation system is the dimensionality reduction and data reconstruction modules.

Therefore, DRTs aims to reduce the packet size by finding the intrinsic dimensionality out of the original dimensionality. Conversely, the data reconstruction modules raise the space from the reduced dimensionality to its original one. As long as the kinesthetic data with fewer dimensions is transmitted over the network, the possible distortion is inevitable. In order to reconstruct the received kinesthetic data on the slave side, we use relatively reconstructing approaches for three-dimensionality reduction techniques. Since the inverse operation of PCA is proposed with itself [62], we can reconstruct the kinesthetic data with inverse PCA on the slave side directly. Similarly, the inversion of SAE is achievable as well as long as the data in input layer, hidden layer, output layer is represented properly. However, UMAP is not inversible. Therefore, a neural network is trained and deployed to discover the relationship between original and UMAP data. After the loss of the whole neural network is acceptable, it will be able to reconstruct the received UMAP data on both master and slave sides.

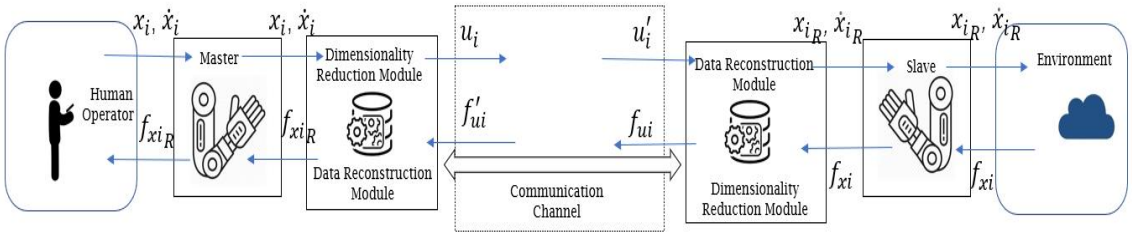


Fig. 2.9 The Proposed Bilateral Telecommunication System with Dimensionality Reduction and Data Reconstruction Modules

2.4.3 PCA-based Kinesthetic Data Reduction and Reconstruction Techniques

As is illustrated in Section 2.4.2 above, the kinesthetic data collected on the master side always has 6 dimensions, while the data on the slave side has 3, which can be represented in (2.18& 2.22). Then, three DRTs are introduced to reduce the dimension of $X_{M\times 6}$ and $F_{M\times 3}$ (M can be assumed as 1000). In this subsection, PCA is the main technique to reduce the dimensionality of the collected data. Also, the inverse operation of PCA, the principle which data reconstruction module obeyed, is also introduced.

The purpose of PCA is to find directions that maximizes the variance. If variance of one variable is higher than others, the principal components can be biased in that direction. Therefore, the first step of PCA is to zero-mean normalize each column of the data input, which means subtracting the average value for each column. Normally, normalized data input not only contains all of the information from the original data input, but also all the features are at the same scale. Additionally, normalization is done to ensure that the first principal component is in the direction of maximum variance.

For $X_{M\times 6}$, we can get $X'_{M\times 6}$ as:

$$X'_{M\times 6} = \begin{bmatrix} x'_1 & y'_1 & z'_1 & \dot{x}'_1 & \dot{y}'_1 & \dot{z}'_1 \\ x'_2 & y'_2 & z'_2 & \dot{x}'_2 & \dot{y}'_2 & \dot{z}'_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_M & y'_M & z'_M & \dot{x}'_M & \dot{y}'_M & \dot{z}'_M \end{bmatrix} \tag{2.26}$$

where $x'_i = x_i - \frac{x_1+x_2+\ldots+x_M}{M}$, and this rule can be applied to every element of any column vector in X'. Then the corresponding covariance matrix C of X' with its eigenvalue λ and eigenvector $\vec{V}$ can be calculated as:

$$C_{6\times 6} = \frac{1}{M} X^T_{6\times M} X_{M\times 6} \tag{2.27}$$

$$C_{6\times6}\vec{V}_{6\times1} = \lambda\vec{V}_{6\times1} \tag{2.28}$$

As the original data has 6 dimensions, the number of $\vec{V}$ and λ should be 6 as well. According to the eigenvalue of λ, the top N values and corresponding eigenvectors should be selected, where N is the number of the reduced dimension. And we select $N = 3$ in this case. Then the selected eigenvectors can form another matrix P as:

$$P_{6\times3} = \begin{bmatrix} \vec{V}_1 & \vec{V}_2 & \vec{V}_3 \end{bmatrix} \tag{2.29}$$

The matrix P is the chosen principal components, which maps the original 6-dimensional space to a new 3-dimensional space by extracting the main feature components of the data. Thus, the data with reduced dimensions $Y_{M\times3}$ can be calculated as:

$$Y_{M\times3} = X'_{M\times6}P_{6\times3} \tag{2.30}$$

As is explained above, P can only be derived if a kinesthetic dataset of size M (i.e., $X_{M\times6}$) is obtained. Therefore, both $Y_{M\times3}$ and $P_{6\times3}$ are encapsulated and transmitted over the network. On the TOP side, the received data can be represented by $Y'_{M\times3}$, which is given in (2.20). The received principal components can be represented as:

$$P'_{6\times3} = \begin{bmatrix} \vec{V}'_1 & \vec{V}'_2 & \vec{V}'_3 \end{bmatrix} \tag{2.31}$$

Simultaneously, a data reconstruction model is deployed on the master side, raising the reduced dimension to its original one. Specifically, PCA is an invertible algorithm which allows the data with reduced dimensions to recover its original dimensions via a simple calculation. As is given below, the reconstructed data X_R can be calculated as either (2.32) or (2.33):

$$X_{R_{M\times6}} = Y'(P')^{-1} \tag{2.32}$$

$$X_{R_{M\times6}} = Y'(P')^{T} \tag{2.33}$$

Therefore, the kinesthetic data with reduced dimensions can not only be transmitted over the network, but it can be recovered easily on the receiver with the assistance of received chosen principal components P'. The case showing in this part has explained the dimensionality reduction and reconstruction process when transmitting the position/velocity data from the OP to the TOP side. On top of that, PCA-based kinesthetic data reduction modules can be used to transmit the force feedback from the TOP to OP as well. The only

difference is the dimensionality of force data M is 3, and the reduced dimensions N can be either 2 in this case.

2.4.4 SAE-based Kinesthetic Data Reduction and Reconstruction Techniques

SAE uses back propagation unsupervised learning to extract features and adapts features to represent high-dimensional input. It tries to learn an identity function to make the input and output signal as consistent as possible.

As is depicted in Fig. 2.10, the proposed dimensionality reduction and reconstruction module is consisted of 5 layers, 1 input layer, 1 output layer, 2 hidden layers and 1 encoded layer. Assuming the input represents the position/velocity data transmitted from the master to slave side, which is given in (2.18). The encoded layer derives the data with reduced dimensions, which refers to the embedding in this module. Additionally, the input and output layer have the same dimensions, which indicates that the commands from the OP to the TOP should remain the same data format. The embedding is the data to be transmitted over the network. In the hidden layer, the data is reduced to a 4-dimensional space at first, and further reduced in the encoded layer. The embedding can be derived from the encoded layer, and the embedding Y is of 3-DoF space, which can be represented as $[\vec{\mathbf{u}}\ \vec{\mathbf{v}}\ \vec{\mathbf{w}}]$.

The reconstruction process is in contrast to the dimensionality reduction process. Correspondingly, it maps from 3-dimensional embedding to a 4-dimensional space in the hidden layer, and outputs the final reconstructed data in the output layer. The reconstruction model of position/velocity data should be deployed on the slave side, which helps reconstructing the embedding to its original dimensions.

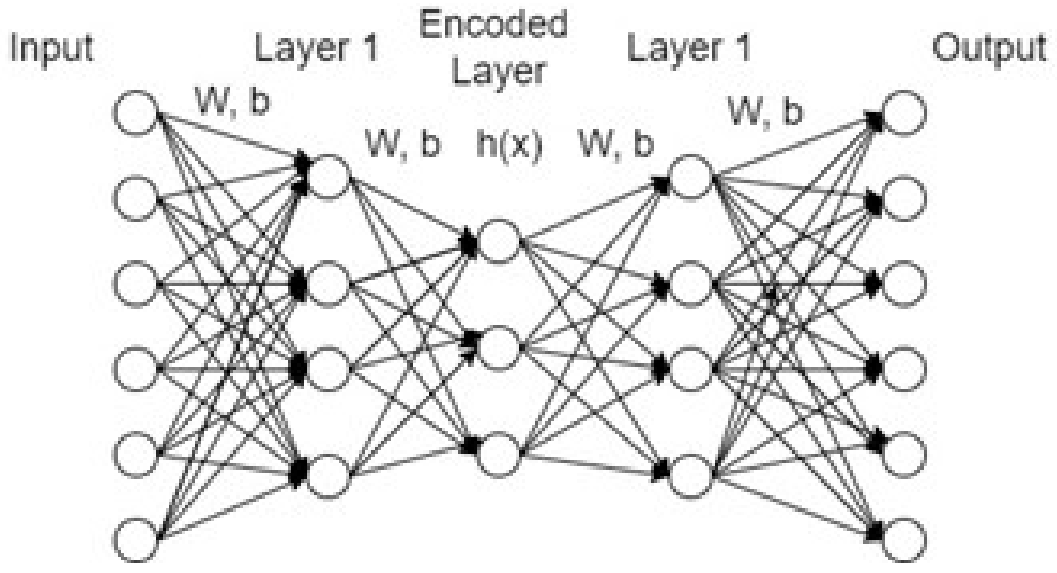


Fig. 2.10 The Proposed SAE-based Dimensionality Reduction and Reconstruction Module

The relationship of the data in the encoded and input layer can be represented as follows:

$$h_{\boldsymbol{W}_{e1},\boldsymbol{W}_{b1},\boldsymbol{W}_{e2},\boldsymbol{W}_{e2}} = f(\boldsymbol{W}_{e2} \cdot f(\boldsymbol{W}_{e1}X + \boldsymbol{b}_{e1}) + \boldsymbol{b}_{e2}) \tag{2.34}$$

$$Y = h_{\boldsymbol{W}_{e1},\boldsymbol{W}_{b1},\boldsymbol{W}_{e2},\boldsymbol{W}_{e2}} \tag{2.35}$$

where $\boldsymbol{W}_{e1}$, $\boldsymbol{b}_{e1}$ are weights and biases for the left hidden layer 1, and $\boldsymbol{W}_{e2}$, $\boldsymbol{b}_{e2}$ are the weights and biases for the encoded layer. The activation function $f(\cdot)$ can usually be set as sigmoid or tanh function.

Similarly, the reconstruction process from the encoded layer to the output layer can be represented as the following formula:

$$X_R = f(\boldsymbol{W}_{d2} \cdot f(\boldsymbol{W}_{d1}Y + \boldsymbol{b}_{d1}) + \boldsymbol{b}_{d2}) \tag{2.36}$$

where $\boldsymbol{W}_{d1}$, $\boldsymbol{b}_{d1}$ are weights and biases for the right hidden layer 1, and $\boldsymbol{W}_{d2}$, $\boldsymbol{b}_{d2}$ are the weights and biases for the output layer. Then the learning objective is as the following formula shows:

$$\min L(X, X_R) = \frac{1}{M} \min \|X - X_R\|^2 \tag{2.37}$$

where M is the size of data, and L can be defined as the loss between original input and reconstructed value. In this case, mean squared error (MSE) is used as the loss function. Set $\theta = \{\boldsymbol{W}_{e1}, \boldsymbol{W}_{b1}, \boldsymbol{W}_{e2}, \boldsymbol{W}_{e2}, \boldsymbol{W}_{d1}, \boldsymbol{W}_{f1}, \boldsymbol{W}_{d2}, \boldsymbol{W}_{f2}\}$, then the target optimal objective of the SAE-based data reduction and reconstruction module can be:

$$\theta = \frac{1}{M} \arg\min_{\theta} \min \|X - X_R\|^2 \tag{2.38}$$

Therefore, The training process aims at finding the best value of each parameter in θ to minimize the MSE. When a well-trained network is proposed, the input to the encoded layer can be used as the dimensionality reduction model, meanwhile, the encoded layer to the output can be viewed as the data reconstruction model.

The design of SAE-based position/velocity dimensionality reduction and reconstructions modules have been discussed above. It is not surprised to know that the architecture of 2 modules for the force feedback is alike. The only difference is the design of the hidden layer and encoded layer, as the data input in a 3D space for force data. Assuming the embedding is in a 2D space, then the hidden layer can be ignored. Only 2 neurons are created in the encoded layer.

2.4.5 UMAP-based Kinesthetic Data Reduction and Reconstruction Techniques

UMAP is a non-linear dimensionality reduction technique which consists in constructing a weighted graph that favors the preservation of local distances over global distances. UMAP is constructed from a theoretical framework based in Riemannian geometry and algebraic topology. Four main hyper-parameters are introduced to control the performance of UMAP, which are the number nearest neighbours k, and the minimum distance d_{min}. In machine learning, hyper-parameters are parameters whose value are used to control the learning process. By contrast, the values of other parameters (typically node weights) are derived via training. Normally, hyper-parameters are tuned to improve the performance of the algorithm before training process.

As is given in (2.18), the input dataset transmitted from the master to slave side can be represented as $X_{M\times 6} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M]^T$, where $\boldsymbol{x}_i \in \mathbf{R}^6 | i = 1,2,...,M$ is the i-th sample with 6 dimensions. The first phase of UMAP is viewed as the construction of a weighted k-neighbour graph. Given an input hyper-parameter k, one should compute the k nearest neighbours for each $\boldsymbol{x}_i$ and get the set $\eta_i = \{\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, ..., \boldsymbol{x}_{i_k}\}$ under the metric d, where $\boldsymbol{x}_{i_j}$ represents the j-th nearest neighbour to $\boldsymbol{x}_i$. For instance, d can be the Euclidean distance between two samples. This computation can be performed via any nearest neighbour or approximately nearest neighbour search algorithm. Then for each $\boldsymbol{x}_i$, we will find its nearest neighbour and the distance. Let

$$\rho_i = \min\{d(\boldsymbol{x}_i, \boldsymbol{x}_{i_j}) | 1 \leq j \leq k, 1 \leq i \leq M, d(\boldsymbol{x}_i, \boldsymbol{x}_{i_j}) > 0\} \tag{2.39}$$

and set σ_i to be the value as:

$$\sum_{j=1}^{k} \exp(\frac{-\max(d_{min}, d(\boldsymbol{x}_i, \boldsymbol{x}_{i_j}) - \rho_i)}{\sigma_i}) = \log_2 k \tag{2.40}$$

where ρ_i denotes the local-connectivity constraint which ensures $\boldsymbol{x}_i$ connects to at least one other point with an edge of weight 1, σ_i is a normalization factor defining the Riemannian metric local to the point x_i, and d_{min} is the minimum distance which controls how tightly UMAP is allowed to pack points together. In practice, d_{min}, quite literally, provides the minimum distance apart that points are allowed to be in the low dimensional representation. Low values of d_{min} will result in clumpier embeddings, while large values will focus on the preservation of the broad topological structure instead. In this case, d_{min} is 10^{-4}.

Afterwards, a weighted directed graph (UMAP graph) $\bar{G} = (V, E, w)$ is built, where V denotes the set of vertices, E is the set of edges, and w is the set of weight for each edge.

In this case, V is the input set $X_{M\times 6}$, and $E = \{(\boldsymbol{x}_i, \boldsymbol{x}_{i_j}) | 1 \leq j \leq k, 1 \leq i \leq M\}$. And we define the weight function w as:

$$w(\boldsymbol{x}_i, \boldsymbol{x}_{i_j}) = \exp(\frac{-\max(d_{min}, d(\boldsymbol{x}_i, \boldsymbol{x}_{i_j}) - \rho_i)}{\sigma_i}) \tag{2.41}$$

Due to the whole algorithm is based on Riemannian metric, each point has a local metric associated with it, illustrating that the local metric of different points is incompatible. For instance, $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the distance from x_i to x_j, can be inconsistent with $d(\boldsymbol{x}_j, \boldsymbol{x}_i)$, the distance from $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$. Therefore, one can think of the weight of an edge as akin to the probability that the given edge exists. Moreover, each edge in the graph is directed with different weight. Let $A_{M\times M}$ be the weighted adjacency matrix of $\bar{G}$, which can be represented as:

$$A_{M\times M} = \begin{bmatrix} 0 & w(\boldsymbol{x}_1, \boldsymbol{x}_2) & w(\boldsymbol{x}_1, \boldsymbol{x}_3) & \cdots & w(\boldsymbol{x}_1, \boldsymbol{x}_M) \\ w(\boldsymbol{x}_2, \boldsymbol{x}_1) & 0 & w(\boldsymbol{x}_2, \boldsymbol{x}_3) & \cdots & w(\boldsymbol{x}_2, \boldsymbol{x}_M) \\ w(\boldsymbol{x}_3, \boldsymbol{x}_1) & w(\boldsymbol{x}_3, \boldsymbol{x}_2) & 0 & \cdots & w(\boldsymbol{x}_2, \boldsymbol{x}_M) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w(\boldsymbol{x}_M, \boldsymbol{x}_1) & w(\boldsymbol{x}_M, \boldsymbol{x}_2) & w(\boldsymbol{x}_M, \boldsymbol{x}_3) & \cdots & 0 \end{bmatrix} \tag{2.42}$$

where each element $w(\boldsymbol{x}_i, \boldsymbol{x}_j) | i \neq j$ is the weight of edge $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in the graph $\bar{G}$. If edge $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is not existed in E, then the weight $w(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is 0. In other words, there are only k non-zero values in each row of $A_{M\times M}$. It is clear that A is an unsymmetrical matrix.

In order to merge two discordant edges with weights $w(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $w(\boldsymbol{x}_j, \boldsymbol{x}_i)$ together, we need to compute the probability of at least one edge exists. The combined weight can be calculated as:

$$w((\boldsymbol{x}_i, \boldsymbol{x}_j), (\boldsymbol{x}_j, \boldsymbol{x}_i)) = w(\boldsymbol{x}_i, \boldsymbol{x}_j) + w(\boldsymbol{x}_j, \boldsymbol{x}_i) - w(\boldsymbol{x}_i, \boldsymbol{x}_j) w(\boldsymbol{x}_j, \boldsymbol{x}_i) \tag{2.43}$$

Then we will apply (2.43) to the proposed directed graph $\bar{G}$. Then the corresponding undirected weighted graph G whose adjacency matrix B can be represented as:

$$B = A + A^T - A \circ A^T \tag{2.44}$$

where $\circ$ denotes the Hadamard (or pointwise) product. For each element A_{ij} of A, it can be interpreted as the probability that directed edge from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ exists, then $B_i j$ in B is the probability that at least one of the two directed edges (either from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ or from $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$) exists. Hence, a new undirected weighted UMAP graph G is sketched, whose adjacency matrix is given by B.

Lastly, the coordinates $\boldsymbol{y}_i \in \mathbf{R}^3, i = 1, 2, \ldots, M$ of the data points in the lower dimensional space (3D space) can be defined by the force-directed graph layout algorithm, which

utilizes a set of attractive forces F^a applied along edges and a set of repulsive forces F^r applied among vertices. Any force-directed layout algorithm requires a description of both the attractive and repulsive forces. This algorithm proceeds by iteratively applying attractive and repulsive forces at each edge or vertex, which amounts to a non-convex optimization problem. By slowly decreasing the attractive and repulsive forces, the convergence to a local minimum is guaranteed. The attractive and repulsive force between vertices i and j can be represented as:

$$F^a_{i,j} = \frac{-2ab\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^{2(b-1)}}{1+\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^{2(b-1)}} w(\boldsymbol{x}_i, \boldsymbol{x}_j)(\boldsymbol{y}_i - \boldsymbol{y}_j) \tag{2.45}$$

$$F^r_{i,j} = \frac{2b(1-w(\boldsymbol{x}_i, \boldsymbol{x}_j))(\boldsymbol{y}_i - \boldsymbol{y}_j)}{(\varepsilon + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)(1+a\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^{2b})} \tag{2.46}$$

where a, b, ε are hyper-parameters.

The forces given above are derived from gradients optimizing the edge-wise cross-entropy between the weighted graph G, and an equivalent weighted graph H constructed from the points $\{\boldsymbol{y}_i | i = 1, 2, \ldots, M\}$. That is, we are seeking to position points $\boldsymbol{y}_i$ such that the weighted graph H induced by those points most closely approximates the graph G , where we measure the difference between weighted graphs by the total cross entropy over all the edge existence probabilities. Since the weighted graph G captures the topology of the source data, and thus a good low-dimensional representation of the overall topology of the data is provided by the equivalent weighted graph H constructed from $\{\boldsymbol{y}_i | i = 1, 2, \ldots, M\}$.

Therefore, there are 2 phases to construct the graph H. The first phase is to build the fuzzy topological representation, as is described above. The second phase is to simply optimize the low-dimensional representation to have as close a fuzzy topological representation as possible, as measured by cross-entropy.

For the phase 1, spectral embeddings in low dimensional space should be initialized. As the weighted undirected graph G and its adjacency matrix $B_{M\times M}$ are given, we can derive the degree matrix $D_{M\times M}$ for G as follows:

$$D_{M\times M} = \begin{bmatrix} k & 0 & 0 & \cdots & 0 \\ 0 & k & 0 & \cdots & 0 \\ 0 & 0 & k & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & k \end{bmatrix} \tag{2.47}$$

This matrix is always diagonal. The degree of a vertex $\boldsymbol{y}_i$ is the i-th element in the diagonal, which represents the number of edges such that $\{\boldsymbol{y}_i, \boldsymbol{y}_j\}$ is an edge for any $\boldsymbol{y}_j$ also in the vertex set. Then the Laplacian matrix of G can be derived as:

$$L_{M\times M} = D_{M\times M} - A_{M\times M} \tag{2.48}$$

Then decompose the Laplacian matrix L by:

$$L_{M\times M} = D^{1/2} L^{sys} D^{1/2} \tag{2.49}$$

where L^{sys} is called the symmetrically normalized Laplacian matrix. The elements of L^{sys} are thus given by:

$$L^{sym}_{i,j} = \begin{cases} 1, & \text{if } i = j \text{ and } D_{i,i} \neq 0 \\ -\dfrac{1}{\sqrt{D_{i,i} D_{j,j}}}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{2.50}$$

Then the eigenvectors and eigenvalues of $L_{M\times M}$ can be calculated as:

$$L_{M\times M} \vec{V}_{M\times 1} = \lambda \vec{V}_{M\times 1} \tag{2.51}$$

By sorting the eigenvalues from large to small, the top 3 eigenvalues out of 6 can be found. Correspondingly, the eigenvectors $[\vec{V}_1 \ \vec{V}_2 \ \vec{V}_2]_{M\times 3}$ are used to form the initialization of $Y_{M\times 3}$. As long as the embeddings in H are derived, the weight of edges also needs to be derived. A smooth approximation of the membership strength between two points in the low dimensional space can be defined as:

$$\Phi(\boldsymbol{y}_i, \boldsymbol{y}_j) = \frac{1}{1 + a(\|\boldsymbol{y}_i - \boldsymbol{y}_j\|)^{2b}} \tag{2.52}$$

where a, b are parameters to be trained to fit against the curve Ψ, which is given as:

$$\Psi(\boldsymbol{y}_i, \boldsymbol{y}_j) = \begin{cases} 1 & , \text{ if } \|\boldsymbol{y}_i - \boldsymbol{y}_j\| \leq d_{min} \\ \exp(-(\|\boldsymbol{y}_i - \boldsymbol{y}_j\|) - d_{min}) & , \text{ otherwise} \end{cases} \tag{2.53}$$

where d_{min} is the preferred minimum distance between 2 vertices given in 2.40 and 2.41. The reason for fitting the curve Φ against Ψ, instead of using Ψ directly, is saving the time and memory cost. Also, Φ has been tested to be the most efficient in the curve family [49]. Furthermore, $a \approx 1.929$, $b \approx 0.7915$, and $d_{min} = 0.001$ are the default values.

Then we need to optimize the positions of the embeddings in the lower dimension so that the weighted graph H induced by those points most closely approximates the graph G. In this case, we measure the difference between 2 weighted graphs by the total cross entropy over all the edge existence probabilities. As is described above, the probability of

an edge exists between 2 points i and j in H can be derived from (2.52), which can be represented as:

$$v(\boldsymbol{y}_i, \boldsymbol{y}_j) = \frac{1}{1 + a(\|\boldsymbol{y}_i - \boldsymbol{y}_j\|)^{2b}} \tag{2.54}$$

Then we simplify the notations, $w(\boldsymbol{y}_i, \boldsymbol{y}_j)$ to $w_{i,j}$, representing the weight of edge $\{\boldsymbol{y}_i, \boldsymbol{y}_j\}$ in the low dimensional space, and $v(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to $v_{i,j}$, the weight of edge $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$ in the original space. Hence, the cross entropy can be derived as:

$$C_{UMAP} = \sum_{j \neq i, j=1}^{M} \sum_{i \neq j, i=1}^{M} \left[v_{ij} \log_2(\frac{v_{ij}}{w_{ij}}) + (1 - v_{ij}) \log_2(\frac{1 - v_{ij}}{1 - w_{ij}}) \right] \tag{2.55}$$

Finally, by using stochastic gradient descent (SGD) with respect to C_{UMAP} in python, the embeddings in the low dimensional space (i.e., $\boldsymbol{y}_i, 1 \leq i \leq M$) are optimized. The optimization process involves minimizing a cost function that measures the discrepancy between the high-dimensional and low-dimensional representations of the data. SGD works by iteratively adjusting the parameters of the low-dimensional representation to minimize the cost function. The algorithm does this by computing the gradient of the cost function with respect to the parameters, and updating the parameters in the direction of the negative gradient. Theoretically, one can use SGD to approach the minimum of any differentiable function [63].

However, UMAP is an inevitable DRT, the data with reduced dimensions will be reconstructed from an LSTM network with multiple layers and neurons. The functional API of Tensorflow in Python can handle LSTM networks with arbitrary layers and neurons in each layer, even multiple inputs or outputs [64], which is applicable in reconstructing data. As is shown in Fig. 6, an example of the an LSTM model, the input is in 3D space $\mathbf{R}^3$, output is in 6-dimensional space $\mathbf{R}^6$, and four hidden layers with different neurons. The MSE of the model is the loss function of this model, which is same with (2.37). Functional API is flexible for creating models with different layers and neurons. The layers and neurons can be added if a higher and more precise reconstructed data is required.

In addition, UMAP can be used to reduce the dimensionality of force feedback in the same way. The force data with reduced dimensions is transmitted from TOP to OP, and the data reconstruction module on the OP side recovers the received data to original dimensions by another LSTM network.

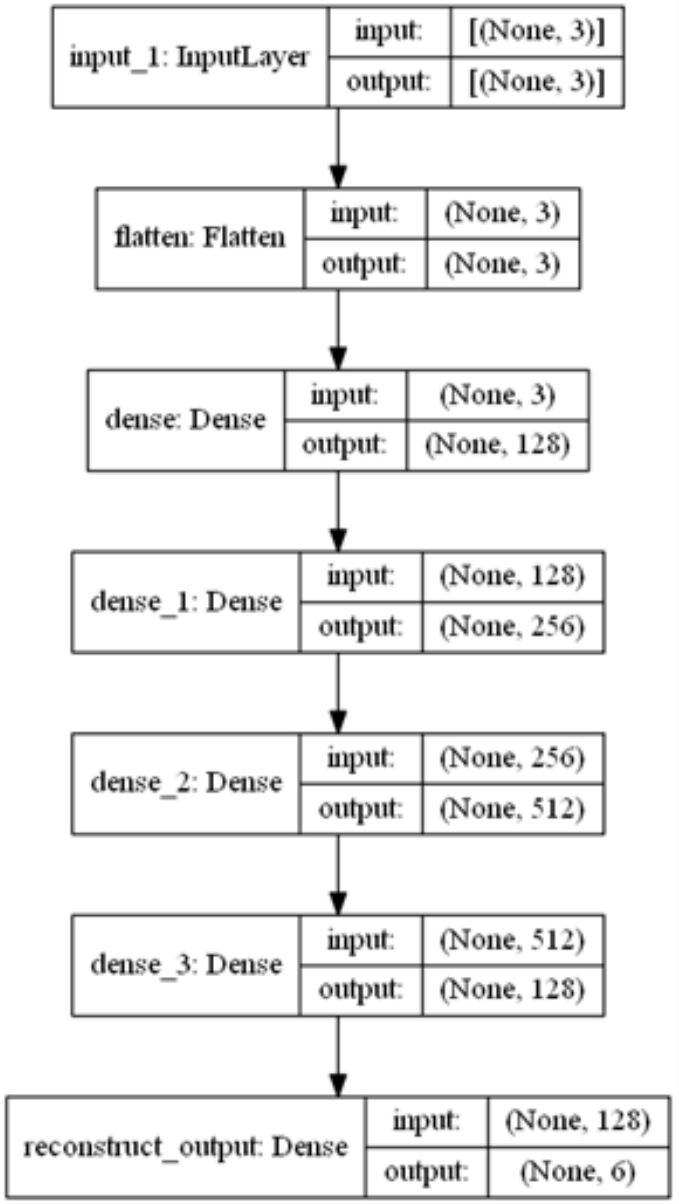


Fig. 2.11 An Example of Functional API for Reconstructing Data From UMAP

2.5 A Bilateral Teleoperation System with Unsupervised Clustering Algorithms for Kinesthetic Data Reduction

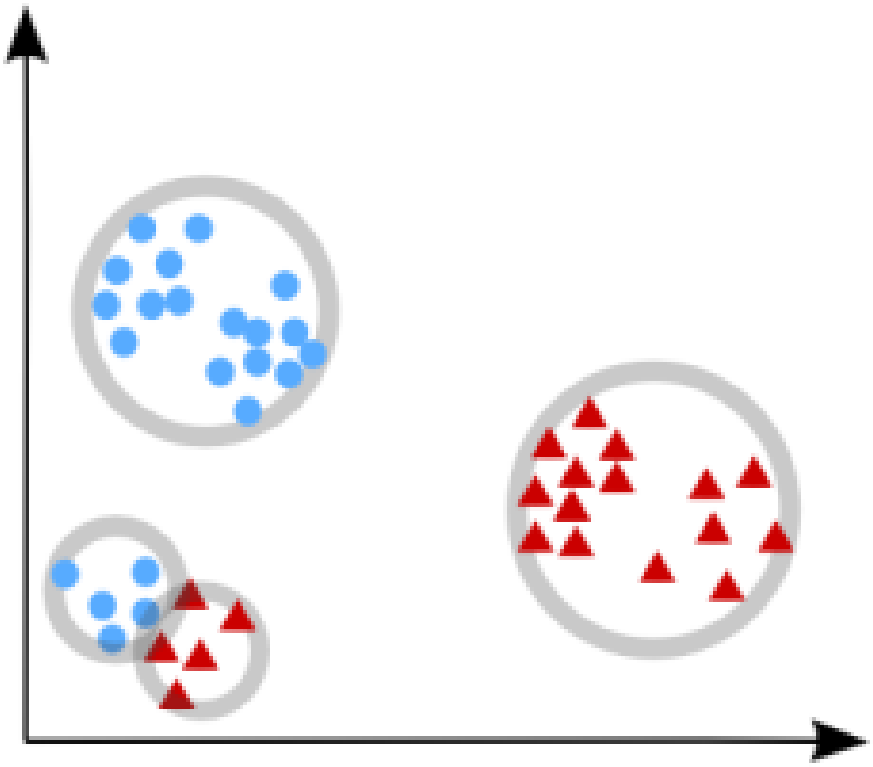
2.5.1 Motivation

Recalling the aforementioned kinesthetic data reduction techniques applying in bilateral systems in Section 2.3, a novel mathematical model for selectively transmitting kinesthetic data is proposed. By using original data as input and transmitting labels as output, a well-trained LSTM network can be derived. However, as is given in (2.17), transmitting labels are derived from Weber's law of JND, illustrating that the newly proposed kinesthetic data reduction model expends applicability of dealing with time series kinesthetic data. LSTM network, as a type of supervised learning, focuses on building machine learning models on labelled datasets. It is somehow arduous to produce labelled datasets where the original kinesthetic data is unlabelled. Furthermore, even if the labelling is done, the speed of labelling is much slower than the speed of data production. Although this time cost does not affect real-time label predictions in LSTM networks, the labelling process on original kinesthetic data does increase the training time of the entire data reduction modules. So, to classify unlabelled data, unsupervised learning algorithms are used.

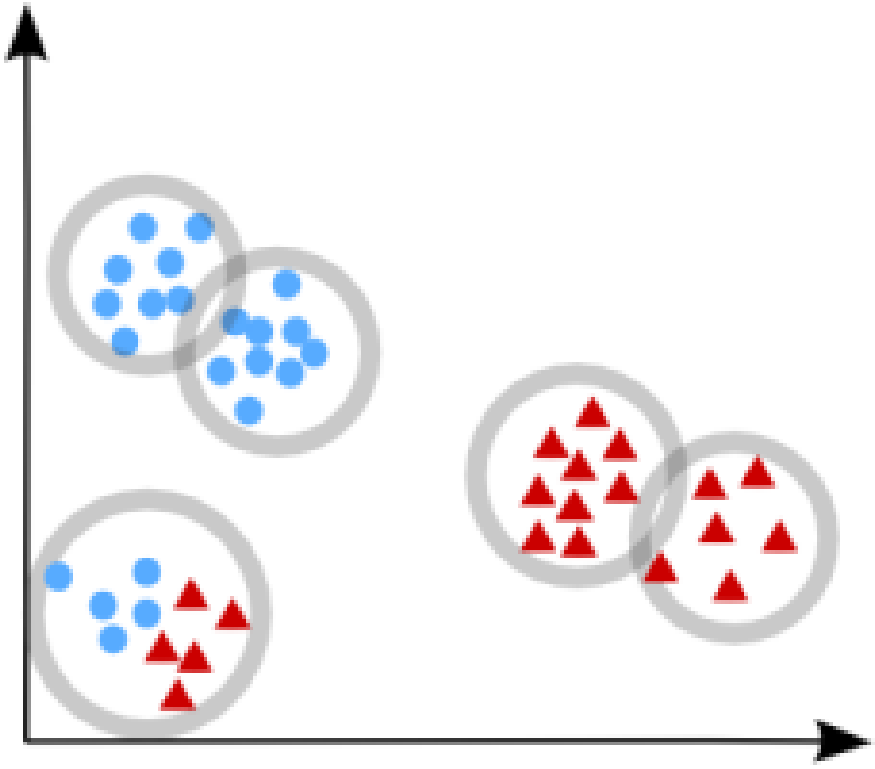
This section aims at using unsupervised methods to segregate the kinesthetic data into different clusterings so that data belonging to certain groups is transmitted over the network, while the other part is not transmitted. Clustering is the process of dividing samples into categories by the intrinsic relationship between the data without knowing any sample labels in advance, resulting in the high similarity between samples of the same category and low similarity between samples of different categories [73, 74]. It is a main task of exploratory data analysis, and a common technique for statistical data analysis, used in many fields, including pattern recognition, image analysis, information retrieval, bioinformatics, data compression [75–79].

Comparing with the supervised learning, the previous research in [80] has mentioned the advantages of unsupervised learning in 3 aspects: i)It can see what human minds cannot visualize; ii) There is less complexity compared to the supervised learning task; iii) It is reasonably easier to obtain unlabeled data.

As is given in Fig. 2.12, there are 2 labels represented with blue and red colours. Supervised clustering can distinguish homogenous clusters (i.e., the cluster in which there are samples from both classes), however unsupervised clustering algorithm may yield 2 or more clusters from 1 homogenous cluster. Therefore, different clusterings derived by supervised and unsupervised algorithms can effect the transparency of the entire system. Also, various unsupervised clustering algorithms can contribute to different clusterings. Therefore, It is worth further simulating and analysing these 2 situations.



(a) Supervised Clustering Algorithm



(b) Unsupervised Clustering Algorithm

Fig. 2.12 An Example of Clustering Algorithms

2.5.2 K-means Clustering-based Models for Kinesthetic Data Reduction

K-means clustering is one of the most popular unsupervised learning algorithms that solve the well-known clustering problem. It is a vector quantization method, originally from signal processing, which aims at dividing n observations into k clusters in which each observation belongs to the cluster with the cluster centroid, serving as a prototype of cluster. Therefore, the selection of centroid for each cluster is the key of k-means clustering.

Following the position/velocity data given in (2.18), k_p samples are arbitrarily selected as initial cluster centroids in $\mathbf{R}^6$, which can be represented as $[\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{k_p}]$. In the simulation process, we assume $k_p = 2$. Then a Euclidean distance matrix between each sample and each centroid can be represented as:

$$D_{M\times k_p} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,k_p} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,k_p} \\ \vdots & \vdots & \ddots & \vdots \\ d_{M,1} & d_{M,2} & \cdots & d_{M,k_p} \end{bmatrix} \tag{2.56}$$

where $d_{i,j}|i \in [1,M], j \in [1,k_p]$ represents the distance between i-th sample and j-th centroid. $d_{i,j}$ can be calculated as:

$$d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{z}_j\| \tag{2.57}$$

The sample $\boldsymbol{x}_i$ is classified according to the minimum distance principle. For each sample, the minimum distance is d_{i,j_i}, then $\boldsymbol{x}_i$ is classified in to clustering c_j. That is, $\boldsymbol{x}_i \in c_j$.

Then mean value of all the samples belonging to each clustering c_j can be calculated as the new centroid of c_j, which can be calculated as:

$$\boldsymbol{z}_j^{(2)} = \frac{1}{N_i^{(2)}} \sum_{\boldsymbol{x}^{(2)}\in c_j} \boldsymbol{x}^{(2)}, \qquad j = 1,2,...,k_p \tag{2.58}$$

where $N_i^{(2)}$ represents the number of samples belonging to cluster c_j at the second iteration.

Subsequently, the sum of distance between each sample and the centroid of its new cluster, also termed as the clustering criterion function can be calculated as:

$$E^{(2)} = \sum_{j=1}^{k_p} \sum_{\boldsymbol{x}^{(2)}\in c_j} \|\boldsymbol{x}^{(2)} - \boldsymbol{z}_j^{(2)}\|^2 \tag{2.59}$$

Then a new distance matrix can be calculated based on new centroids. Correspondingly, new clustering criterion function can be derived. This iteration ends till E is no longer

changed, or the difference between the preceding two times is less than the set threshold, then the clustering is finished. Otherwise, turn to (2.56) to continue iteration.

After the process mentioned above, original position/velocity data can be automatically divided into k_p clusters. Amongst these clusters, only a portion of data is transmitted over the network. We sort the number of samples in each cluster, then the clusters can be represented as $C = \{c_{n1}, c_{n2}, ..., c_{nk_p}\}$, where the corresponding number in each cluster has a relationship of $N_{n1} > N_{n2} > ... > N_{nk_p}$. Assuming data in top k'_p clusters of C is selected to transmit over the network, that is, $C' = \{c_{n1}, c_{n2}, ..., c_{k'_p}\}$. Then the correspondingly transmitted data from OP to TOP in Fig. 2.13 can be represented as $Y_{M\times 6} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{z}_{k'_p+1}, \ldots, \boldsymbol{z}_{k'_p+2}, \ldots, \boldsymbol{z}_{nk_p}, \ldots, \boldsymbol{x}_M]^T$, where $\boldsymbol{x}_i \in C'$. Therefore, the transmitted data consists of 2 parts, original data within C' and centroids out of C'.

The data is transmitted over the network, and the data recovery module predicts the position/velocity data between 2 receive intervals by following (2.7) that TOP moves uniformly when no data receive. When interaction happens with the remote environment, force data is being collected. The collected force data is divided into k_f clusters by k-means clustering, and only data in top k'_f clusters is transmitted over the network. We can assume that $k_f = 2$. In the same way, as is given in (2.8), the data recovery module on OP side recovers force feedback data during the interval of data receive. It assumes the actual force feedback is equal to the previously received data, which is the value of centroid.

As is shown above, only kinesthetic data in a part of clusters is transmitted over the network, which effectively reduces data transmission in bilateral teleoperation systems.

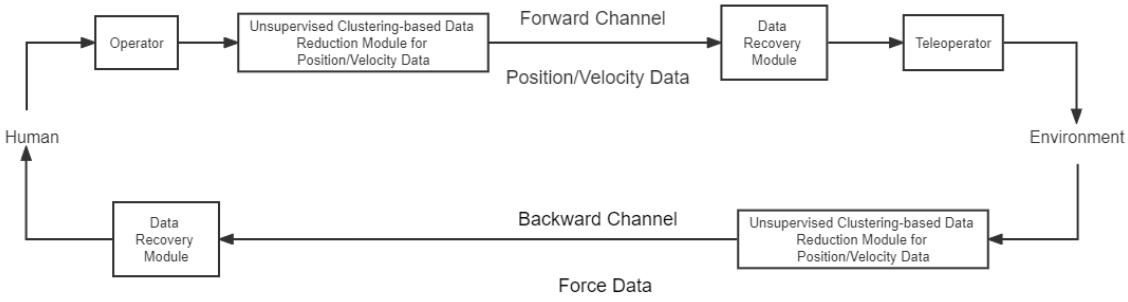


Fig. 2.13 A Bilateral Teleoperation System with Unsupervised Clustering-based Data Reduction Modules

2.5.3 Agglomerative Hierarchical Clustering-based Models for Kinesthetic Data Reduction

Hierarchical Clustering is a kind of clustering algorithm that creates a hierarchical nested clustering tree by calculating the similarity between data points of different categories. In a clustering tree, the raw data points of different categories are the lowest layer of the tree, and the top layer of the tree is the root node of a cluster. In contrast to k-means clustering, which clusters the data itself, the hierarchical clustering algorithm is often used in studies that require analysis of the links and correlations between clusters [81]. The agglomerative algorithm of hierarchical clustering combines the two most similar data points among all data points by calculating the similarity between two types of data points, and iterates this process repeatedly.

In this section, the kinesthetic data reduction modules based on hierarchical clustering are introduced. Simultaneously, data recovery modules are the same as principles mentioned before in Section 2.5.2, which are simply raised here. As long as the kinesthetic data is collected, dissimilarity is measured by distance between pair of observation in the hierarchical clustering , such as Euclidean distance. From (2.18), we can calculate the distance between each samples as:

$$D_{M\times M} = \begin{bmatrix} 0 & d_{1,2} & d_{1,3} & \cdots & d_{1,M} \\ d_{2,1} & 0 & d_{2,3} & \cdots & d_{2,M} \\ d_{3,1} & d_{3,2} & 0 & \cdots & d_{3,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{M,1} & d_{M,2} & d_{M,3} & \cdots & 0 \end{bmatrix} \tag{2.60}$$

where $d_{i,j}|i \in [1,M], j \in [1,M]$ represents the distance between i-th sample and j-th sample. In this case, the Euclidean distance between 2 points can be calculated as:

$$d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \qquad i,j = 1,2,...,M \tag{2.61}$$

It is clear that $D_{M\times M}$ is a symmetric matrix due to each element is a norm. Amongst the elements in $D_{M\times M}$, there exists a minimum distance apart from 0 between $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. Then $\boldsymbol{x}_a$ and $\boldsymbol{x}_a$ form 1 cluster called $c_{a,b}$. The distance between each sample and certain cluster can be calculated from average linkage, which is given as:

$$d_{i,c} = \frac{\sum_{\boldsymbol{x}_j \in c} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{N_c} \tag{2.62}$$

where N_c is the number of samples in cluster c. As a consequence, the new distance matrix $D_{(M-1)\times(M-1)}$ can be calculated as:

$$D_{(M-1)\times(M-1)} = \begin{bmatrix} 0 & d_{1,2} & d_{1,3} & \cdots & d_{1,c_{a,b}} & \cdots & d_{1,M} \\ d_{2,1} & 0 & d_{2,3} & \cdots & d_{2,c_{a,b}} & \cdots & d_{2,M} \\ d_{3,1} & d_{3,2} & 0 & \cdots & d_{2,c_{a,b}} & \cdots & d_{3,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{c_{a,b},1} & d_{c_{a,b},2} & d_{c_{a,b},3} & \cdots & 0 & \cdots & d_{c_{a,b},M} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{M,1} & d_{M,2} & d_{M,3} & \cdots & d_{M,c_{a,b}} & \cdots & 0 \end{bmatrix} \tag{2.63}$$

Then, another minimum value apart from 0 can be selected from (2.63), which contributes to another cluster. By doing this iteration, it continues looping till only one cluster consists of all data points at the root of the tree. That is, there are M clusters each consists of a single data point at the leaves. In other words, when D is a matrix of only 1 column and 1 row, the clustering ends. However, this process can be simplified by given a default value of clustering, which can be the number of preferred clusters. In this case, L is the number of position/velocity data clusters, normally equals to 10.

The selection of clusterings to be transmitted over the network is the same with itself in Section 2.5.2. Only a part of position/velocity data in k_p clusters is transmitted. In this case, we assume $k_p = 4$ The remaining part, such as hierarchical clustering force feedback is similar with (2.60)-(2.63), and the number of clusters to be transmitted k_f can be assumed as 4 as well. However, the force is normally in a 3D space, while position/velocity data has a higher dimensionality.

To sum up, by selecting data in a portion of clusterings to be transmitted, both k-means and hierarchical clustering techniques can effectively reduce the kinesthetic data transmission over the network.

2.6 A Bilateral Teleoperation System with GBDT-Based Predictive Schemes

2.6.1 Motivation

As the aforementioned predictive schemes in Section 2.2.3, force predictors should be designed for the further reduction of kinesthetic data transmission. Predictive coding is proposed to fill the blank of transmission and estimate future haptic data from data previously received. Once the predictors are deployed in the teleoperation system, users on the OP side can sense the force feedback from the predictor without any transmission delay. However, the precondition of this assumption is that the proposed predictors should be precise enough, without degrading the transparency of the whole system. From the previous research, ZOH predictor, FOLP and the third-order predictor estimates the force feedback from previous 1 or 2 data samples, which is not applicable to deal with a time series of force feedback [27–29]. The previous techniques focused on the kinesthetic data reduction duting the transmission time, while the loss of transparency due to the reduction was not being considered. It is hard to believe that the future feedback can be predicted from previous 1 or 2 samples without the degrading transparency. Therefore, the rate of transmission and predictive accuracy should be considered simultaneously when desigining the predictive scheme.

In order to increase the whole accuracy of the prediction in the whole system, GBDT is introduced as its good performance in many prediction problems, including online advertising [38], search ranking [39], and instance transfer [40]. From the previous research, GBDT has been proved to provide predictive accuracy that other algorithms cannot be trumped [65]. GBDT is a machine learning technique for optimizing the predictive value of a model through successive steps in the learning process. Each iteration of the decision tree involves adjusting the values of the coefficients, weights, or biases applied to each of the input variables being used to predict the target value, with the goal of minimizing the loss function (the measure of difference between the predicted and actual target values). The gradient is the incremental adjustment made in each step of the process; boosting is a method of accelerating the improvement in predictive accuracy to a sufficiently optimum value.

As is shown in Fig. 2.14, an example of GBDT algorithm, it is clear to see that the final output is consisted of several different trees. The labels of each tree should be the loss between the actual and the predicted value. In each iteration, a new tree is delivered based on the that loss. In one tree, each branch is split by a specific condition, for instance, to minimize the sum of square error of different ways to split. Once a well-trained tree is

delivered, the predictive value is the sum of each tree plus the original value of the weak learner and the loss in each tree.

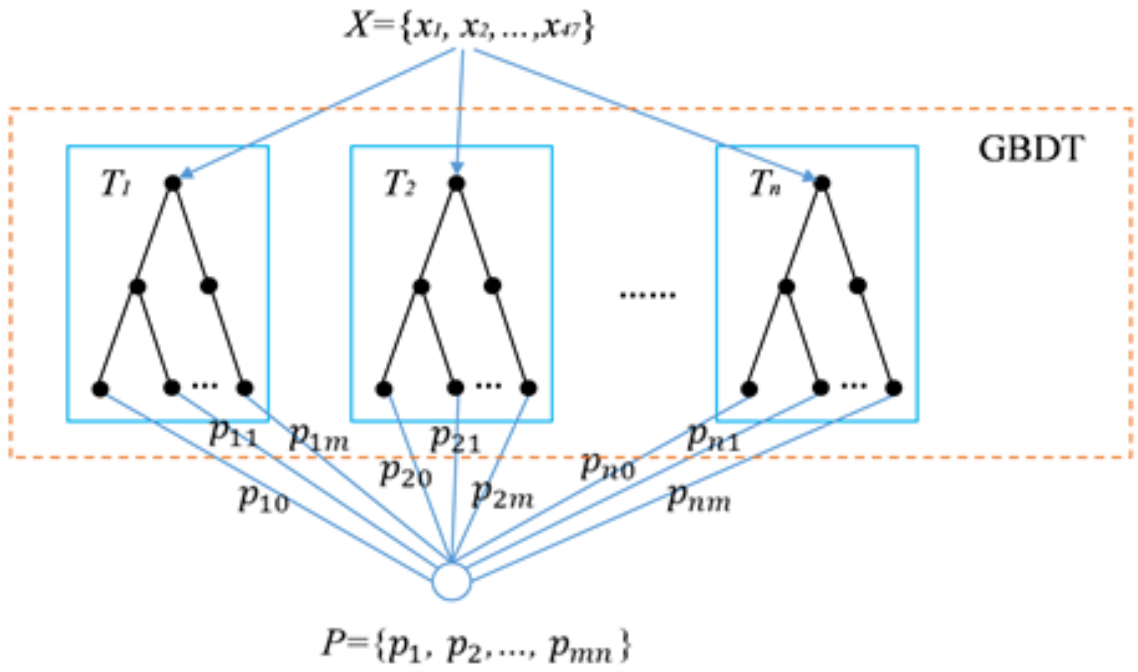


Fig. 2.14 Overview of GBDT Algorithm

2.6.2 System Model

The proposed system is depicted in Fig. 2.15. The design of data reduction and recovery modules have been introduced in Section 2.2& 2.3. The data reduction module can either be reducing the amount of transmission or the size of each packet. And the data recovery (data construction) module transfers the discrete received data to the original form collected on the OP side. It assumes the OP moves at a constant velocity during the interval of data receive, then sends the command to the TOP side. Additionally, 2 same and well-trained GBDT modules are deployed on the both OP and TOP sides. The update control module is a JND-based verifier, which determines whether the current GBDT module should be updated or not.

As long as one triggers the OP, a sensor starts collecting the position and velocity data of the system, which can be represented as $(x_0, x_1 \ldots, x_i, \ldots, x_M)$ and velocity $(\dot{x}_0, \dot{x}_1 \ldots, \dot{x}_i, \ldots, \dot{x}_M)$. However, only a part of it is transmitted over the network, which is $(x_0, \ldots, x_i, \ldots, x_j, \ldots, x_N)$ and $(\dot{x}_0, \ldots, \dot{x}_i, \ldots, \dot{x}_j, \ldots, \dot{x}_N)$, where $N \leq M$. Simultaneously, human can feel the force feedback predicted from the GBDT module on the master side, which can be represented as:

$$F_i' = \Lambda_0(x_i, \dot{x}_i | X_{train}) \tag{2.64}$$

where F_i' represent the i-th predicted force data on the master side, x_i and $\dot{x}_i$ is the current position/velocity data. Λ_0 is the first well-trained data based on previously collected data set X_train. Similarly, the GBDT module deployed on the TOP side can also be represented as Λ_0. In order to make sure the transparency of the whole system retain at a high level, the module on 2 sides should be updated once the degradation is detected. Later, the position and velocity data is received on the TOP side. Correspondingly, the reconstructed data is acquired from the data recovery module, which can be represented as $X_r = (\tilde{x}_0', x_1', \ldots, \tilde{x}_i', x_{i+1}', \ldots, \tilde{x}_j', x_{j+1}', \ldots, x_M')$ and $\dot{X}_r = (\tilde{\dot{x}}_0', \dot{x}_1', \ldots, \tilde{\dot{x}}_i', \dot{x}_{i+1}', \ldots, \tilde{\dot{x}}_j', \dot{x}_{j+1}', \ldots, \dot{x}_M')$, where $\tilde{x}_i', \tilde{x}_j'$ and $\tilde{\dot{x}}_i', \tilde{\dot{x}}_j'$ are the receiving position and velocity data at time i and j correspondingly. The relationship data between two receive interval has been given in (2.7). The TOP moves in the manner of the reconstructed data, and starts interacting with the environment. Therefore, a time series of force feedback is collected, which can be interpreted as $F = (F_0, F_1 \ldots, F_i, \ldots, F_M)$. At the same time, the update control module compares the real force feedback with the predicted force feed back from the GBDT module. The predicted force feedback from the GBDT module on the TOP side can be derived from:

$$F_i'' = \Lambda_0(X_r[i], \dot{X}_r[i] | X_{train}) \tag{2.65}$$

where $X_r[i]$ and $\dot{X}_r[i]$ represent the i-th position/velocity data from data recovery module.

Afterwards, the update control module triggers the model update as long as the difference between the predicted and actual force is perceivable to human OP, which can be defined as the following expression:

$$\begin{cases} k_f = \frac{\|F_i - F_i''\|}{\|F_i\|} > \alpha_f & , \text{ Model Updates} \\ k_f = \frac{\|F_i - F_i''\|}{\|F_i\|} \leq \alpha_f & , \text{ Model Remains} \end{cases} \tag{2.66}$$

where α_f is the Weber's parameter for force, normally equals to 0.1.

Assuming the model needs to be updated at the k-th force data, then the new GBDT model can be updated from the training data set with this new data, which can be written as $\Lambda_1 : \{X_{train}, (X_r[k], \dot{X}_r[k], F_k)\}$. By extending this to n-th model updates, the corresponding GBDT model can be represented as $\Lambda_n : \{X_{train}, (X_r[k], \dot{X}_r[k], F_k), ..., (X_r[k_n], \dot{X}_r[kk_n], F_{k_n})\}$. During the model update process, the parameters to define the GBDT is transmitted over the network from TOP to OP side, including leaf nodes, branches, the split condition, etc..

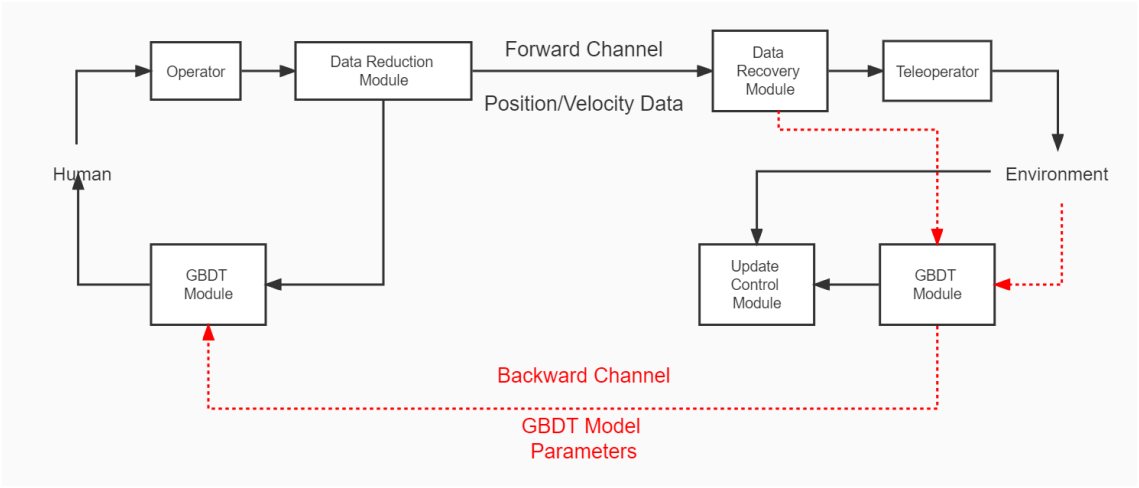


Fig. 2.15 The Bilateral Teleoperation System with GBDT-based Predictive Modules

2.6.3 Design of GBDT-based Predictive Modules

The proposed GBDT-based predictive scheme, which concentrates on establishing a direct mathematical map between position/velocity data and force feedback, aims at increasing the perceptual transparency without a higher packet rate. Assuming that there exists a training data set P of the OP's velocity and position, which is successfully shared with TOP. The motion of TOP defers to P and the corresponding data set U of force feedback from the environment can be understood as a specific mathematical map based on P. Therefore, a joined data set of P and U can be written as:

$$X_{train_{R\times 9}} = \left[\begin{array}{cc|c} \boldsymbol{x}_{p_1} & \dot{\boldsymbol{x}}_{p_1} & \boldsymbol{f}_{u_1} \\ \boldsymbol{x}_{p_2} & \dot{\boldsymbol{x}}_{p_2} & \boldsymbol{f}_{u_2} \\ \vdots & \vdots & \vdots \\ \boldsymbol{x}_{p_R} & \dot{\boldsymbol{x}}_{p_R} & \boldsymbol{f}_{u_R} \end{array}\right] \tag{2.67}$$

where R is the size of training data, $\boldsymbol{x}_{p_i}$ and $\dot{\boldsymbol{x}}_{p_i}$ represent the i-the position/velocity data in P, and $\boldsymbol{f}_{u_i}$ is the i-the force data in U. Additionally, all of the kinesthetic data are in 3D space, which can be written as $\boldsymbol{x}_{p_i}, \dot{\boldsymbol{x}}_{p_i}, \boldsymbol{f}_{u_i} \in \mathbf{R}^3$.

The GBDT model is an addition expression composed of K basis,

$$\hat{\boldsymbol{f}}_i = \sum_{m=1}^{K} F_m(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i), F_m \in N \tag{2.68}$$

in which N is the function space of total basis models.

A loss function L reflects on the deviations between the predictive and actual values. The most commonly used loss function is the squared loss function, which is given as follows:

$$L = \sum_{i=1}^{R} l(\boldsymbol{f}_i, \hat{\boldsymbol{f}}_i) = \frac{1}{2} \sum_{i=1}^{R} (\boldsymbol{f}_i - \hat{\boldsymbol{f}}_i)^2 \tag{2.69}$$

A fabulous model is a balancing act of minimizing the deviations and suppressing the overfitting of the model. Therefore, a regularized item for concealing the complexity is introduced to the objective function Obj,

$$Obj = \sum_{i=1}^{R} l(\boldsymbol{f}_i, \hat{\boldsymbol{f}}_i) + \Omega(F_m) = \frac{1}{2} \sum_{i=1}^{R} (\boldsymbol{f}_i - \hat{\boldsymbol{f}}_i)^2 + \Omega(F_m) \tag{2.70}$$

where Ω represents the complexity of the basis. The complexity of a boosting tree is subject to its depth and leaf nodes. If the number of leaf nodes in a tree is M and the value of each node is w, the regular item for defining its complexity is

$$\Omega(F_m) = \gamma M + \frac{1}{2} \lambda \sum_{j=1}^{M} w_j^2, \qquad \gamma, \lambda \in R \tag{2.71}$$

related to M and the norm of the features in the nodes, where γ, λ are constants.

GBDT is a forward optimization algorithm that the model is gradually established to optimize the objective function from beginning to end. At t-th iteration, a regression tree F_t is constructed to minimize the following regularized loss as the objective function:

$$Obj^{(t)} = \sum_{i=1}^{R} l(\boldsymbol{f}_i, \hat{\boldsymbol{f}}_i^{(t)}) + \Omega(F_t) \tag{2.72}$$

$$\hat{\boldsymbol{f}}_i^{(t)} = \sum_{m=1}^{t-1} F_m(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) = \hat{\boldsymbol{f}}_i^{(t-1)} + F_t(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) \tag{2.73}$$

Then the second-order approximation of the objective function by using Taylor series can be written as:

$$Obj^{(t)} \approx \sum_{i=1}^{R} \left[l(\boldsymbol{f}_i, \hat{\boldsymbol{f}}_i^{(t-1)}) + g_i F_t(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) + \frac{1}{2} h_i F_t^2(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) \right] + \Omega(F_t) \tag{2.74}$$

where $g_i = \frac{\partial l(\boldsymbol{f}_i, \hat{\boldsymbol{f}}_i^{(t)})}{\partial \hat{\boldsymbol{f}}_i^{(t-1)}}$ and $h_i = \frac{\partial^2 l(\boldsymbol{f}_i, \hat{\boldsymbol{f}}_i^{(t)})}{\partial \hat{\boldsymbol{f}}_i^{2(t-1)}}$ are the first and second-order derivatives of the loss function. There is a constant item $l(\boldsymbol{f}_i, \hat{\boldsymbol{f}}_i^{(t-1)})$ which does not affect the objective so

that we can ignore that item and rewrite (2.66) as:

$$\widetilde{Obj}^{(t)} = \sum_{i=1}^{R} \left[g_i F_t(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) + \frac{1}{2} h_i F_t^2(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i) \right] + \Omega(F_t) \tag{2.75}$$

In each regression tree, there exists a function q mapping each sample to one of the leaf nodes. Then we can use an alternative objective function as:

$$\widetilde{Obj}^{(t)} = \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i \right) w_j^2 \right] + \Omega(F_t) \tag{2.76}$$

$$\Omega(F_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \tag{2.77}$$

where $I_j = i | q(x_i) = j$ is the sample set of j-th leaf, and w_j is the assigned weight. T is the number of leaf nodes in F_t, and γ and λ are constant parameters which control the importance of regularizations. For optimizing the objective function, we can set the first-order derivative of (2.76) to be 0, then the derived value of j-th leaf and the corresponding value of the objective function are:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{2.78}$$

$$\widetilde{Obj}^{(t)*} = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{2.79}$$

The empirical loss function is squared loss, then $g_i = \hat{\boldsymbol{f}}_i^{(t-1)} - \boldsymbol{f}_i$, $h_i = 1$. Moreover, for traversing all the features and selecting the best splitting point, the gain of all the possible splits for each leaf node is given by:

$$\Delta Obj = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \tag{2.80}$$

where leaf node I is split into I_L and I_R. The regularization parameter γ can be interpreted as the minimum threshold of the split gain. Leaf nodes splitting is recursively conducted until a predefined depth.

2.7 Conclusions

This chapter has proposed a bilateral teleoperation system with various machine learning algorithms to reduce the kinesthetic data transmission over the network. Additionally, one novel predictive scheme based on GBDT is also given for enhancing the performance of predictive modules in bilateral teleoperation systems. The current data reduction and predictive techniques are shown in Section 2.2. While the proposed systems in Section 2.3 to Section 2.6 will prove very useful in applying different machine learning techniques to bilateral teleoperation systems, such as LSTM networks, DRTs, unsupervised clusterings, and GBDTs. It was the aim to achieve the 1ms-challenge of bilateral teleoperation systems throughout this thesis, from a novel mathematical model to derive the transmission status of each data to a novel idea of reducing the size of kinesthetic data transmission and kinesthetic data clustering. Essential foundations have been given in this thesis, with generally known results kept to a minimum.

In Section 2.2, a brief historical background to Weber's law of JND is given, ranging from its infancy to the PD-based codecs and ZOH/FOLP predictive schemes. Weber's law of JND is a psychological model which calculates the degree to which a human can distinguish between two stimuli. This relationship is linear which normally relates to the magnitude of the reference stimulus. Then, PD-based codecs and two predictive schemes based on the limitations of human perception are proposed. However, it is far more enough to describe the limitations of human perception by a linear model. More accurate mathematical models can thus be proposed, and further enhance the kinesthetic data reduction and predictive modules in bilateral teleoperation systems. Thus, various machine learning algorithms are proposed for this purpose.

In Section 2.3, LSTM networks are applied to solve the problem of selectively transmitting kinesthetic data on top of PD-based codecs. As Weber's law of JND is not the best mathematical model to represent humans' perception deadzone [16–19, 58], a more accurate mathematical model determining the transmission status of kinesthetic data is required to be proposed. Simultaneously, LSTM networks are good at dealing with time-series data, which can be used as the tool to solve the problem. Furthermore, PD-based codecs can help build LSTM networks by labelling the training dataset, i.e., using two different values to represent the transmission status of each data. Then, a well-trained LSTM network can be trained from original data and its corresponding labels by tuning parameters. During the testing process, the LSTM network can predict the transmission status of newly collected data based on last several data within the timestep. And practice has proved that the transmission status of the current data is not only related to the last transmitted data.

In Section 2.4, DRTs and data reconstruction techniques are applied to solve the problem of reducing the size of kinesthetic data transmission over the network. Generally, it is totally a new idea by abandoning selective transmission of data, instead, using machine learning algorithms to reduce the dimensionality of data. Since its novelty, three different DRTs and data reconstruction techniques are proposed to see their performance compared to PD-based codecs and LSTM-based models. DRTs aim at map raw data with high dimensionality to its corresponding embedding in low space. Then, only embeddings will be transmitted from the master to slave side. Data reconstruction techniques help recover embeddings to its original high space. All of three DRTs are popular machine learning algorithms for solving dimensionality reduction problems. And both PCA and SAE are invertible algorithms of which their corresponding data reconstruction algorithms can be derived directly from embeddings. An LSTM network has been build for recovering UMAP embeddings to preferred data.

In Section 2.5, clustering techniques are applied to extend an unsupervised learning solution to the problem of selective transmission of kinesthetic data. Different from output labels required in supervised learning, unsupervised learning tries to separate data into different clusters according to the internal properties of data. Thus, selective transmission of data can be achieved by selecting a part of data clusters for transmission, and data in those clusters outside the selection can be replaced by the centroid of its cluster. In this case, the amounts of clusters and clusters for selective transmission in each clustering technique should be discussed in a further step. However, we can see that unsupervised clustering algorithms can also reduce kinesthetic data transmission theoretically.

In Section 2.6, GBDTs are applied to solve the problem of force prediction in bilateral teleoperation systems. In order to achieve 1-ms challenge, a predictive module is deployed on the master side to predict the forcec feedback from the slave side. However, current predictive schemes are based on Weber's law of JND, which results in high frequency model updates due to inaccuracies. Thus, a more accurate predictive model should be proposed to increase the stability of the whole system. GBDT is the most powerful prediction algorithm, which gives predictions in the form of an ensemble of weak prediction models (typically decision trees). A large amount of data is required to train the model. Predictive models are deployed on the both master and slave side. The predictive model can be updated as well once the perceptual difference is detected.

Chapter 3

Simulation and Performance Analysis of LSTM-Based Kinesthetic Data Reduction Techniques

3.1 Introduction

The vital objective of this section is to investigate the transmission rate and transparency of the proposed mathematical model. The transmission rate and transparency is considered as two criteria to measure the performance of the system model. The haptic perceptually weighted peak signal-to-noise ratio (HPW-PSNR) is introduced to measure the transparency of the model. Furthermore, we compare the proposed model with the traditional PD-based codecs in the situation of the same kinesthetic data sets. The main kinesthetic data collected on the foam is used in the simulation.

In Section 3.2, experiments of kinesthetic data collection are stated in details. The haptic device and tool for establishing virtual different environment are also given in this section. Moreover, the format of collected kinesthetic data is denoted there.

There are 2 metrics for analysing the performance introduced in Section 3.3, which are HPW-PSNR and transmission rate. HPW-PSNR is an objective quality measure which is carefully weighted depending on the perceptual significance of the haptic signal degradation. Additionally, data transmission rate refers to the percentage of collected kinesthetic data to be transmitted over the network. This metric reflects on the impact of data reduction techniques on the amount of data to be transmitted.

In Section 3.4, predicted labels from well-trained LSTM networks are given. Correspondingly, the transmitted data along each axis is also shown. The proposed mathematical model is efficient in reducing the transmitting packet rate of both velocity and force data.

The transmission rate is less than 50% when LSTM-based kinesthetic data reduction models are applied.

Besides, comparison with traditional PD-based codecs is given in Section 3.5. The proposed LSTM-based model has a lower transmission rate for velocity signals than PD-based codecs. Also, both models have similar results on transmission rate of force data when deadband parameter is from 0.1 to 0.15. Otherwise, the proposed model transmits less data than PD-based codecs. In addition, 2 models have similar performance in maintaining transparency, while PD-based codecs have lower degradation for force data, and LSTM-based models degrade less for velocity data.

Performance analysis under dragging is argued in Section 3.6. A higher transmission rate is required when dragging, which is over 76.8% for force data, and 44.6% for velocity data. As a result, the scale of friction is normally smaller than pressure, leading to a different transmission rate.

Results analysis and their significance are given in Section 3.7. Two proposed LSTM-based mathematical models can reduce kinesthetic data transmission in bilateral teleoperation systems effectively. By comparing with current PD-based codecs, proposed models can not only maintain a higher transparency of the system, but reduce kinesthetic data transmission, which satisfies the objectives of our project.

3.2 Kinesthetic Data Collection

All of the collected kinesthetic data is shared in Chapter 3 to 5. Kinesthetic data includes velocity, position, pressure, friction, inertia, etc., which is related to users' movements and the texture of the remote environment. Therefore, the premise of data collection is to collect data on different textures with multiple motions as much as possible. We divide human's motions into two categories, which include dragging and tapping. Tapping, also known as penetration, is when the cursor moves in a direction perpendicular to the plane. And the contact force plays a big role in force data. Also, the position/velicity data variation along the z-axis is more significant than the other 2 axes. Conversely, dragging refers to the lateral movement on the plane. For instance, the cursor moves from point A to B along the plane, where A, B are the points both on the plane. In this case, forces suffered by the OP include friction and pressure. As is illustrated in Section 2.4.1, the relationship between the friction and pressure is linear. On top of that, the kinesthetic data is collected on 4 different textures, which are marble, wood, ceramic tiles and foam. These four materials are various in roughness, viscosity and hardness, which are eligible to represent the complicated and dynamic remote environment. Two metrics used to distinguish different materials are the coefficient of dynamic friction μ and the viscosity

δ. μ is a value that shows the relationship between two objects and the normal reaction between the objects that are involved. We utilized those 4 virtual materials preserved in computer haptics and active interface (CHAI3D) to distinguish the environment, which will be introduced in the following paragraph. Besides, the sampling rate in this experiment is fixed at 1 kHz, which satisfies the requirement of haptic communications.

What's more, the Phantom Omni haptic device is used in the experiment. As is shown in Figure 5, it is a commercial, portable haptic device with 6 DoFs. It is based on a serial architecture, which means that the handle is connected to the housing by a single serial chain. The 6 DoFs include two 3D vectors, i.e. position and force. Moreover, the velocity data of i-th sample can be approximately equal to the average velocity within in sampling interval due to the 1 kHz high rate, which can be expressed by:

$$\vec{v}_i = \begin{cases} 0 \quad , & \text{If } i = 0 \\ \dfrac{\vec{x}_i - \vec{x}_{i-1}}{T_s}, & \text{Otherwise} \end{cases} \tag{3.1}$$

where $T_s = 1$ ms.

Fig. 3.1 A Phantom Omni Device

Additionally, to construct various remote environments and collect alternative data, CHAI3D is introduced in this experiment, as is depicted in Fig 3.2. CHAI3D is an open source set of C++ libraries for computer haptics, visualization and interactive real-time simulation. Written entirely in C++, CHAI3D was designed to make it easier and more intuitive for developers to create applications that combine 3D modelling with force-feedback rendering capabilities. In the past 15 years, CHAI3D has developed into one of the most popular open-source multi-platform haptic simulation frameworks and has been used in a large number of research and production projects in various fields such as games, games, simulators, educational software, and interactive art, scientific visualization and medical applications.

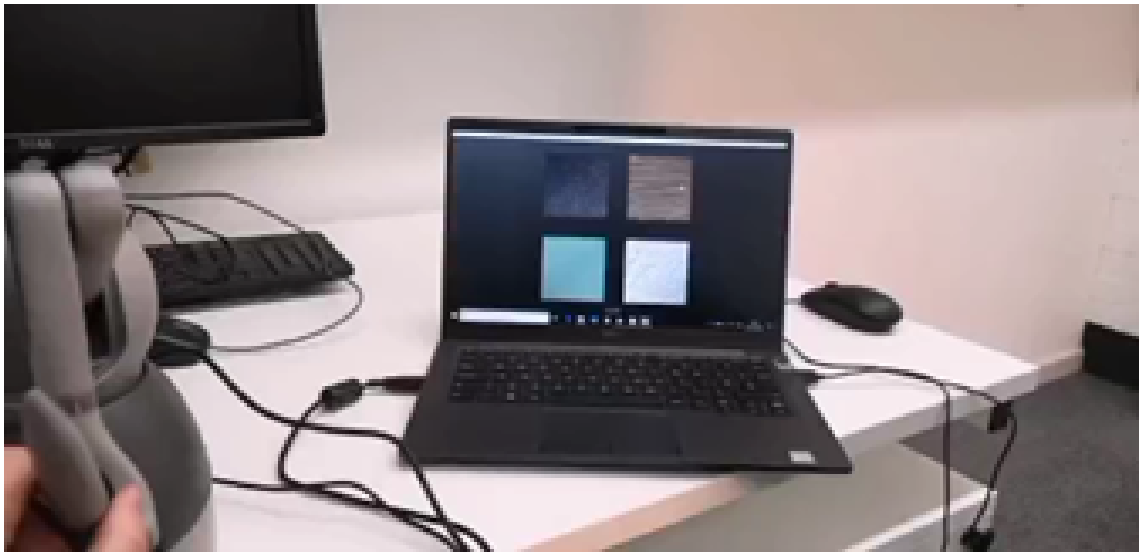


Fig. 3.2 An Experiment of Kinesthetic Data Collection

In each experiment, we collect kinesthetic data for each motion on each texture. Therefore, there are totally eight datasets are collected. During the penetration process, we slowly move the cursor perpendicular to the plane, ensuring minimal lateral displacements. Oppositely, we try to maximize the lateral displacement range of dragging to distinguish it from tapping. The collected haptic data format is given in Fig. 3.3. The standard kinesthetic data in 1ms contains ten degrees, a 3D-force vector, a 3D-velocity vector, a 3D-position vector and the sampling time. The velocity vector can be calculated easily from (3.1) by taking the derivatives of the displacement. We find the displacement of cursor in 1 ms, then calculate the corresponding velocity. The collected kinesthetic data is a time series of data with 1ms sampling interval.

time	SF_x	SF_y	SF_z	MV_x	MV_y	MV_z	MP_x	MP_y	MP_z
1311	0.000000	0.000000	0.000000	-13.175000	-1.121000	-7.581000	0.008278	-0.127982	-0.237417
1312	0.030432	0.062474	0.369221	-13.569000	-2.925000	-8.379000	-0.005291	-0.130907	-0.245796
1313	0.155065	0.265443	0.607589	-10.144000	-2.166000	-8.159000	-0.015435	-0.133073	-0.253955
1314	0.142542	0.133319	0.707640	-4.205000	-3.422000	-6.733000	-0.019640	-0.136495	-0.260688
1315	0.189214	0.218373	0.729449	-0.742000	-3.627000	-4.229000	-0.020382	-0.140122	-0.264917
1316	0.139959	0.103953	0.741903	-0.520000	-3.640000	-3.320000	-0.020902	-0.143762	-0.268237
1317	0.126341	0.181666	0.748237	-0.175000	-1.825000	-2.411000	-0.021077	-0.145587	-0.270648
1318	0.075739	0.143280	0.808117	-2.495000	-2.589000	-6.672000	-0.023572	-0.148176	-0.277320
1319	0.239520	0.273331	0.875451	-3.064000	-3.460000	-8.477000	-0.026636	-0.151636	-0.285797
1320	0.156093	0.092227	0.904975	-1.239000	-2.663000	-6.037000	-0.027875	-0.154299	-0.291834
1321	0.250534	0.244997	0.887166	0.523000	-1.869000	-6.021000	-0.027352	-0.156168	-0.297855
1322	0.187347	0.095001	0.836146	2.154000	-2.903000	-6.015000	-0.025198	-0.159071	-0.303870
1323	0.175207	0.265160	0.785350	2.566000	-3.858000	-4.212000	-0.022632	-0.162929	-0.308082
1324	0.109615	0.125833	0.753176	1.341000	-3.776000	-2.409000	-0.021291	-0.166705	-0.310491
1325	0.182936	0.092805	0.677450	3.144000	-3.919000	0.022000	-0.018147	-0.170624	-0.310469
1326	0.120730	0.015901	0.656347	0.880000	-5.596000	0.615000	-0.017267	-0.176220	-0.309854
1327	0.221868	0.057655	0.631603	1.014000	-3.769000	0.617000	-0.016253	-0.179989	-0.309237
1328	0.135363	0.028669	0.644200	-0.525000	-4.569000	1.515000	-0.016778	-0.184558	-0.307722
1329	0.222922	-0.096131	0.680774	-1.496000	-4.485000	4.232000	-0.018274	-0.189043	-0.303490
1330	0.100974	-0.081573	0.740373	-2.484000	-4.396000	6.940000	-0.020758	-0.193439	-0.296550
1331	0.171241	-0.228468	0.788611	-2.029000	-3.506000	6.327000	-0.022787	-0.196945	-0.290223
1332	0.059456	-0.118843	0.806968	-0.766000	-4.535000	9.356000	-0.023553	-0.201480	-0.280867
1333	0.126599	-0.254145	0.839974	-1.441000	-3.551000	6.617000	-0.024994	-0.205031	-0.274250
1334	-0.004782	-0.203021	1.387641	-22.881000	-41.701000	166.029000	-0.047875	-0.246732	-0.108221
1335	0.096230	-0.065892	1.291197	3.851000	-3.182000	-13.911000	-0.044024	-0.249914	-0.122132
1336	0.069010	0.061441	0.957330	13.995000	-8.971000	-19.475000	-0.030029	-0.258885	-0.141607

Fig. 3.3 An Example of Kinesthetic Data Format

3.3 Performance Metrics

3.3.1 Transparency Test

In a bilateral teleoperation system, transparency refers to whether the mechanical impedance felt by the OP is the same as the impedance of the environment [2]. The measurement of transparency can be divided into subjective measurement and objective measurement. The subjective measurement is how scientists measure what people say. For instance, Weber's law of JND is derived from many subjective measurements. On the flip side, the objective measurement are quantifiable, unbiased, and mathematically calculable. A reliable and quantitative way for measuring transparency signal is termed haptic perceptually weighted peak signal -to-noise ratio (HPW-PSNR) [66]. It is an objective quality metric that carefully measures the perceptual significance of the signal degradation from weighting the difference between the original and distorted signal. It enhances the standard peak-signal to noise ratio (PSNR) measure while incorporating Weber's law of JND. The mathematical formulations are described as follow:

$$MSE = \frac{1}{N}\sum_{n\in N}\|\vec{v}_n - \vec{\hat{v}}_n\|^2 \tag{3.2}$$

$$HPW = \begin{cases} C & , \text{ If } \|\vec{v}_n - \vec{\hat{v}}_n\| \leq \alpha_v\|\vec{v}_n\| \\ k(\|\vec{v}_n - \vec{\hat{v}}_n\| - \alpha_v\|\vec{v}_n\|) + C & , \text{ Otherwise} \end{cases} \tag{3.3}$$

$$HPW - PSNR = 10\log_{10}(\frac{\|\vec{v}_{max} - \vec{v}_{min}\|^2}{MSE \times HPW}) \tag{3.4}$$

where $\vec{v}_{max}$ and $\vec{v}_{min}$ are the maximum and minimum values of the original haptic signal $\vec{v}$, and $\vec{\hat{v}}$ represents the distorted signal. Haptic perceptually weight (HPW) denotes a haptic perceptual weighting function. It is a pairwise function which consists of two parts. When the transmitted signal is within the deadband of the original signal, HPW is a constant (typically, $C = 1$). Otherwise, k is a penalty factor that weights haptic degradations beyond the JND. The range of k is in the interval $[0, 1]$. MSE represents the mean square function of the original and transmitted signals. As is given in Table 2.1, the human perceptual discrimination for velocity is 8%-12%. We assume $\alpha_v = 0.1$ in (3.3). Similarly, (3.3) and (3.4) can be used to do the transparency test of force signals. Normally, $\alpha_f = 0.1$.

3.3.2 Data Transmission Rate

Data transmission rate refers to the percentage of collected kinesthetic data to be transmitted over the network. Due to the proposed data reduction techniques, only a part of the original data is transmitted. This metric reflects on the impact of data reduction techniques on the amount of data to be transmitted. It is easy to derive the following relationship as:

$$R = \frac{N_{transmitted}}{N} \times 100\% \tag{3.5}$$

where $N_{transmitted}$ refers to the data to be transmitted derived from the data reduction module, and N is the amount of collected kinesthetic data.

3.4 Transmitted Signals From LSTM-based Data Reduction Techniques

In this section, the transmitted label prediction is given at first. Then the transmitted and real velocity/force signals are compared. We give an example compare the real and transmitted kinesthetic data when tapping on the foam surface. The label prediction is given in Fig. 3.6, which explicitly shows the predicted label from the proposed LSTM-based mathematical model. When label prediction is -1, the corresponding collected velocity data is not transmitted. The newly collected velocity data is only transmitted when the label is 1. As is given in Table 2.2, the normalized label prediction is a series of float number, which lies between $[-0.5, 0.5]$. Then a threshold is introduced to transfer these float numbers to integers, which is give as bellow:

$$\bar{Y}[i] = \begin{cases} -1, & \text{If } Y[i] \leq 0 \\ 1, & \text{If } Y[i] > 0 \end{cases} \tag{3.6}$$

where $Y[i]$ is the predicted i-th transmission label derived from the LSTM-based model, and $\bar{Y}[i]$ is the corresponding transmission flag. After the de-normalizing, the prediction label becomes a set of -1 and 1.

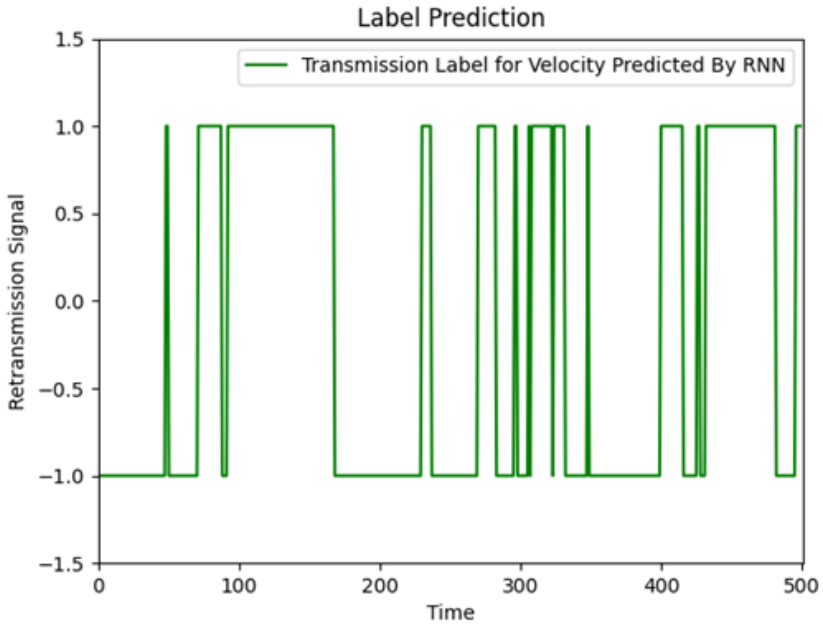
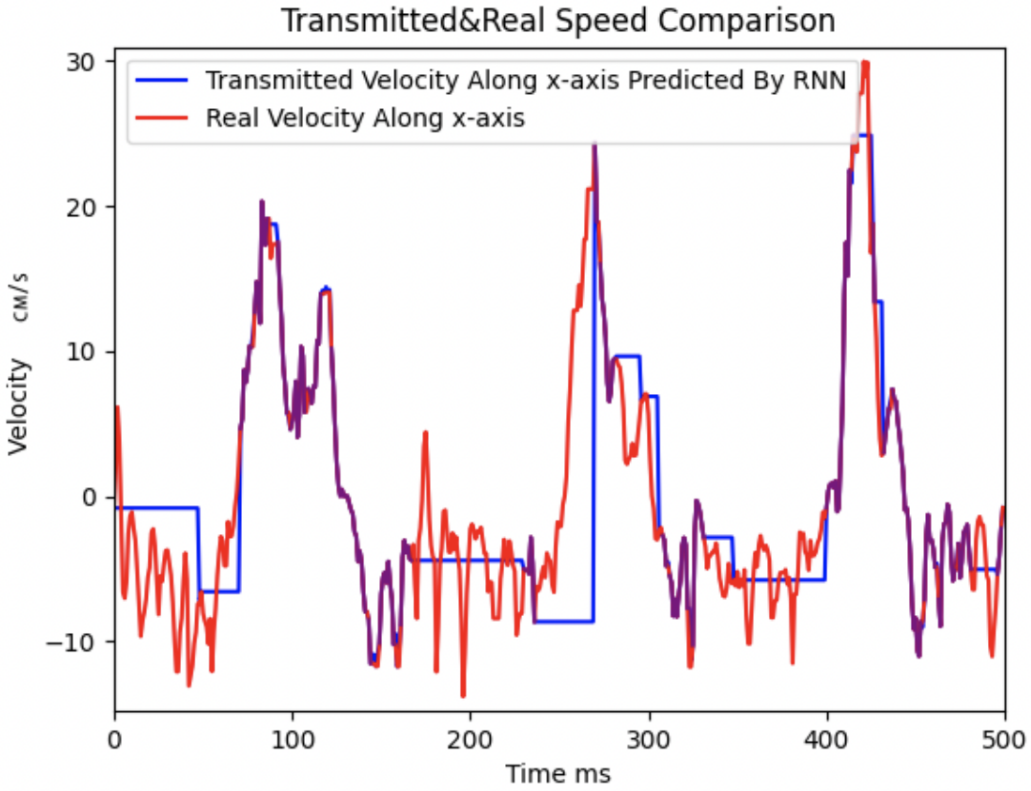
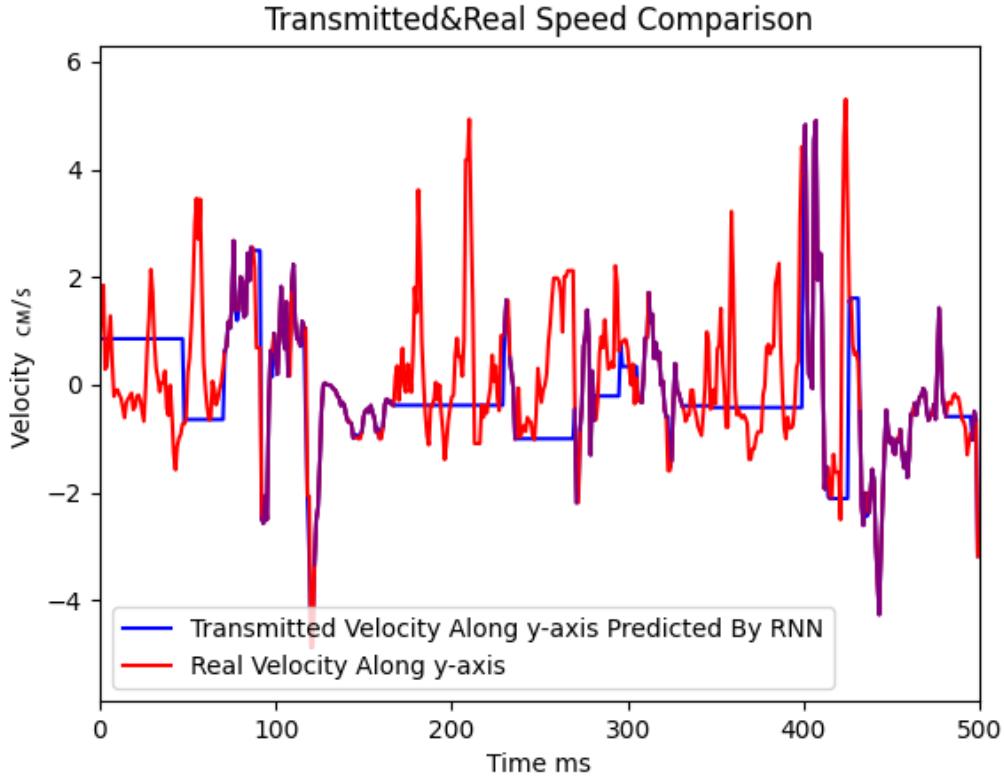


Fig. 3.4 An Experiment of Kinesthetic Data Collection

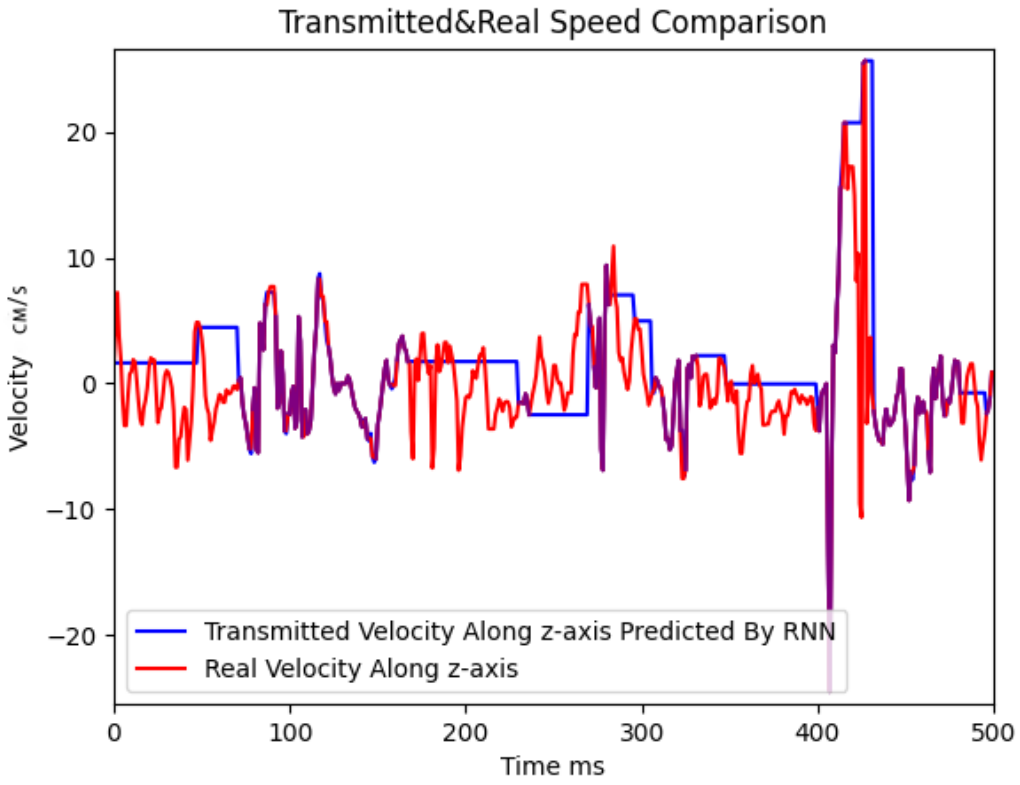
Fig. 3.5 compares the actual and the predicted velocity by LSTM networks along the three axes, which is then transmitted. The output data comes from well-trained LSTM networks with multiple hidden layers. Specifically, the loss of LSTM networks is normally less than 10^{-4} for velocity signals, and 10^{-2} for force signals. The figures of velocity data consist of three parts: the velocity which is going to transmit over the network (purple part), and the value is not transmitting (blue part). From the simulation results, only 42.8% of velocity data will transmit over the network in the next 500 ms, which effectively reduces the packeting rate, and HPW-PSNR is over 42 dB. Nevertheless, there is a distortion along x-axis around 240-270 ms that the transmitted value is nearly -10 cm/s, while the real velocity is greater. And the velocity has a sudden jump at 270 ms. This issue happens mainly due to the mismatch between the trained model and the previously collected data (within the timestep). As is mentioned in (2.9), the training process is not only dependent on the collected data, but timestep and parameters in the LSTM network. This issue can be solved by tuning parameters to retrain a model with lower loss or change the number of timestep. However, the velocity is normally consisted of three axes, the distortion on one axis cannot affect the prediction dramatically.



(a) Transmitted & Real Velocity Comparison Along x Axis



(b) Transmitted & Real Velocity Comparison Along y Axis



(c) Transmitted & Real Velocity Comparison Along z Axis

Fig. 3.5 Transmitted & Real Velocity Comparison Along Each Axis When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\ \mathrm{N \cdot s/m^2}$

Simultaneously, we collected the haptic force feedback in the next 500ms, which is shown in Fig. 3.7. As is illustrated before, the most force feedback during the penetration process is produced along z-axis, which means force along the other 2 axes can be ignored. In fact, the force along the x and y direction is less than 10% of the total. It is given that the human perceptual discrimination for force is 10%. We assume $\alpha_f = 0.1$. Fig. 3.6 shows the predicted label of the force transmission. Therefore, only 40.4% of force data will transmit over the network, and the corresponding HPW-PSNR is around 22 dB. The minimum HPW-PSNR for force is 14 dB, which is also tolerable for the whole system.

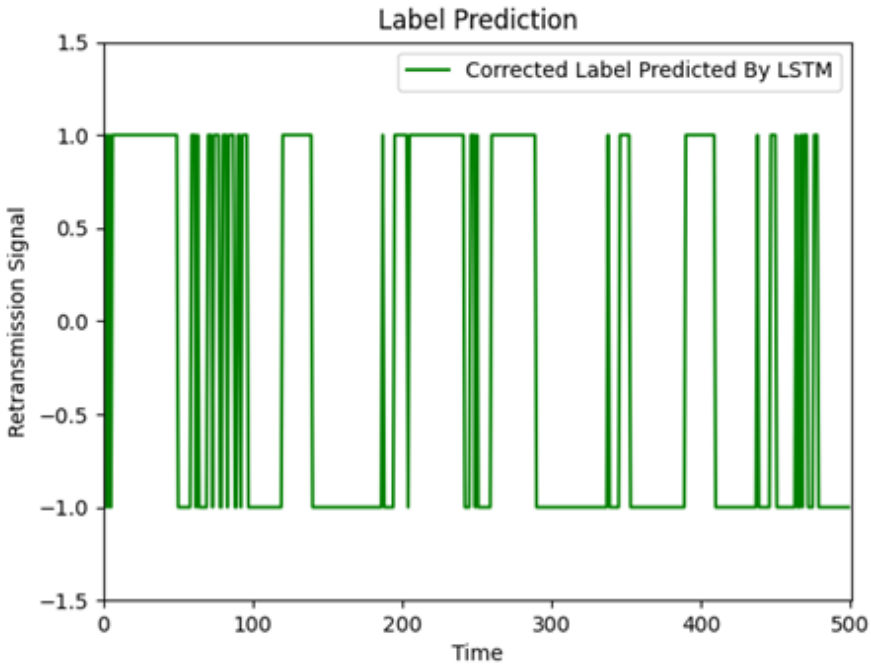


Fig. 3.6 Transmitted Label Prediction for Force

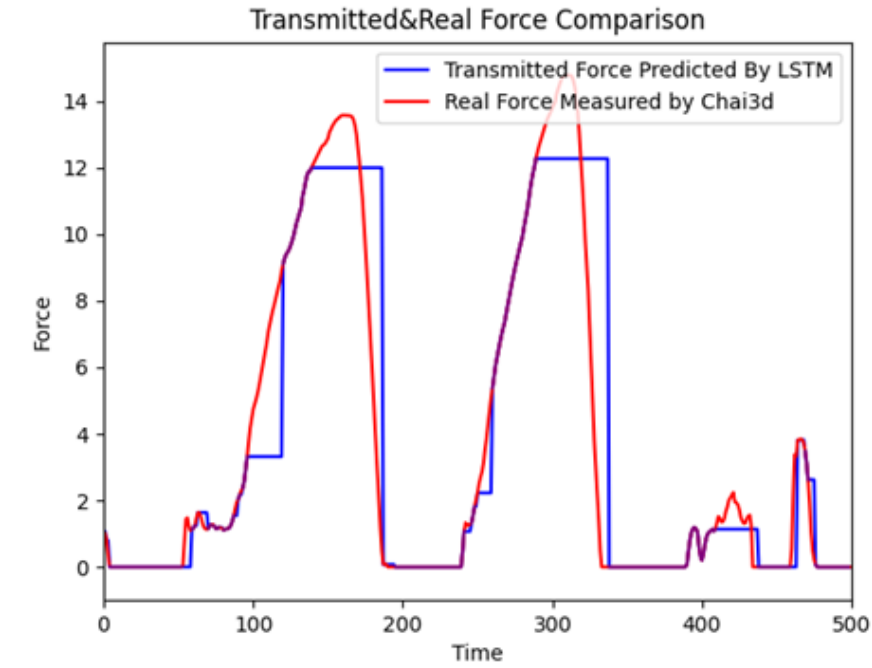


Fig. 3.7 Transmitted & Real Comparison Along Each Axis When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\,\mathrm{N \cdot s/m^2}$

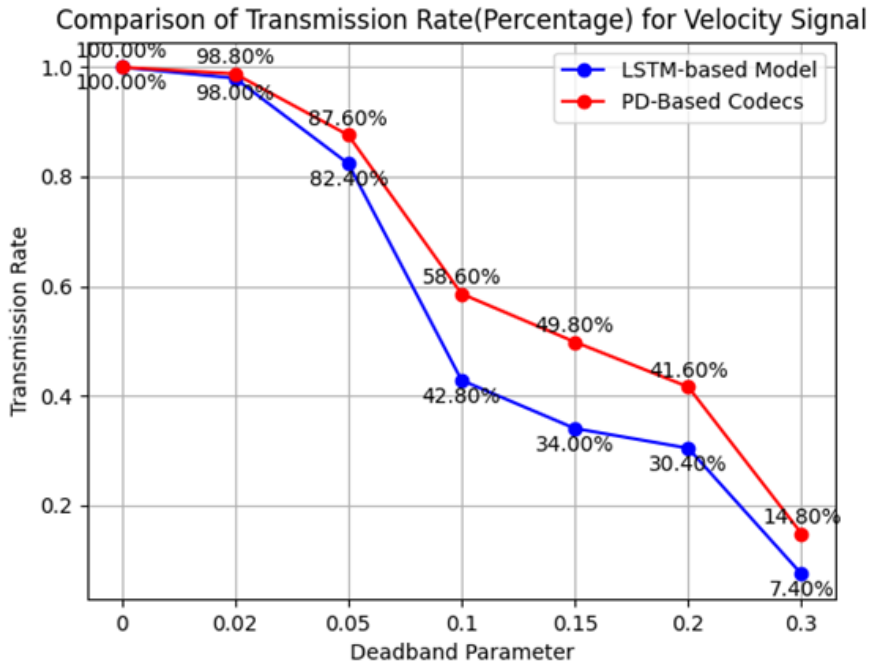
To summarize, the proposed mathematical model is efficient in reducing the transmitting packet rate of both velocity and force data. Over half of the kinesthetic data is not transmitted. The haptic signal degradation of the system is low.

3.5 Comparison with PD-Based Codecs

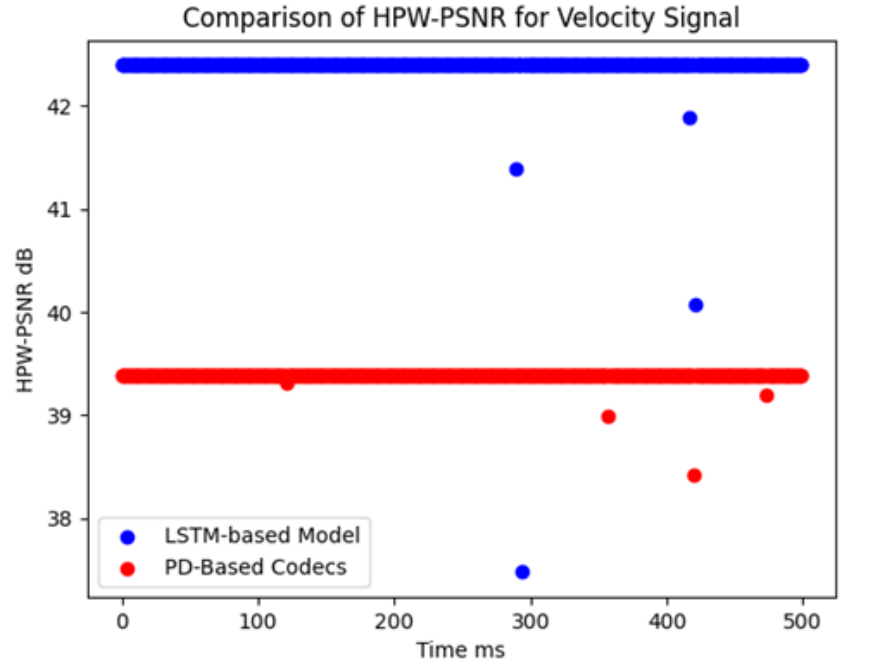
This section compares the performance between the proposed LSTM-based data reduction model and the conventional PD-based codecs. First, we compare the data transmission rate of both velocity and force signals on different deadband parameters. The two models are tested under the same dataset. As is shown in Fig. 3.8a, the rates are quite close when the dead parameter is small (0.02-0.05). However, the difference becomes greater with the increase of the deadband parameter. Therefore, it can be concluded that the proposed model has a lower transmission rate for velocity signals than PD-based codecs.

Similarly, the comparison of transmission rate for force signal is given in Fig. 3.9a. The PD-based codecs have a lower transmission rate when the deadband parameter is small. Conversely, the LSTM-based model will transmit fewer data under greater deadband parameters. Both models have similar results when deadband parameter lies in $[0.1, 0.15]$. Therefore, it can be concluded that the proposed model is suitable for the situation of higher deadband parameters, while the PD-based model fits the environment requiring high transparency.

Fig. 3.8b& 3.9b compare the HPW-PSNR for both velocity and force signals under the same dataset. As is shown in Fig. 3.8b, the transparency of the proposed model is higher when transmitting the velocity signal. As for the force signal, the two models are similar. Both of the models have degradation randomly. The minimum HPW-PSNR for the LSTM-based model is 14 dB, and 21 dB for the PD-based codecs. However, the overall of HPW-PSNRs for both models are greater than 22 dB. Therefore, both mathematical models have similar performance for maintaining transparency, while PD-based codecs have lower degradation.

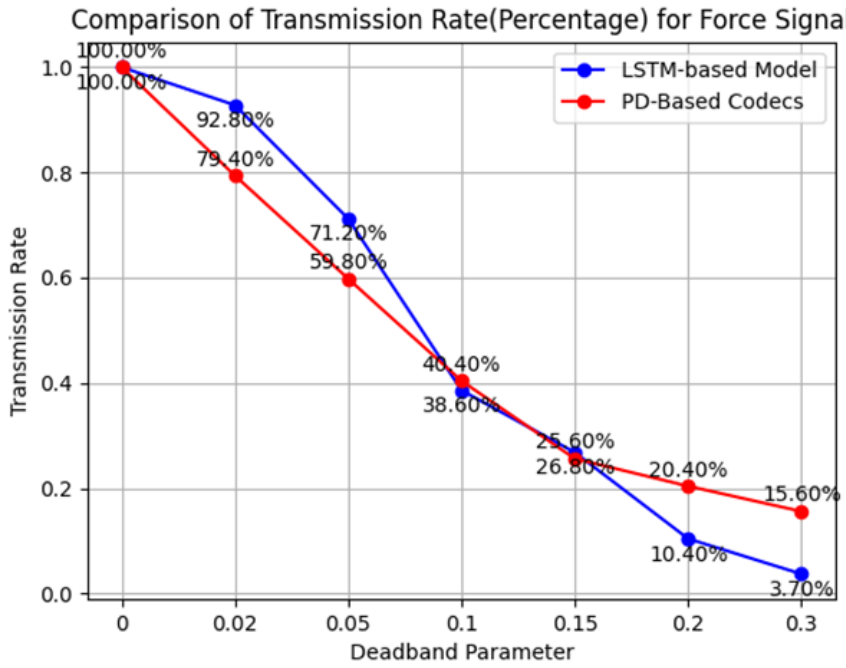


(a) Comparison of Transmission Rate (Percentage) for Velocity Signal When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\ \mathrm{N \cdot s/m^2}$

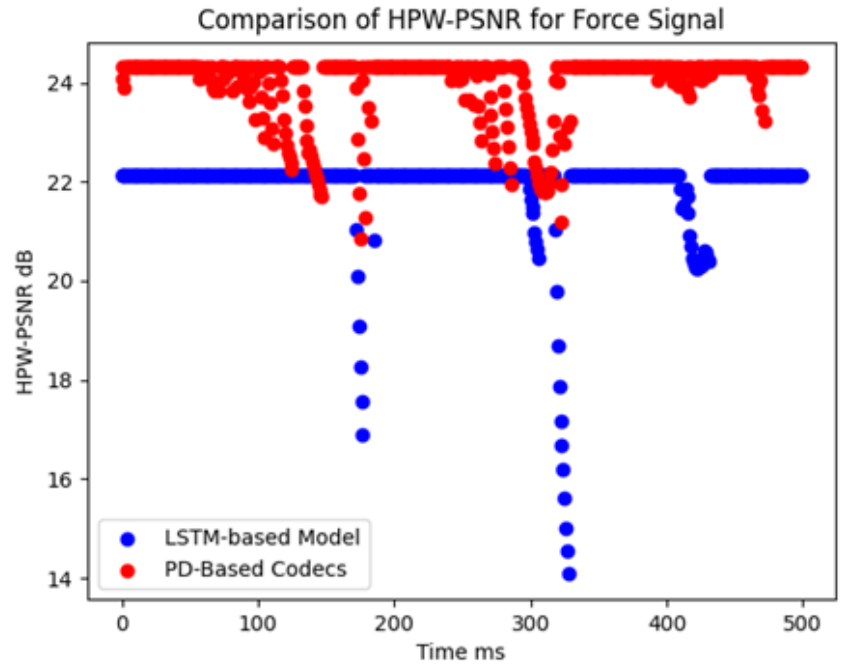


(b) Comparison of HPW-PSNR for Velocity Signal When Tapping on the Foam Surface When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\ \mathrm{N \cdot s/m^2}$, $\alpha_v = 0.1$, $k = 1$, $C = 1$, $N = 500$

Fig. 3.8 Rate-PSNR Figure for Velocity Signals



(a) Comparison of Transmission Rate (Percentage) for Velocity Signals When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\ \mathrm{N \cdot s/m^2}$



(b) Comparison of HPW-PSNR for Force Signals When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\ \mathrm{N \cdot s/m^2}$, $\alpha_v = 0.1$, $k = 1$, $C = 1$, $N = 500$

Fig. 3.9 Rate-PSNR Figure for Force Signals

3.6 Performance Analysis Under Different Motions

We simply classify the user's motions into two categories: tapping and dragging. The movement of dragging is much more complicated than tapping since it moves arbitrarily on the surface. The tapping process normally means the motion that interacts with one or more points on the surface. However, the dragging process mainly causes lateral movements of the object on the surface. Thus, if we move the cursor in the same plane during the experiment, the dragging motion can be represented by a figure which contains speed along x and y axes. Additionally, the scale of friction is normally smaller than the pressure, leading to a different transmission rate. In order to see the movement of the cursor in the 3D space more intuitively, velocity is replaced by speed data. Speed is a scalar representing the magnitude of velocity. By using speed, original three velocity figures along x, y, z axes can be simplified by one figure, as is given in Fig. 3.10. For the transmission of velocity data, the corresponding HPW-PSNR is 40 dB, and transmission rate is 44.6%. Simultaneously, for force data, HPW-PSNR is 33 dB and the transmission rate is 76.8%. As is shown in Fig. 3.11, the friction is normally under 1N. Therefore, to keep the system's transparency, it is illustrated that more transmission of force is required when dragging (over 75%).

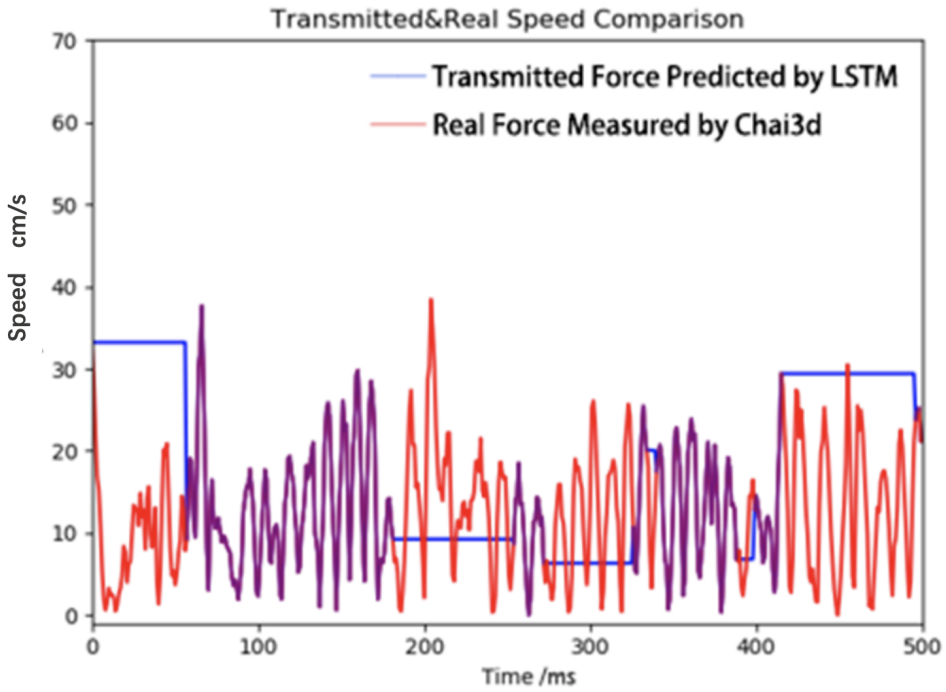


Fig. 3.10 Transmitted & Real Speed Comparison When Dragging on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\,\mathrm{N \cdot s/m^2}$

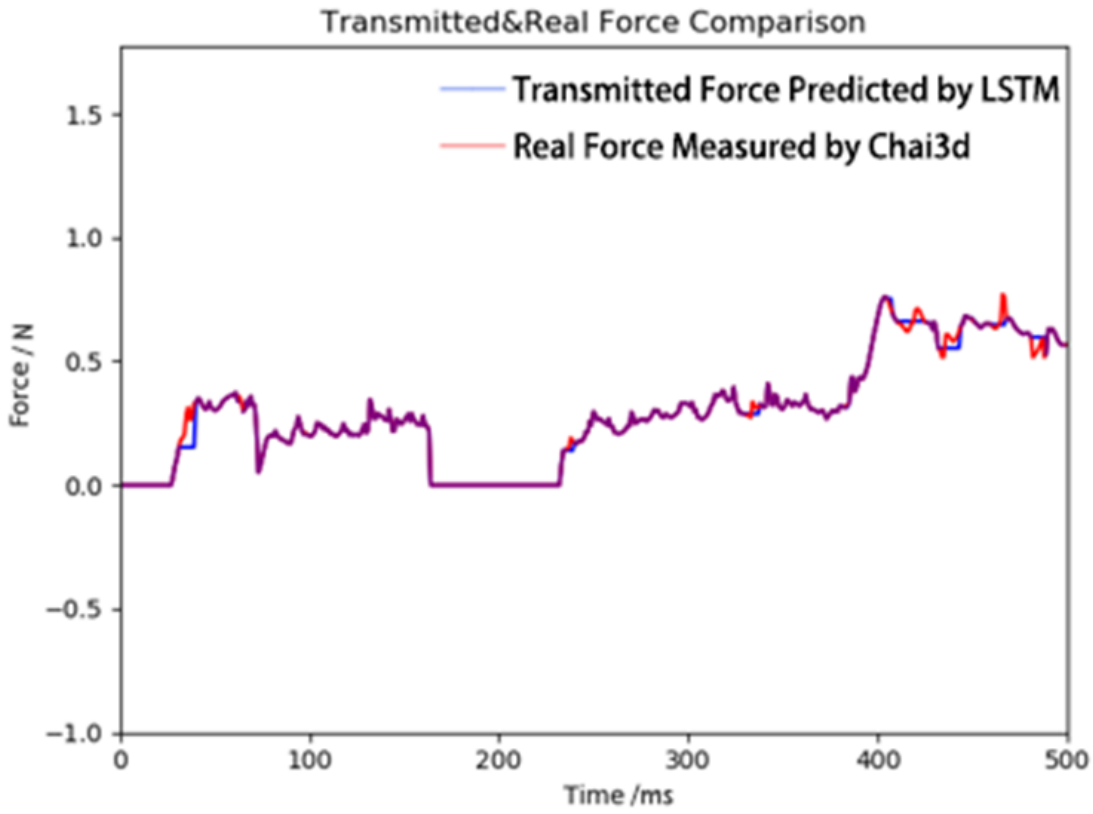


Fig. 3.11 Transmitted & Real Speed Comparison When Dragging on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\ \mathrm{N \cdot s/m^2}$

3.7 Conclusions

3.7.1 Discussion

In this chapter, two LSTM-based mathematical models are proposed for reducing both velocity and force data transmission in bilateral teleoperation systems. Results have shown that LSTM-based mathematical models can reduce the transmission rate effectively. When $\alpha_v = 0.1$ and $\alpha_f = 0.1$, we can find that LSTM-based mathematical models allow only 42.8% of velocity data and 38.6% of force data to be transmitted, which is smaller than PD-based codecs. However, for velocity data, the transmission rate of the proposed LSTM-based model is always smaller PD-based codecs, regardless of α_v.

Apart from less data transmission, LSTM-based mathematical models can still maintain the transparency of the system comparing to traditional PD-based codecs. For instance, when $\alpha_v = 0.1$, HPW-PSNR of the LSTM-based mathematical model is 42.5 dB, which is higher than traditional PD-based codecs. In this case, the proposed model can not only

reduce the velocity data transmission, but increase the transparency of the whole system. For force data, when $\alpha_f = 0.1$, HPW-PSNR of the LSTM-based mathematical model is 22 dB, which is slightly smaller than PD-based codecs. The transparency of the system with proposed LSTM-based model does not degrade much.

Comparing to results of dragging, the transmission rate of in velocity and force data is smaller. Since the friction (normally less than 2 N) is always smaller than pressure, it makes sense that more kinesthetic data needs to be transmitted to maintain a high transparency.

Only results from one type of environment (foam) are given in this chapter. The reason for this is that results are similar from other textures. However, friction on the foam is smaller than the other three textures, which can be used to compare the performance of the proposed model under different motions. When the friction is relatively equal to pressure, i.e., dragging on the marble surface, transmission rates of kinesthetic data can be reduced effectively (41.1% of velocity data and 37.4% of force data) without degrading the transparency.

To summarize, LSTM-based mathematical models can replace current PD-based codecs at a lower transmission rate without degrading the transparency of the system.

3.7.2 Summary

Simulation results and analysis of the proposed LSTM-based kinesthetic data reduction techniques are given in this chapter. We use the Phantom Omni haptic device to collect data on the virtually built environment for collecting kinesthetic data. The kinesthetic data format and the experimental process is also given in this chapter. Subsequently, we discuss the methodology to build LSTM networks on the master and slave side. The training algorithm for LSTM networks is also given.

As for the result analysis, it is illustrated that LSTM-based mathematical models perform well in both data reduction and perceptual transparency. As for velocity data transmission, only 42.8% of data is transmitted over the network with HPW-PSNR of 42.5 dB ($\alpha_v = 0.1$). Comparing with the PD-based codecs, the proposed model transmits velocity signals with a lower rate than PD-based codecs as well as with less degradation. As for force data transmission, only 38.6% of data is transmitted over the network with HPW-PSNR of 22 dB ($\alpha_f = 0.1$).Comparing with the PD-based codecs, the proposed model is suitable for the situation of higher deadband parameters. And both mathematical models have similar performance for maintaining transparency, while PD-based codecs have lower degradation. Additionally, for transmitting the force feedback of dragging, a higher packet rate is required.

To summarize, the proposed LSTM-based bilateral teleoperation system shows good performance when dealing with a time series of data. It has a higher performance for velocity transmission than the conventional PD-based codecs. For force transmission, the two systems perform similarly. But the proposed model determines the transmission status of the newly collected kinesthetic data on both master and slave sides without a pairwise comparison. The proposed LSTM-based perceptual threshold is more practical and maintains the transparency. Therefore, we believe it will improve the practicality of bilateral teleoperation systems in the future.

Chapter 4

Simulation and Performance Analysis of Dimensionality Reduction Techniques

4.1 Introduction

The previous chapter simulated and analysed the transmitted kinesthetic data from the proposed LSTM-based data reduction techniques. Two metrics have been given to compare the proposed model with PD-based codecs and LSTM-based models proposed in section 2.3, which are HPW-PSNR and data transmission rate. In all cases, however, two schemes can be used to reduce the kinesthetic data transmission, one is reducing the packet rate,and the other is lowering the size of each packet. In this section, the simulation results of three DRTs are given, analysed and compared with each other.

There are three metrics for analysing the results, while only two are introduced in Section 4.2, including quatitative metrics of DRTs and bits transmission. Performance metrics of DRTs, on the other hand, assess the data in low dimensions in terms of the preservation of global and local structure. Five metrics for measuring different DRTs are stated, which are Continuity (CON), Trustworthiness (TRU), Relative rank error (RRE), KL divergences, and Accuracy (ACC). Additionally, bits transmission estimates the bits to be transmitted over the network.

Simulation results of 3 DRTs are presented and explained in Section 4.3. The data visualisation of the force data of 2 dimensions ia also presented. It is clear that the UMAP has a good advantage of preserving both the local and global structure of the original data.

As for Section 4.4, the performance of 3 data reconstruction techniques are given. The transparency based on HPW-PSNR shows that both LSTM networks and PCA reconstruction techniques perform similar , and better than SAE. Therefore, a further comparison between is given in 4.5, by comparing the transmission bits of these 2 proposed techniques

with traditional PD-based codecs. It is thus clear that LSTM network needs less average bits transmission.

4.2 Performance Metrics

The kinesthtic data collection has been introduced in Section 3.2. Three different types of performance metrics are introduced in this section, which are quantitative comparison of DRTs, transmission bits and the transparency of the reconstructed signal. While the latter, transparency test, has been introduced in Section 3.3.1.

4.2.1 Performance Quantitative Metrics of DRTs

Five performance metrics are used for quantitative comparison [67]. A k-NN matrix and a rank matrix are used to calculate the metrics in this simulation. The k-NN matrix contains the sample indices (from 0 to $n-1$) of k-th nearest neighbour of the current sample $A_{n\times k}$. The rank matrix contains $R_{n\times n}$ the rank of each sample to sample, whereas entry R_{ij} gives the rank that sample j has to i (the how many 'closest' neighbours among j and i).

CON measures a point in terms of the preservation of k-NNs from the embedding to the original space (layer) [68]. The greater it is, the better preservation it holds. It can be calculated as:

$$CON = 1 - T_{CON} \sum_{i=1}^{M} \sum_{j \in N_{i,k}^{(l)}, j \notin N_{i,k}^{(l')}} (r_{i,j}^{(l')} - k) \tag{4.1}$$

$$T_{CON} = \frac{2}{Mk(2M-3k-1)} \tag{4.2}$$

where T_{CON} is the normalization term, $r_{i,j}^{(l')}$ is the rank of $x_j^{(l')}$ in the k-NN of $x_i^{(l')}$, M is the size of data set, $N_{i,k}^{(l)}$ is the set of indices to the k-NNs of $x_i^{(l')}$. k is set as 50 in this case.

Similarly, TRU is going from the input to the embedding space. It measures a point in terms of the preservation of k-NNs from the original to embedding space [69]. It can be calculated as:

$$TRU = 1 - T_{TRU} \sum_{i=1}^{M} \sum_{j \in N_{i,k}^{(l')}, j \notin N_{i,k}^{(l)}} (r_{i,j}^{(l)} - k) \tag{4.3}$$

$$T_{TRU} = \frac{2}{Mk(2M-3k-1)} \tag{4.4}$$

where T_{TRU} is the normalization term, $r_{i,j}^{(l)}$ is the rank of $x_j^{(l)}$ in the k-NNs of $x_i^{(l)}$. The greater, the better.

As for the difference between the embedding and input data, RRE measures the average of changes in neighbour ranks between two spaces [68]. It calculates the changes in neighbour rank from embedding to input space first, then the changes from input to embedding space. The average value of them form the average change, which can be expressed as follows:

$$RRE = \frac{MR_k^{(l,l')} + MR_k^{(l',l)}}{2} \tag{4.5}$$

$$MR_k^{(l,l')} = T_{RRE} \sum_{i=1}^{M} \sum_{j \in N_{i,k}^{(l)}} \frac{|r_{i,j}^{(l)} - r_{i,j}^{(l')}|}{r_{i,j}^{(l)}} \tag{4.6}$$

$$MR_k^{(l',l)} = T_{RRE} \sum_{i=1}^{M} \sum_{j \in N_{i,k}^{(l')}} \frac{|r_{i,j}^{(l')} - r_{i,j}^{(l)}|}{r_{i,j}^{(l')}} \tag{4.7}$$

$$T_{RRE} = \frac{1}{M \sum_{k'=1}^{k} \frac{|M-2k'|}{k'}} \tag{4.8}$$

where T_{RRE} is the normalization term. As is given from the definition of RRE, the smaller change is, the better neighbour connectivity holds.

Those aforementioned three metrics are used to measure the local neighbourhood connectivity between the input and embedding space. For measuring the global connectivity, a sum of KL divergences is used [70]. It maps the Euclidean distance between each points to probability distance in the original space at first. Then it measures the difference of overall probability distance between the original and embedding space. In this case, the greater, the global connectivity it has.

It can be represented as:

$$D = \sum_{i=1}^{M} \sum_{j=1}^{M} \left| e^{-\frac{\|d_{i,j}^{(l)}\|^2}{2\sigma^2}} \right| \tag{4.9}$$

$$KL = \frac{1}{D} \sum_{i=1}^{M} \sum_{j=1}^{M} e^{-\frac{\|d_{i,j}^{(l)}\|^2}{2\sigma^2}} \left(\frac{\|d_{i,j}^{(l')}\|^2}{2\sigma^2} - \frac{\|d_{i,j}^{(l)}\|^2}{2\sigma^2} \right) \tag{4.10}$$

where D is the sum of the distances mapping from Euclidean space to Gaussian space, which transfers distance to probability. $d_{i,j}^{(l)}$ represents the Euclidean distance between

$x_i^{(l)}$ and $x_i^{(j)}$, $d_{i,j}^{(l')}$ is the Euclidean distance between $x_i^{(l)}$ and $x_i^{(l')}$, and $\sigma^2 = 0.05$ in this experiment. The greater, the better.

ACC is a support vector machines (SVM) linear classifier working in the latent space, space, calculated as follows: (i) using 5-fold cross-validation to evaluate the performance of the embedding results under the linear kernel SVM classifier; (ii) calculating the overall accuracy as the average of the five test accuracy numbers. It estimates the average accuracy between the data in input space and embedding space. The larger, the better.

4.2.2 Bits Transmission

Previous researches have illustrated that 8 important bits are enough to represent each float kinesthtic data by performing an exclusive-or operation between the predicted and previously predicted value [71, 72]. Under this assumption, we assume that each haptic data is encoded to 8 bits. Therefore, we can approximately estimate the bits transmitted over network from the following formula:

$$N_b = 8 * N_t \tag{4.11}$$

where N_t represents the amount of data to be transmitted, and N_b is the bits transmission. For different DRTs, the bits transmitted can be different. For instance, the transmission bits of data processed by PCA include the real transmission data and eigenvectors, as is given in (2.32).

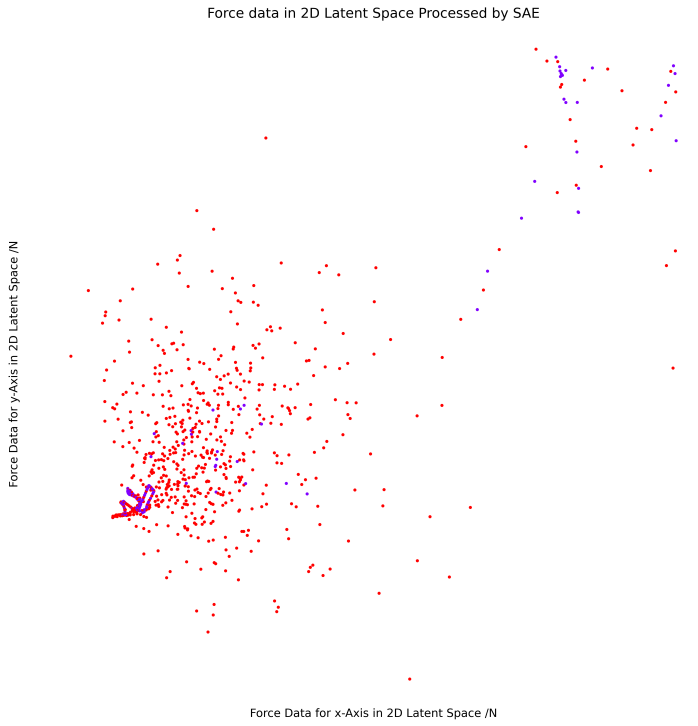
4.3 Results on Performances of Three Dimensionality Reduction Techniques

As is explained in Section 2.4.2, the original position/velocity data is in a 6 dimensional space ($\mathbf{R}^6$), and dimensionality reduction module assumes the intrinsic dimensionality of position/velocity data is 3. Therefore, the dimension of the low space is ($\mathbf{R}^3$) for position/velocity data. However, for 3D force data, the space of embedding is 2, which makes it easier to present the embeddings in this thesis.

Fig. 4.1 compared the distribution of embeddings used 3 different DRTs, i.e. PCA, SAE and UMAP. We use force data collected on marble and do the simulation. Data points are classified into 2 categories, blue and red. Blue points represent the data to be transmitted over network, and red points are not transmitted. The classification is based on JND, which is also given in (2.2). As is shown in Fig. 4.1a& 4.1b, both PCA and SAE cannot classify the labelled kinesthetic data well and have much overlapped data. But UMAP retains the labels of data effectively, while preserving many details in the

learned manifolds. This result represents that UMAP has a higher local neighbourhood connectivity due to the consideration of k-NNs. k is the hyper-parameter defined in (2.39) and (2.40). For each embedding, the distribution of its 50 neighbours are preserved.

(a) PCA Embedding



(b) SAE Embedding

(c) UMAP Embedding, $NN = 50$

Fig. 4.1 Distribution of Embeddings of Force Data Collected on marble Generated by PCA, SAE and UMAP

The quantitative analyses on performances of 3 DRTs for different kinesthetic data are given in Table 4.1 and 4.2. The comparison of 3 DRTs on position/velocity data is in Table 4.1. The original space is $\mathbf{R}^6$, and the embedding space is $\mathbf{R}^3$. All of three dimensionality reduction techniques preserve the local connectivity from the latent to input really well. PCA performs better in CON and TRU, while UMAP has a smaller value in RRE. KL represents the preservation of the global structure mapping from the embedding to input space. UMAP takes the advantage of its maintenance in global structure and has a smaller value in KL. Furthermore, ACC of UMAP has a smaller average value in all of four environments, which represents that it delivers the best transformation for downstream tasks.

Table 4.1 Quantitative comparison on three DRTs for position/velocity data

Material	Metric	PCA	SAE	UMAP
Marble	CON	**0.988**	0.982	0.961
	TRU	**0.987**	0.951	0.961
	RRE	0.062	0.036	**0.021**
	KL	0.049	0.067	**0.003**
	ACC	0.525	0.528	**0.764**
Wood	CON	0.965	0.920	**0.975**
	TRU	0.963	0.915	**0.982**
	RRE	0.065	0.074	**0.014**
	KL	0.038	0.027	**0.003**
	ACC	0.506	0.648	**0.654**
Ceramic Tiles	CON	0.980	**0.987**	0.975
	TRU	0.978	**0.987**	**0.987**
	RRE	0.051	0.015	**0.013**
	KL	0.025	0.025	**0.001**
	ACC	0.581	0.616	**0.689**
Foam	CON	**0.985**	0.984	0.973
	TRU	0.984	**0.987**	0.984
	RRE	0.063	0.032	**0.015**
	KL	0.022	**0.001**	0.002
	ACC	0.575	**0.707**	0.691

Table 4.2 Quantitative comparison on three DRTs for force data

Material	Metric	PCA	SAE	UMAP
Marble	CON	**0.989**	0.988	0.940
	TRU	0.950	**0.953**	0.945
	RRE	0.036	**0.033**	0.040
	KL	0.010	0.013	**0.002**
	ACC	0.694	0.694	**0.828**
Wood	CON	0.957	**0.989**	0.952
	TRU	0.834	0.960	**0.971**
	RRE	0.102	0.030	**0.027**
	KL	0.007	0.020	**0.002**
	ACC	0.842	0.842	**0.932**
Ceramic Tiles	CON	**0.981**	**0.981**	0.947
	TRU	0.924	0.924	**0.973**
	RRE	0.051	0.053	**0.024**
	KL	0.009	0.012	**0.002**
	ACC	0.804	0.922	**0.984**
Foam	CON	0.964	**0.982**	0.952
	TRU	0.812	0.931	**0.956**
	RRE	0.110	0.048	**0.028**
	KL	0.013	0.035	**0.002**
	ACC	0.754	0.754	**0.821**

To sum up, UMAP delivers the best transformation for both position/velocity and force data in terms of the local and global connectivity. Furthermore, UMAP is allowed to preserve more details in the learned manifolds with retaining the labels of data simultaneously.

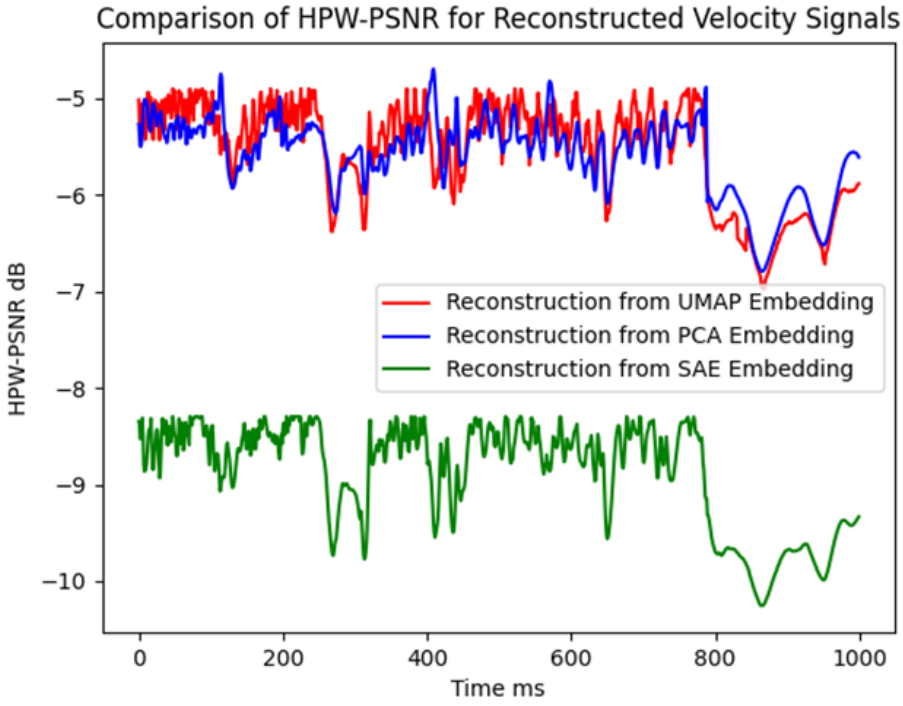
4.4 Results on Performances of Three Data Reconstruction Techniques

HPW-PSNR is a metric of measuring the perceptual significance of the signal degradation from weighting the difference between the original and distorted signal, which has been given in (3.2- 3.4). It is introduced in measuring the performance of data reconstruction modules by comparing the original and reconstructed data.

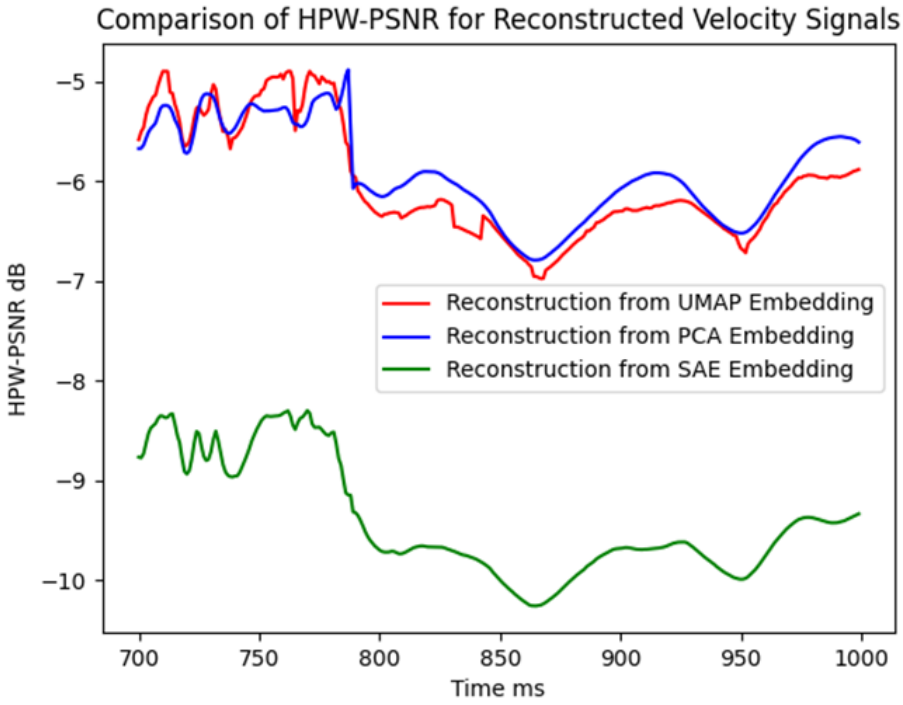
Fig. 4.2 compares the HPW-PSNR of three different reconstructed velocity signals with the original data. The original data is collected when the cursor contacting with marble. There are totally 1000 data points are used to do the simulation and test apart from the training data. As is mentioned before, the sampling rate of the device is 1 ms, which means all of the test data is collected within 1000ms. As is shown in Fig. 4.2a, the overall HPW-PSNRs of PCA and UMAP are better than that in SAE. The average value of HPW-PSNR of PCA is -5.58 dB, and average HPW-PSNR value of UMAP is -5.53 dB. Since HPW-PSNR is a metric of transparency between the reconstructed and original signal, both PCA and UMAP preserve the transparency of velocity signal very well, while UMAP performs slightly better than PCA. On top of that, it is interesting to find that the three data reconstruction techniques' HPW-PSNRs have the same trend of increment and decrement. In other words, the derivations of three HPW-PSNRs are quite close. As is shown in Fig. 4.2b, in the time slot 700-1000 ms, all of the values drop from 780 ms to 850 ms and 925 ms to 950 ms. They both increase from 950 to 1000 ms. It can be concluded that all of three data reconstruction techniques have the same tendency of upgrading or degrading the transparency, while LSTM networks for reconstructing UMAP data performs better than inverse PCA and SAE.

Fig. 4.3 compares the HPW-PSNRs of three different reconstructed force signals with the original data. The original force feedback is collected when the cursor moves on the marble. It is obvious to see that the reconstruction from UMAP embedding has the best performance with the average HPW-PSNR of 28.58 dB. The average HPW-PSNR of PCA is approximately equal to that in LSTM network for UMAP data reconstruction, and the average value is 28.28 dB. However, both LSTM network and PCA reconstruction techniques perform better than SAE.

Test results of which data collected on wood, ceramic tiles and foam have the similar results. Therefore, both inverse PCA and LSTM-based reconstruction techniques can be used to reconstruct the embeddings.



(a) Time=0-1000 ms



(b) Time=700-1000 ms

Fig. 4.2 Comparison of HPW-PSNR for Velocity Signals When Contacting with the Marble Surface, $\mu = 0.9$, $\delta = 0.4\,\mathrm{N \cdot s/m^2}$, $\alpha_f = 0.1$, $k = 1$, $C = 1$, $N = 1000$

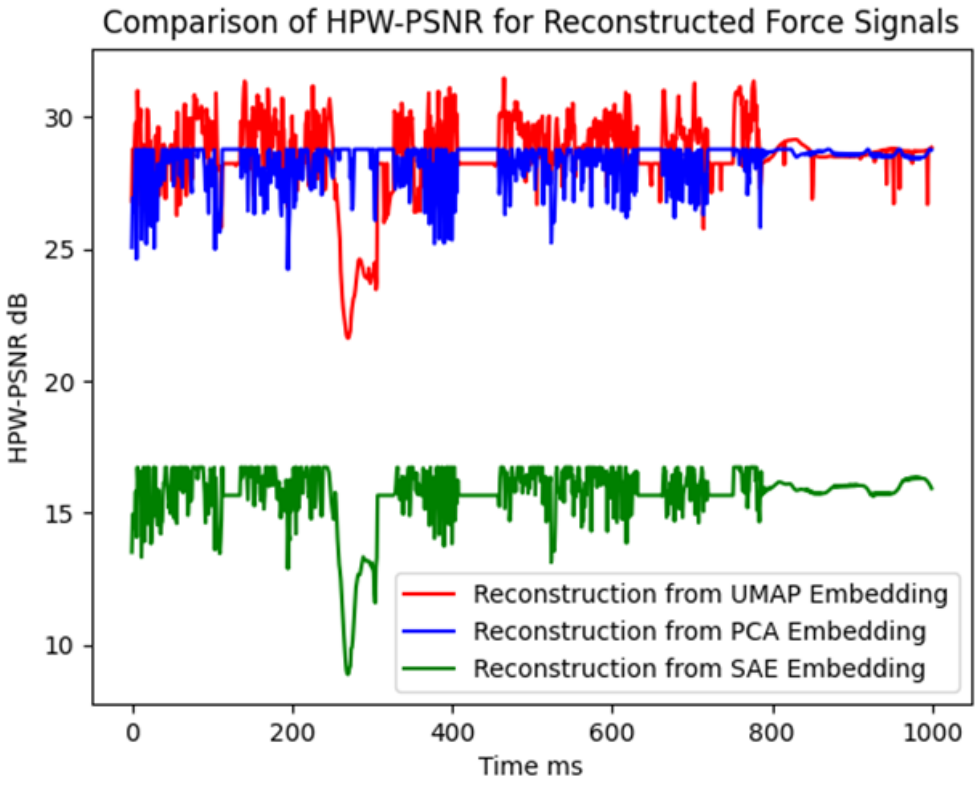


Fig. 4.3 Comparison of HPW-PSNR for Velocity Signals When Contacting with the Marble Surface, $\mu = 0.9$, $\delta = 0.4\,\mathrm{N}\cdot\mathrm{s/m^2}$, $\alpha_f = 0.1$, $k = 1$, $C = 1$, $N = 1000$, Time=0-1000 ms

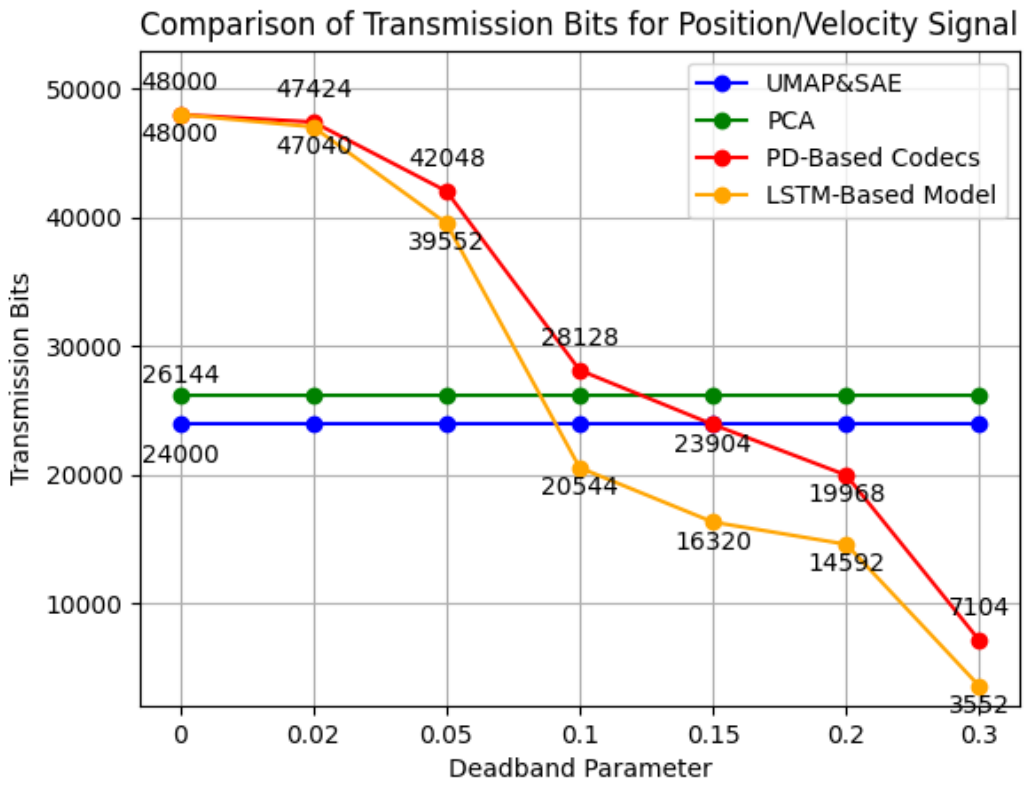
4.5 Comparison of Bits Transmission over the Network with PD-Based Codecs and LSTM-based Models

This section compares the bits transmitted over the network when applying different dimensionality reduction techniques on kinesthetic data with PD-based codecs and LSTM-based models. PD-based codecs illustrate that the kinesthetic data should be transmitted over the network when the newly collected data is outside of the deadzone of the previously transmitted data. Therefore, the perceptual threshold is a parameter to control the bits transmitted over the network.

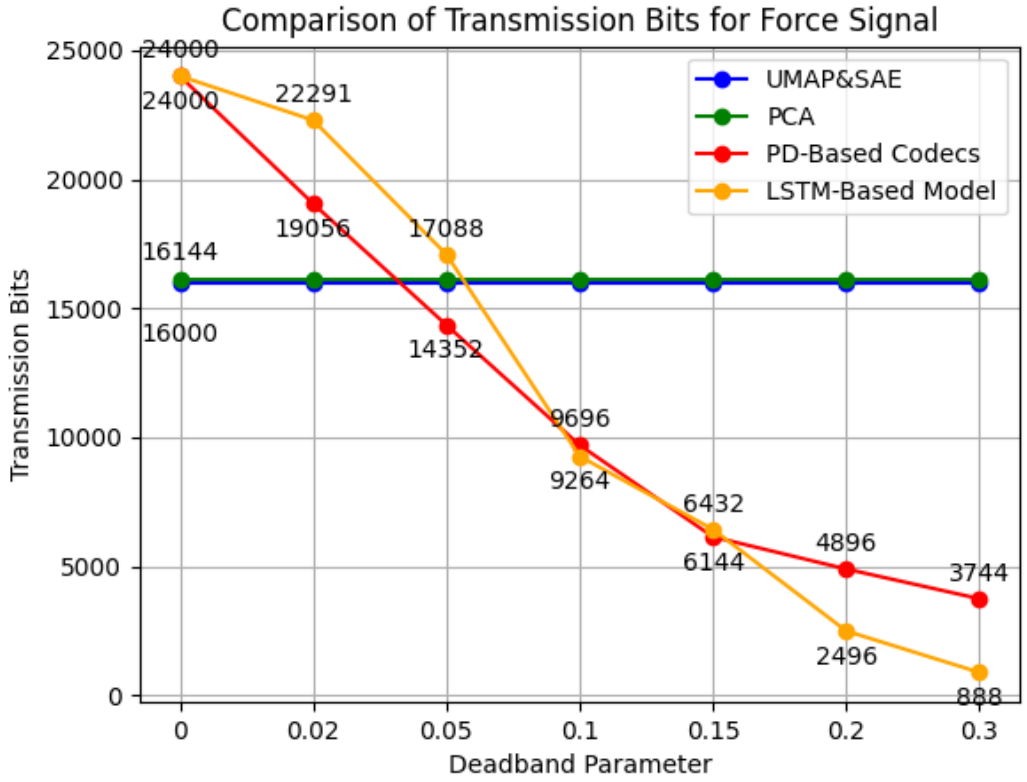
We assume that each kinesthetic data sampled in 1 ms (sampling rate is 1kHz) on each dimension is encoded to 8 bits (1 byte), and 1000 samples are transmitted at the same time. Comparisons of transmission bits for both position/velocity and force signals are shown in Fig. 4.4. As is given, for position/velocity data, UMAP&SAE techniques transmit the averagely smallest data comparing with PCA and PD-based codecs. The reason for PCA

transmits larger data is that the eigenvector (2.32) should be transmitted over the network for reconstructing data. Thus, it has more data to be transmitted. Comparing to PD-based codecs and LSTM-based models, UMAP&SAE techniques only transmit more data when deadband parameter α_v is greater than 0.1. Normally, α_v is 0.1, and all of four techniques have the similar performance, while LSTM-based models transmit least data over the network. However, the greater deadband parameter, the more degradation of transparency will be introduced into the system. The ideal deadband parameter is less than or equal to 0.1, and in that case, three dimensionality reduction techniques transmit much smaller data than PD-based codecs and LSTM-based models.

For force data, all of four techniques perform similarly when α_f is 0.05. With the increase of α_f, PD-based codecs and LSTM-based models transmit less data. 0.05 is a threshold value, when α_f is smaller it, three DRTs are favoured. Otherwise, LSTM-based models transmit less data over the network.



(a) Comparison of Bits Transmitted over the Network with Traditional PD-Based Codecs and LSTM-based Model for Position/Velocity Data



(b) Comparison of Bits Transmitted over the Network with Traditional PD-Based Codecs and LSTM-based Models for Force Data

Fig. 4.4 Comparison of Bits Transmitted over the Network with PD-Based Codecs and LSTM-based Models

4.6 Conclusions

4.6.1 Discussion

In this chapter, results of the performance of three DRTs under four types of environment are given in the beginning. Five metrics are introduced as the basis for verification. As is given in Table 4.1 and Table 4.2, data with best test results in the same metric for different DRTs is highlighted in bold text. It is clear that most of values in UMAP are highlighted for both position/velocity and force signals, which indicates that UMAP can not only preserve the global structure of original data, but also has higher accuracy. Simultaneously, PCA and SAE have similar results, which is consistent with the observation of three DRTs shown in Fig. 4.1. Additionally, UMAP performs better under smooth surfaces (lower μ) since collected position/velocity and force data fluctuations are not obvious. When tapping on marble, force data varies in a larger range, which complicates the global structure of data.

Results of the performance of three data reconstruction techniques are given in section 4.4. As for transparency test, LSTM-based models for reconstructing UMAP embeddings and inverse PCA preserve higher transparency than SAE for both position/velocity and force signals. Only data collected on marble surfaces is presented in this chapter, since UMAP performs worst on marble compared to the other three textures. However, Fig. 4.2 has shown that UMAP embeddings can be well reconstructed by LSTM network comparing to the other two data reconstruction techniques. In fact, reconstructing UMAP embeddings on other textures maintains higher transparency (average 3 dB in HPW-PSNR).

Comparison of transmission bits over the network with PD-based codecs and LSTM-based models is given in section 4.5. Generally, those techniques with fewer transmission bits are preferred. Both PD-based codecs and LSTM-mathematical models are mathematical models which help bilateral teleoperation systems selectively transmit data. On the contrary, DRTs aim at reducing transmission bits of each collected data. It is worth discussing how we choose between these two ideas. As is shown in Fig. 4.4, there exists a threshold that divides the selection of DRTs or mathematical models. As for position/velocity data, when α_v is less than 0.1, three DRTs are recommended. More specifically, UMAP should be selected since it transmits less data than PCA, and maintains higher transparency than SAE. Otherwise, when α_v is greater than or equal to 0.1, LSTM-based models should be selected as it transmits less data. Thus, the selection threshold for position/velocity data is 0.1. Similarly, we can find the selection threshold for force data is 0.05. UMAP is preferred when α_f is less than 0.05. When α_f is equal to 0.05, two types of techniques have similar results, while PD-based codecs transmit least data. When α_f is

greater than 0.05, LSTM-based models are advised to reduce force data transmission over the network.

To summarize, DRTs can reduce kinesthetic data transmission over the network effetively without degrading the transparency of the system. Among the proposed three DRTs, UMAP outperforms the other two techniques in preservation of global and local structure of the original data, and maintains good transparency in its corresponding data reconstruction technique. When DRTs compare to PD-based codecs and LSTM-based models, a selection of two types of techniques is required. In general, when the deadband parameter is relatively high, LSTM-based models are recommended. When it is small, UMAP can be a good choice.

4.6.2 Summary

In this chapter, we discussed the metrics for the performance of dimensionality reduction (CON, TRU, RRE, KL ACC) and data reconstruction techniques (HPW-PSNR).

As for the result analysis, it is found that UMAP delivers the best transformation for both position/velocity and force data from the angle of quantitative comparison. UMAP has an average greater value in TRU, ACC, and an average smaller value KL and ACC. Thus, UMAP performs well in both local and global structure, and has the best accuracy out of three dimensionality reduction techniques. Comparing with the data reconstruction techniques, both PCA and UMAP perform better than SAE, and the average values of HPW-PSNR for PCA and UMAP are close. Additionally, all of three data reconstruction techniques have the same tendency of upgrading or degrading the transparency when dealing with the velocity signal.

Compared to the conventional PD-based codecs and LSTM-based model, all of three proposed dimensionality reduction techniques can transmit smaller data with lower degradation when deadband parameter is small. What's more, UMAP and SAE transmit smaller data bits than PCA since they do not need to transmit information for helping reconstructing data.

To summarize, the proposed bilateral teleoperation system with UMAP dimensionality reduction and data reconstruction modules shows good performance compared with the other two techniques. It has higher performance for preserving both local and global structure of the original data with learned manifolds. As for transparency, the reconstructed values have a higher HPW-PSNR than others. As for bit transmission, it transmits the smallest amount of the data over the network. We believe it has a bright future for reducing the size of kinesthetic data transmission in the future.

Chapter 5

Simulation and Performance Analysis of Unsupervised Clustering Techniques and GBDT-based Force Predictive Module

5.1 Introduction

Simulation results and performance analysis of two unsupervised clustering techniques and a GBDT-based force feedback predictive technique are presented in this chapter. Results of unsupervised clustering techniques for kinesthetic data reduction can be viewed as an extension of proposed LSTM-based models in Chapter 3. More specifically, clustering techniques ensure samples in the same cluster are more similar than those which in other clusters. Furthermore, the clusters are well separated from each other. Also, results of proposed GRDT-based predictive coding are also given to refine kinesthetic data reduction models.

Three metrics for analysing the performance of unsupervised clustering techniques are introduced in Section 5.2, which are Silhouette coefficient (SC), Calinski-Harabasz index (CHI) and Davies-Bouldin index (DBI). They measure similarities of points in one cluster, and separations amongst different clusters. Then the optimal number of cluster K can be selected based on these metrics. In addition, the elbow method is introduced to estimate the range of optimal number of clusters K for 2 clustering techniques.

In addition, we compare the proposed GBDT-based predictive technique with PD-based codecs in three metrics, including transparency based on HPW-PSNR, predictive rate and number of model updates. HPW-PSNR has been introduced in Section 3.3.1. The predictive rate is a subjective metric which justifies whether thepredictive force feedback is the feeling same with real one to human OP. Thus, Weber's law of JND is used for the subjective test. The number of model updates measures the stability of the system. Model

updates will happen when the difference between predictive and real force feedback can be felt by users. The new model will be sent over the network from the slave to master side. Generally, a system requiring fewer model updates is more stable.

Then simulation results of two clustering techniques with different number of clusters are presented in Section 5.3. First, we use the elbow method to find the range of optimal K among the 2 clustering techniques. Then, optimal Ks are determined by looking into how values of three metrics change with K.

In Section 5.4, clustering techniques for kinesthetic data reduction are compared with LSTM-based models and PD-based codec. All of these techniques are tested under the same dataset, which is collected when tapping on the foam surface within 500 ms. LSTM-based models have a higher HPW-PSNR when α_v is 0.1. Even though, clustering techniques have higher HPW-PSNRs for force feedback. As for transmission rate, LSTM-based models are more suitable for the situation with a higher deadband parameter, while unsupervised clusterings can effectively reduce the transmission rate when deadband parameter is low.

Simulation results and performance analysis of GBDT-based force predictive technique are given in Section 5.5. Compared with traditional ZOH and FOLP predictive schemes, the proposed model maintains higher transparency and requires fewer model updates, which shows that the GBDT predictive scheme is bette in both transparency and stability.

5.2 Performance Metrics

Three metrics for measuring the performance of two unsupervised clustering techniques are discussed in Section 5.2.1. They mainly focus on measure the internal relationship (the relationship between one object and its cluster) and external relationship (the relationship between one cluster with other clusters). Elbow method is also introduced in this section. In Section 5.2.2, two more metrics are introduced to test the proposed GBDT-based predictive scheme. A subjective test model based on Weber's law is given. Also, the amount of model updates is another indicator to test the stability of the whole system.

5.2.1 Three Metrics for Analysing Unsupervised Clustering Techniques

From kinesthetic data reduction techniques based on the unsupervised clustering proposed in Section 2.5, the number of K in k-means clustering is normally given artificially. As for hierarchical clustering, pairs of clusters are merged as one moves up until only 1 cluster exists. However, it is so impractical to classify kinesthetic data to 1 cluster that a number of cluster k should be predefined before hierarchical clustering as well. Therefore, metrics for selecting the optimal K are given in this section.

Silhouette coefficient (SC) measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation) [82]. The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering. Scores around 0 indicate overlapping clusters. For all the points $i \in C_i$, the average dissimilarity of i to all other objects of C_i can be defined as:

$$a(i) = \frac{1}{N_{C_i} - 1} \sum_{j \in C_i, i \neq j} d(i,j) \tag{5.1}$$

where N_{C_i} is the number of points belonging to cluster C_i, and $d(i,j)$ is the distance between data points i and j in the cluster C_i. $a(i)$ can be interpreted as a metric of how well i is assigned to its cluster, the smaller the value, the better the assignment.

Let us consider another cluster C_j which is different from C_i, then the average dissimilarity of i to all other objects of C_j can be defined as:

$$d(i,C_j) = \frac{1}{N_{C_j}} \sum_{j \in C_j} d(i,j) \tag{5.2}$$

where N_{C_j} is the number of points belonging to cluster C_j, and $d(i,j)$ is the distance between data points i in the cluster C_i and j in the cluster C_j.

After computing $d(i,C_j)$ for all clusters $C_j \neq C_i$, we select the smallest of those numbers and denote it by:

$$b(i) = \min_{C_j \neq C_i} d(i,C_j) \tag{5.3}$$

In this case, the cluster C_j is termed the neighbour of object i, for which this minimum is attained (that is, $b(i) = d(i,C_j)$). $b(i)$ can be interpreted as a metric of how dissimilar i is assigned to its neighbour, the greater the value, the better the assignment.

Then the Silhouette value of 1 data point i can be defined as:

$$s(i) = \begin{cases} 1 - \dfrac{a(i)}{b(i)} & \text{,if } a(i) < b(i) \\ 0 & \text{,if } a(i) = b(i) \\ \dfrac{a(i)}{b(i)} - 1 & \text{,if } a(i) > b(i) \end{cases} \tag{5.4}$$

We can even write this in 1 formula:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{5.5}$$

The mean value of all $s(i)$ in a cluster measures how tightly grouped all the points in the cluster are. Meanwhile, the mean $s(i)$ over all data points measures how appropriately

that the data has been clustered. Thus, the term SC has been introduced for maximizing the mean $s(i)$ over all data of the dataset [83], which can be represented as:

$$SC = \max_{k} \tilde{s}(k) \tag{5.6}$$

where k is a specific number of clusters, and $\tilde{s}(k)$ represents the mean $s(i)$ over all data of the entire dataset when k clusters are given.

CHI can be used to evaluate the model without knowing the underlying ground-truth labels, where the quantities and characteristics inherent to the dataset are used to verify how well the clustering works [84]. It measures how similar an object is to its own cluster compared to other clusters. The similarity of a point to its own cluster, termed as cohesion, is estimated by distances from data points in a cluster to its cluster centroid, which can be calculated as:

$$Co = \frac{\sum_{k=1}^{K} \sum_{i \in C_k} d(i, c_k)}{N - K} \tag{5.7}$$

where C_k is the k-th cluster, c_k is the centroid of C_k, N is the total number of dataset, and K is the number of cluster. Co can be interpreted as a metric of how similar the object is to its cluster, the smaller the value, the dense it is.

As opposed to coherence, separation is estimated based on the distance of cluster centroids from the global centroid, which can be represented as:

$$Se = \frac{\sum_{k=1}^{K} n_k \cdot d(c, c_k)}{K - 1} \tag{5.8}$$

where c is the global centroid, c_k is the centroid of C_k, and K is the number of cluster. Se can be interpreted as an indicator of the degree of segregation between each cluster, the greater the value, the disconnection it is.

Therefore, CHI is given as:

$$CHI = \frac{Se}{Co} \tag{5.9}$$

Higher value of CHI means the clusters are dense and well separated.

DBI is a method used for measuring the validity of clustering [85]. Measurement with DBI is maximizing inter-cluster distances and minimizing intra-cluster distances amongst points. The maximum of inter-cluster distance means difference between each cluster is more clear. Simultaneously, the minimum of intra-cluster distance indicates that each object in the cluster has a high level of similarity.

Separation of objects in a cluster is denoted as the average distance from data points to the centroid, which can be calculated as:

$$S_i = \frac{1}{N_{C_i}} \sum_{j \in C_i} d(j, c_i) \tag{5.10}$$

where N_{C_i} is the number of points in the cluster C_i, c_i is the centroid respectively, j represents the j-th data point in the cluster, and $d(j, c_i)$ means the distance between j and centroid c_i.

Additionally, the distance between cluster C_i and C_j denotes the separation of 2 clusters, which can be calculated as:

$$M_{i,j} = d(c_i, c_j) \tag{5.11}$$

Assuming $R_{i,j}$ measures te performance of the clustering scheme. This metric, by definition has to account for the separation $M_{i,j}$ between the i-th and j-th cluster, which ideally has to be as large as possible. Moreover, Si denotes the within cluster scatter for cluster i, which has to be as low as possible. Hence, $R_{i,j}$ can be constructed as:

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}} \tag{5.12}$$

where $R_{i,j}$ is a non-negative value.

Then DBI can be given by given the maximum $R_{i,j}$ of the cluster C_i as:

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{i,j} \tag{5.13}$$

where k is the number of cluster.

Last but not least, elbow method is introduced to estimate the range of number of clusters k. The method consists of plotting inertia (the sum of squared distances of samples to their closest centroid) as a function of the number of clusters and choosing the elbow of the curve as the number of clusters to use [86]. The value of inertia in elbow method can be calculated as:

$$I = \sum_{i=0}^{N} \min_{c_j \in C} d(x_i, c_j)^2 \tag{5.14}$$

An elbow plot shows at what value of k, the distance between the centroid of a cluster and the other data points in the cluster is at its lowest. It is a heuristic to help us determine the range of number of clusters.

5.2.2 Predictive Rate

We define the prediction is successful when the human cannot notice the difference between the predictive and actual force feedback, which obeys Weber's law of JND. Otherwise, it is a wrong prediction.

Therefore, the predictive rate can be expressed as follow:

$$R = \frac{N_p}{N} \times 100\% \tag{5.15}$$

$$\begin{aligned} \text{If} \quad & \|\vec{F}_n - \vec{\hat{F}}_n\| \leq \alpha_f \cdot \|\vec{\hat{F}}_n\| \quad N_p \text{ plus } 1 \\ \text{Else} \quad & \qquad\qquad\qquad\qquad\quad N_p \text{ plus } 0 \end{aligned} \tag{5.16}$$

where $\vec{F}_n$ is the n-th predictive force vector, and $\vec{\hat{F}}_n$ is the reference force vector collected by the sensor. α_f is the noticeable threshold for force, which is 10% (8-12% in the Table 2.1). N_p represents the number of predicted samples which are indistinguishable for human from the reference force feedback for the human OP, while N is the total amount of reference force feedback collected by the sensor. The value of N_p increases each time by 1 if the predicted and real data is imperceptible. It is a subjective test which justifies whether the predictive force feedback is the feeling same with real one to human OP.

5.2.3 Number of Model Updates

From the previous description of a bilateral teleoperation system with predictive schemes in Section 2.5.2, model update control modules are required to synchronize the force predictive module once the predicted value has a intolerable variance with the real one.

However, there is no doubt that the transmission of new predictive model via backward channel can increase the time delay. On top of that, frequent model updates can make the system unstable. Therefore, the fewer model updates, the better stability and less transmission though the network.

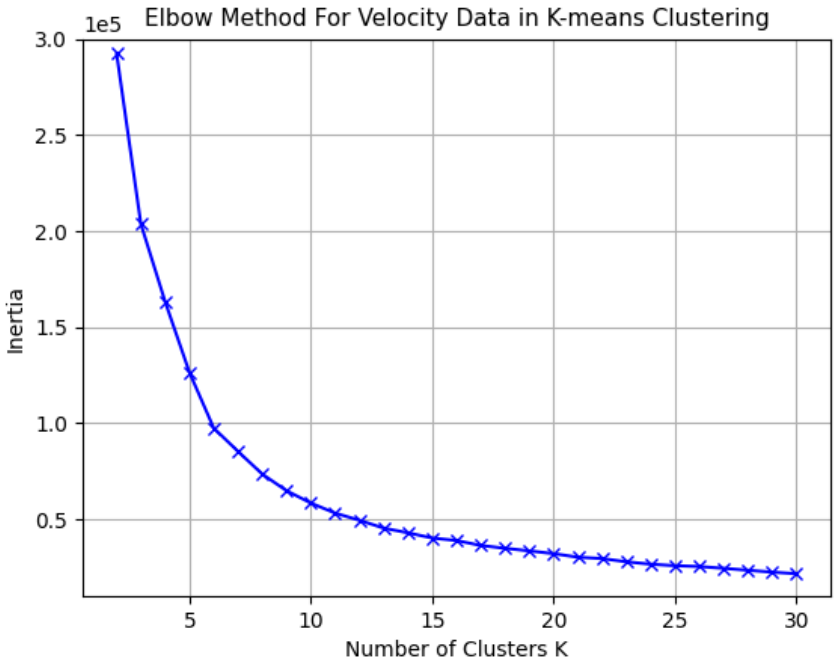
The model update module triggers the updates by obeying (2.66). As long as the difference between the predicted and real force feedback is perceivable to human OP, a newly obtained model needs to be transmitted from the TOP to OP side. There are 2 predictive schemes in comparison, including proposed GBDT-based predictive scheme and PD-based codecs. Both numbers of model updates are counted during the simulation under different noticeable threshold for force α_f.

5.3 Determination of Number of Clusters in Unsupervised Clustering Techniques

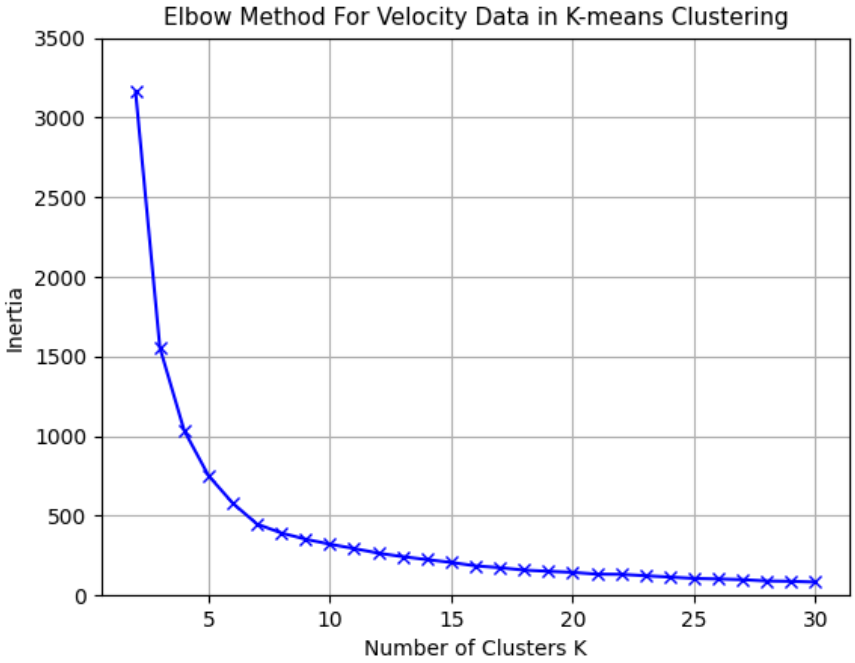
In this section, the kinesthetic data used is the same as in the previous Section 3.4& 3.5, which is collected when tapping on the foam surface within 500 ms. Thus, a dataset contains 500 sets of velocity data and another 500 sets of force data in $\mathbf{R}^3$ is used.

Fig. 5.1a and Fig. 5.1b show how inertia varies with the number of clusters K as it grows. Inertia is the sum of squared distances of samples to their closest cluster centre. In general, inertia decreases with an unstable slope. The elbow refers to a turning point before which the slope decreases dramatically and after which the slope drops gently. For both velocity and force data, inertia drops dramatically when K lies in the range 1 to 5. Also, it drops relatively slowly when K is greater than 15. Thus, in this case, both optimal Ks for velocity and force data in k-means clustering are in the scope of 5 to 15.

Nonetheless, the optimal Ks in hierarchical clustering are even greater according to the figures depicted in Fig. 5.2a& 5.2b. Comparing to Fig. 5.1a and Fig. 5.1b, average inertia of hierarchical clustering is greater than that of k-means clustering. Also, the downtrend of inertia in hierarchical clustering is not as smooth as that in k-means clustering. For velocity data in hierarchical clustering, we can subjectively determine that the optimal K is 16 for which satisfies all the requirements of an elbow. Furthermore, the optimal K for force data should be the point at which has the most significant change in slopes. That is, K is 13. However, the aforementioned two elbow points are given subjectively, thus an unbiased conclusion can be drawn from figures that both optimal Ks for velocity and force data in hierarchy clustering range from 10 to 20.

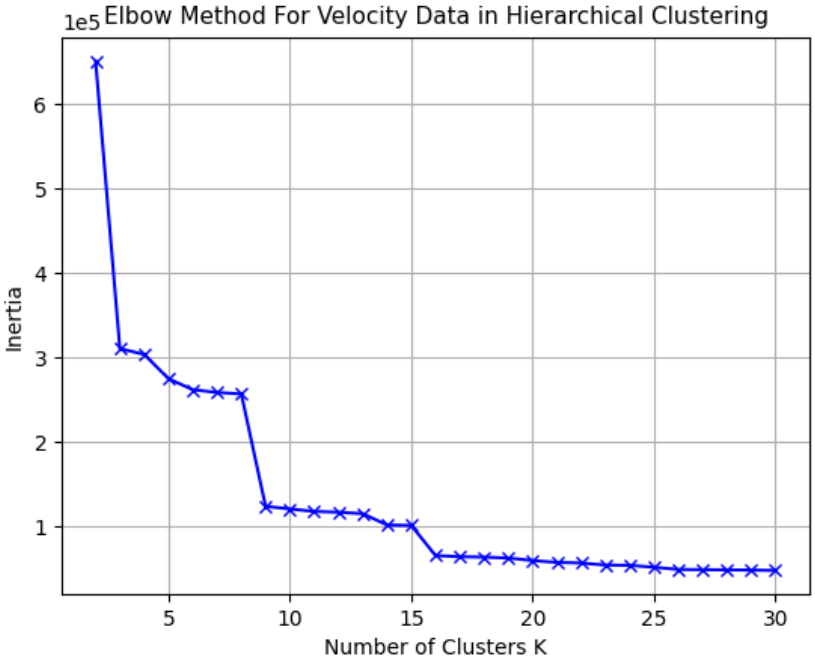


(a) Elbow Method for Velocity Data in K-means Clustering

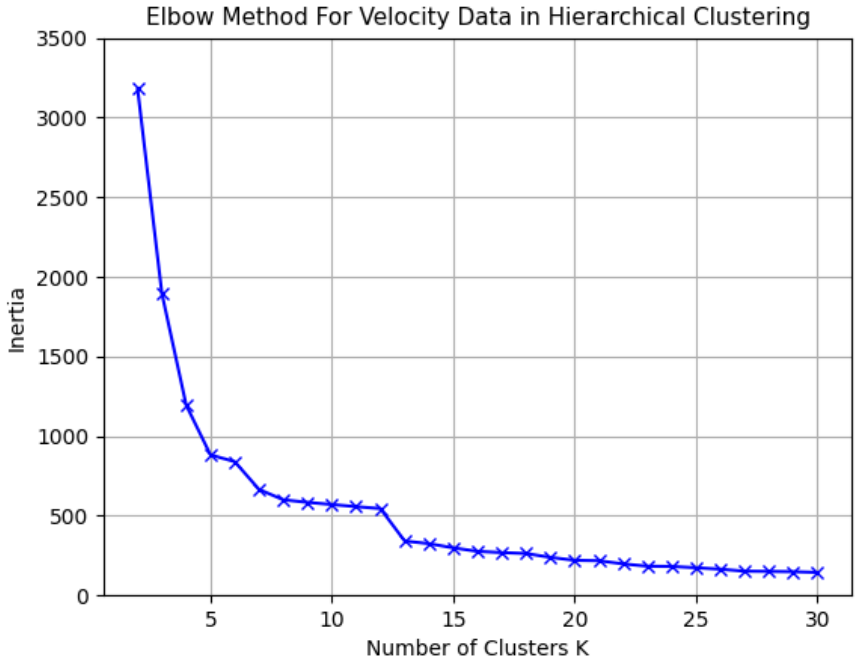


(b) Elbow Method for Force Data in K-means Clustering

Fig. 5.1 Elbow Method in K-means Clustering

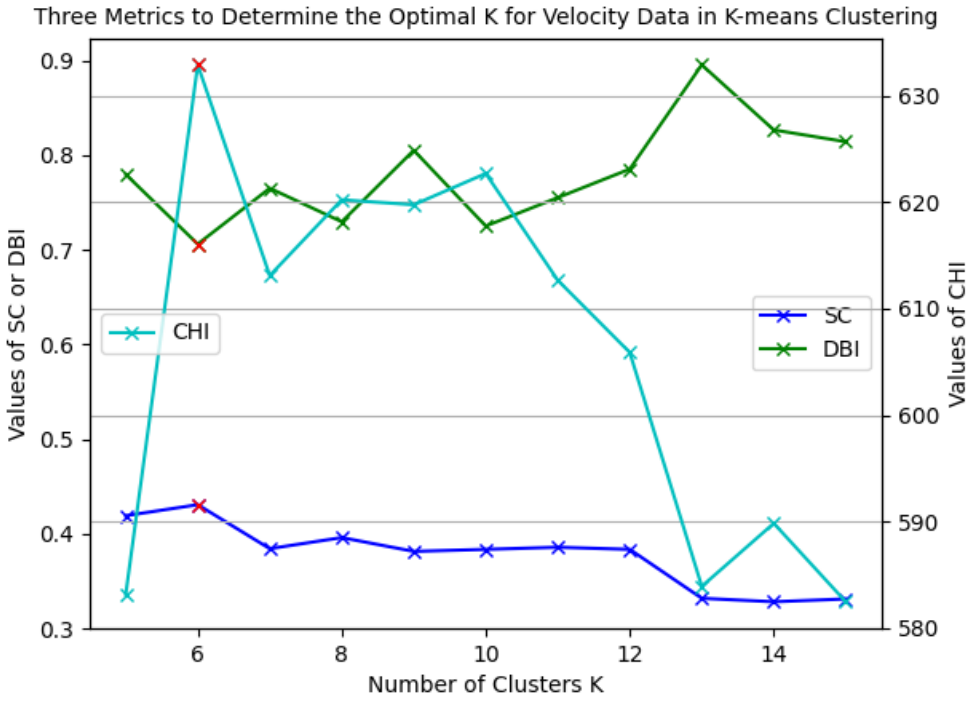


(a) Elbow Method for Velocity Data in Hierarchical Clustering

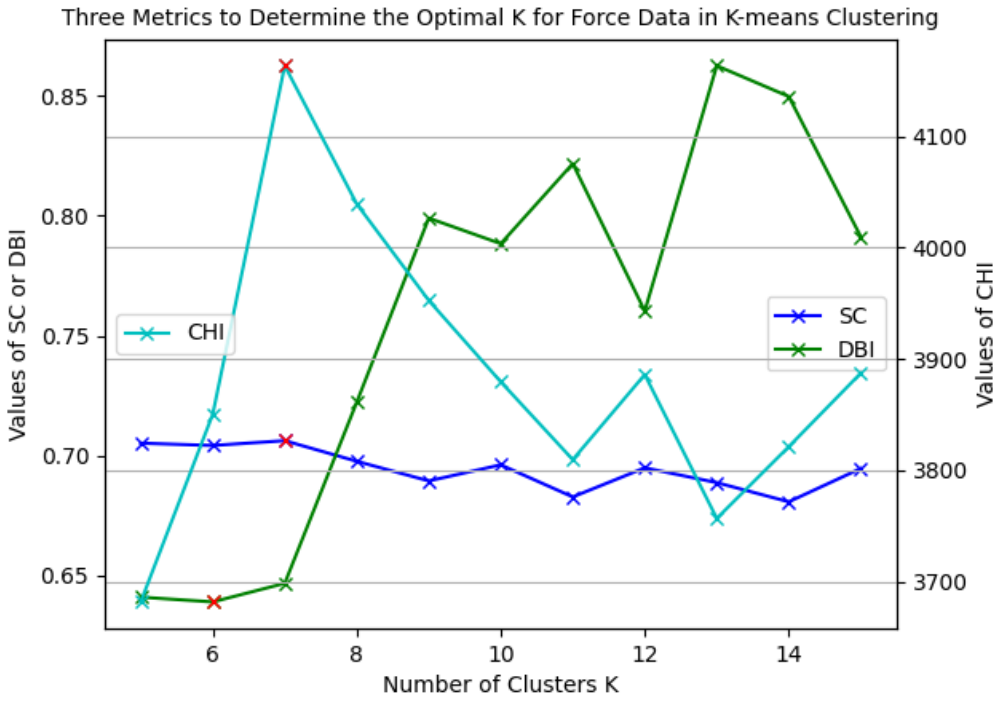


(b) Elbow Method for Force Data in Hierarchical Clustering

Fig. 5.2 Elbow Method in Hierarchical Clustering

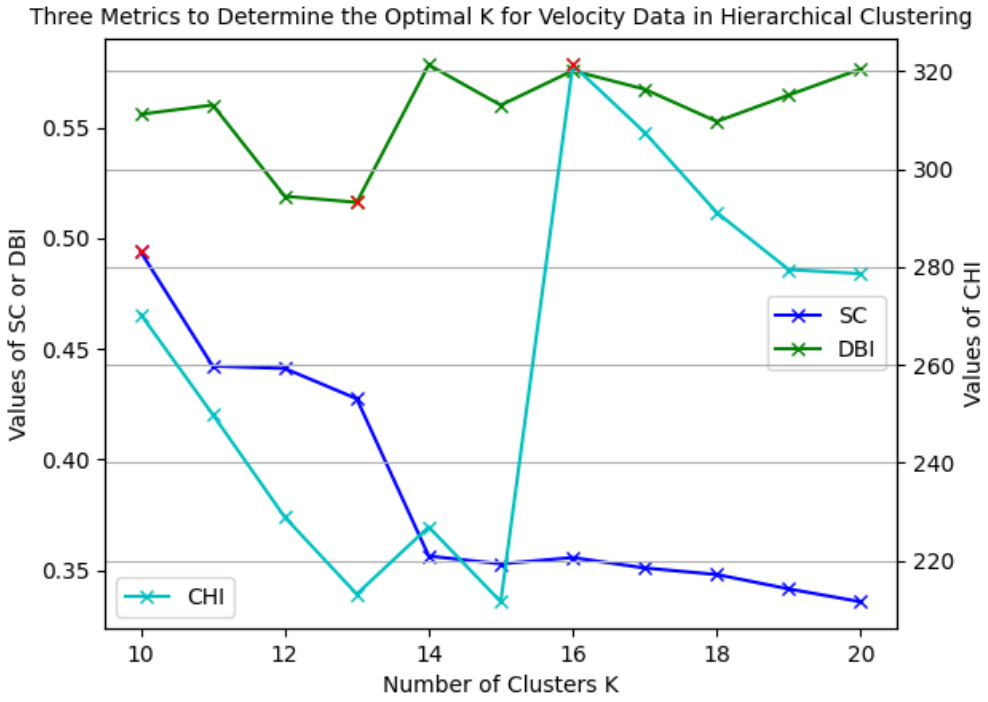


(a) Three Metrics to Determine the Optimal K for Velocity Data in K-means Clustering

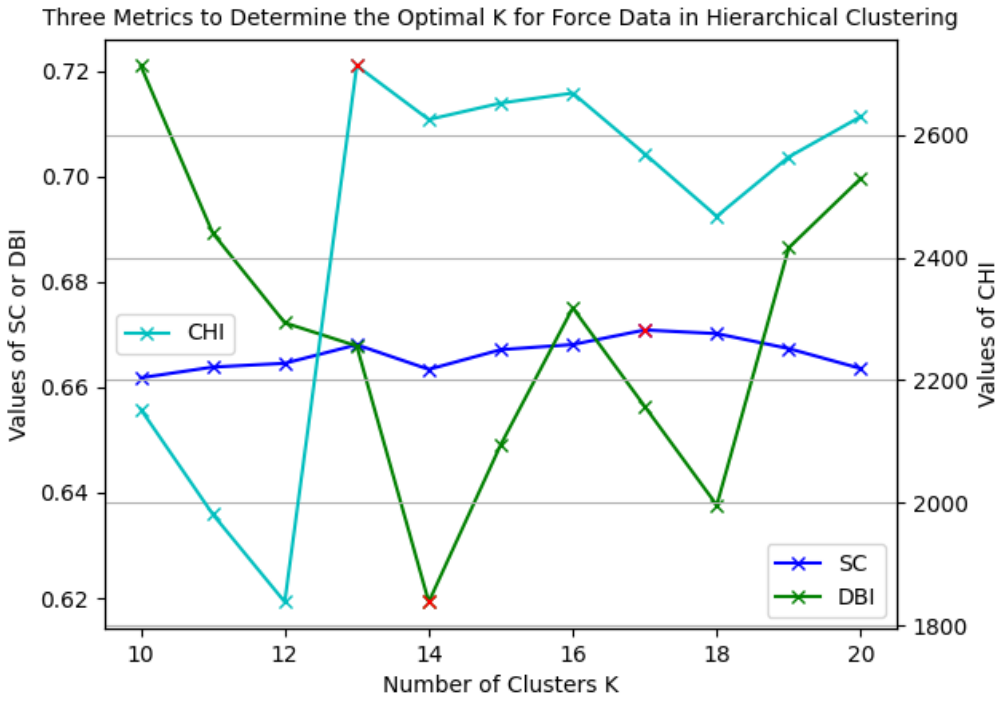


(b) Three Metrics to Determine the Optimal K for Force Data in K-means Clustering

Fig. 5.3 Three Metrics to Determine the Optimal K in K-means Clustering



(a) Three Metrics to Determine the Optimal K for Velocity Data in Hierarchical Clustering



(b) Three Metrics to Determine the Optimal K for Force Data in Hierarchical Clustering

Fig. 5.4 Three Metrics to Determine the Optimal K in Hierarchical Clustering

After giving a fuzzy range of optimal K for two unsupervised clustering techniques, an accurate K can be determined under the judgement of three metrics (SC, DBI, and CHI). As is mentioned in Section 5.2, the preferred number of clusters K should have a lower value of DBI, and higher values of both SC and CHI, contributing to well separated clusters in which objects are highly similar. Fig. 5.3a shows how values of three metrics vary with the number of clusters K. The maximum of SC and CHI, and the minimum of DBI are all marked in red. Therefore, for velocity data in k-means clustering, the optimal K should be 6. Similarly, the optimal K for force data in k-means clustering is also 7 even if the minimum of DBI is at 6. The difference between DBIs at 7 and 6 is 0.008, which is too small to make a big difference.

When it comes to hierarchical clustering, the optimal K for velocity data is hard to determine. As is depicted in Fig. 5.4a, three maximum and minimum distribute at three different Ks. What's more, values of three metrics fluctuate rapidly. Therefore, we choose $K = 10$ as it creates minimum of clusters. For force data, the optimal K is 14 since the value of SC changes steadily, and the difference between CHIs at 13 and 14 is 87.69, which is small enough to be noticeable.

To sum up, the optimal K for two unsupervised clustering techniques are estimated by elbow method at first. Then based on three metrics, the accurate optimal K for different types of kinesthetic used in two techniques can be determined.

5.4 Comparison of Unsupervised Clustering Techniques, PD-based Codecs and LSTM-based Kinesthetic Data Reduction Techniques

This section compares the performance of unsupervised clustering techniques with traditional PD-based codecs and proposed LSTM-based kinesthetic data reduction techniques. Only data in a part of clusters is transmitted over the network when unsupervised clustering techniques are applied. Specifically, clusters hold maximum number of data are selected to be transmitted. Otherwise, only centroids of other clusters are transmitted once. Concretely, for k-means clustering techniques, top two clusters with the maximum size of data are transmitted. Simultaneously, four clusters of data classified by the hierarchical clustering is transmitted.

In the beginning, HPW-PSNRs of four kinesthetic data reduction techniques are compared in Fig. 5.5& 5.6. For velocity data, the LSTM-based data reduction model perform better than the other three techniques. It has nearly 3 dB greater than PD-based codecs.

Meanwhile, k-means clustering, hierarchical clustering and PD-based codecs have similar HPW-PSNR, which approximately equal to 39 dB.

However, unsupervised clustering techniques perform better when dealing with force data. As is shown in Fig. 5.6, k-means has the highest value of HPW-PSNR at average 30 dB, which is 6 dB greater than PD-based codecs and 8 dB greater than LSTM-based models. We can conclude that LSTM networks are suitable for reducing velocity data, while unsupervised clusterings can be applied in force data.

Comparison of the transmission rate can be found in Fig. 5.7. As is shown in Fig. 5.7a, the transmission rate of clustering techniques is a constant when the number of clusters K and the number of top clusters $k_p \& k_f$ to be transmitted the network are given. In this case, 40.6% of velocity signals is transmitted when using hierarchical clustering, and 37.0% for k-means clustering. On top of that, for velocity signals, K_{km} is 6, K_{hie} is 10, TC_{km} is 2 and TC_{hie} is 4. Both unsupervised clusterings have similar rates of transmission. Comparing with LSTM networks, less data requires to be transmitted when α_v is from 0 to 0.1. Otherwise, clusterings perform better than traditional PD-based codecs except α_v is greater than 0.2.

We can draw similar conclusions when analysing the transmission rate for force signals. In this case, K_{km} is 7, K_{hie} is 14, TC_{km} is 2 and TC_{hie} is 4. Comparing with LSTM-based models and PD-based codecs, unsupervised clustering techniques have lower rates when α_f is less than or equal to 0.1. In this case, 36.8% of force signals is transmitted when using hierarchical clustering, and 35.2% for k-means clustering. As long as α_f is greater than 0.1, LSTM-based models is supposed to be the first choice for reducing force data transmission.

To put it concisely, unsupervised clustering techniques are suitable for the situation of lower deadband parameters. In other words, when the difference between real and predicted stimuli is easy to be observed by users, unsupervised clustering techniques can be selected. For the transparency of the whole system is hard to degrade, i.e. a higher deadband parameter, LSTM-based models can be used.

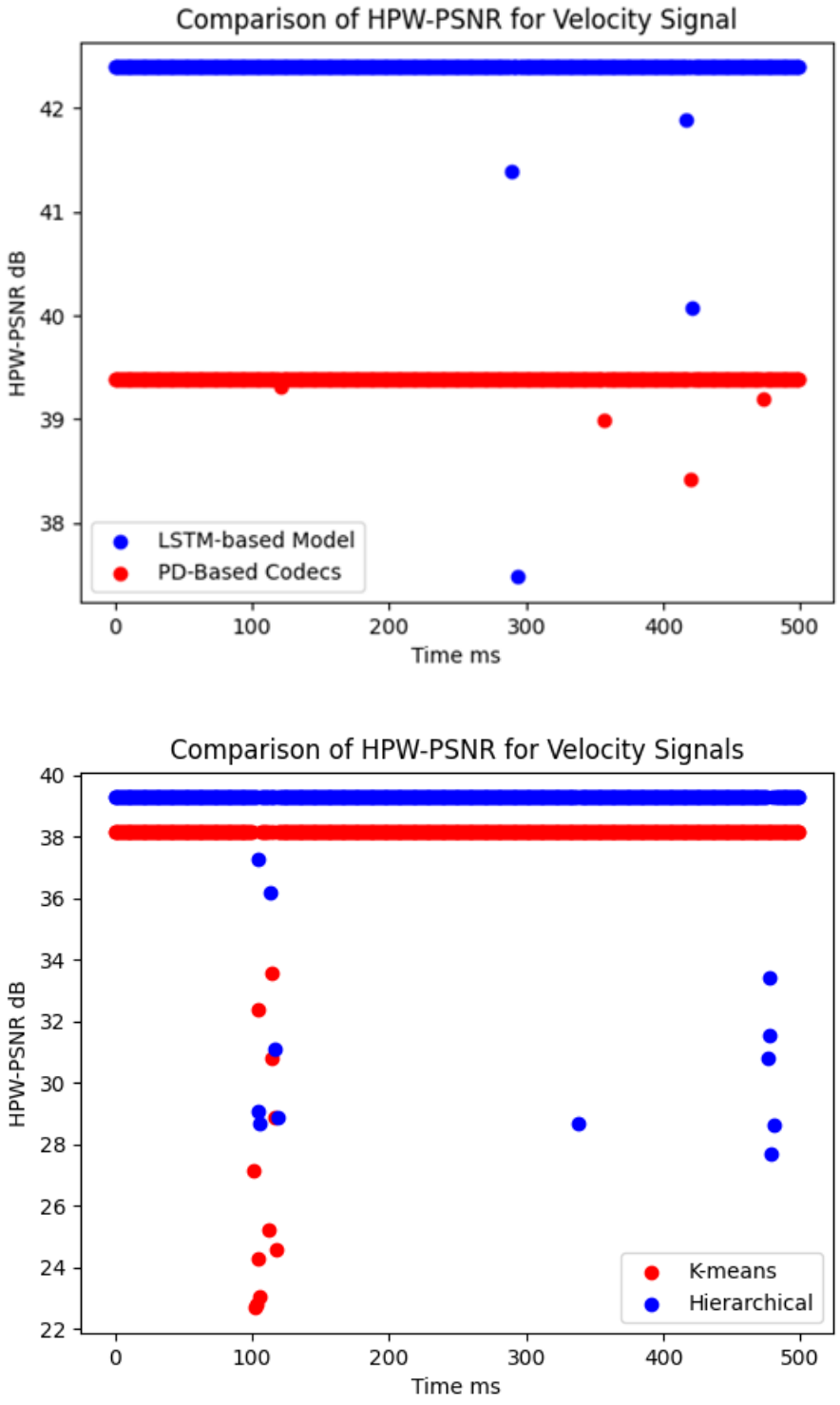


Fig. 5.5 Comparison of HPW-PSNR for Force Signals When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\ \mathrm{N \cdot s/m^2}$, $\alpha_v = 0.1$, $k = 1$, $C = 1$, $N = 500$

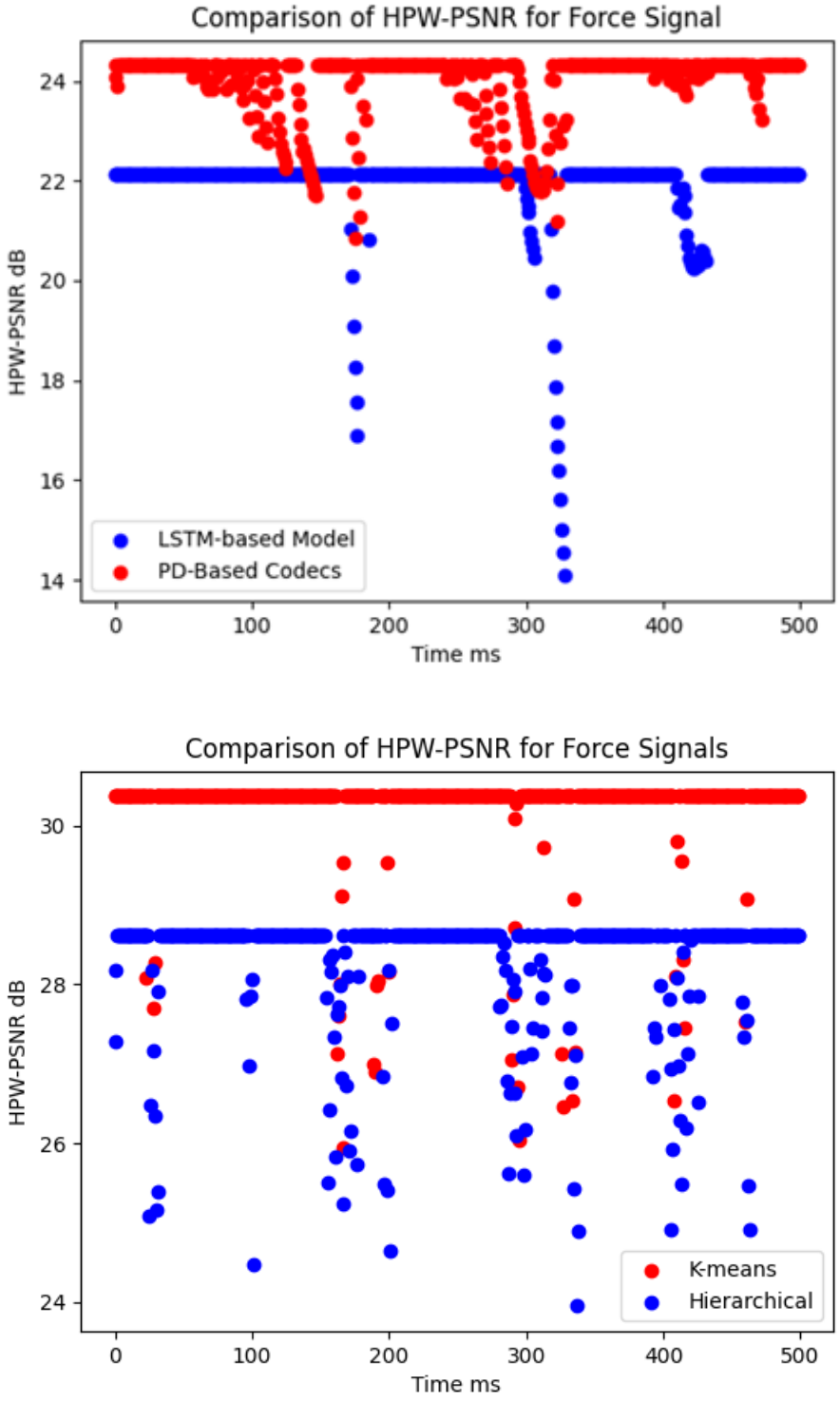
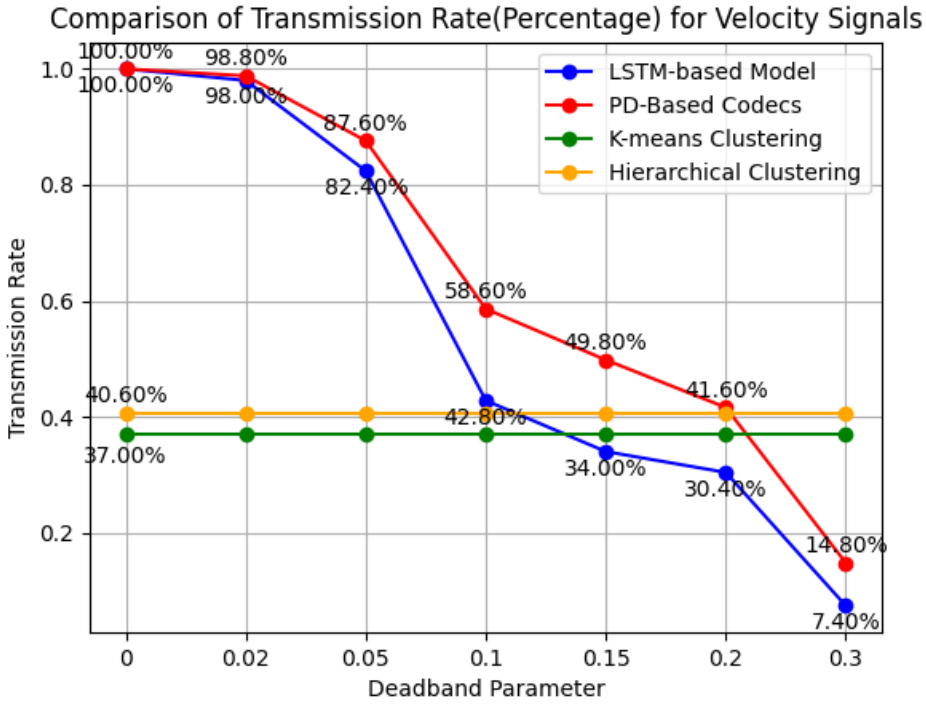
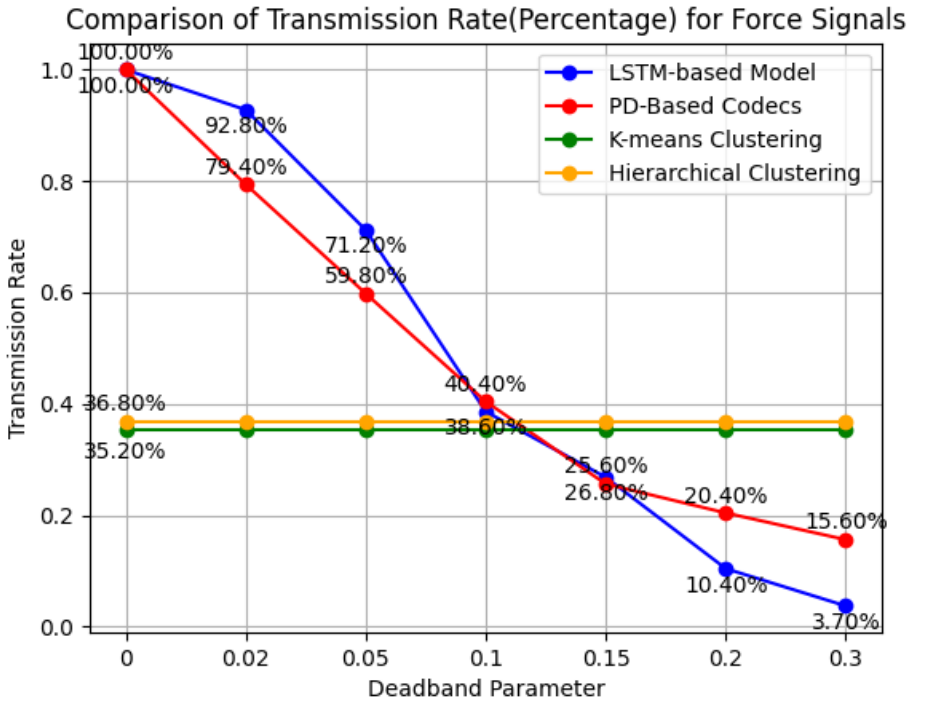


Fig. 5.6 Comparison of HPW-PSNR for Force Signals When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $\delta = 0.9\ \mathrm{N \cdot s/m^2}$, $\alpha_v = 0.1$, $k = 1$, $C = 1$, $N = 500$



(a) Comparison of Transmission Rate (Percentage) for Velocity Signals When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $K_{km} = 6$, $K_{hie} = 10$, $TC_{km} = 2$, $TC_{hie} = 4$

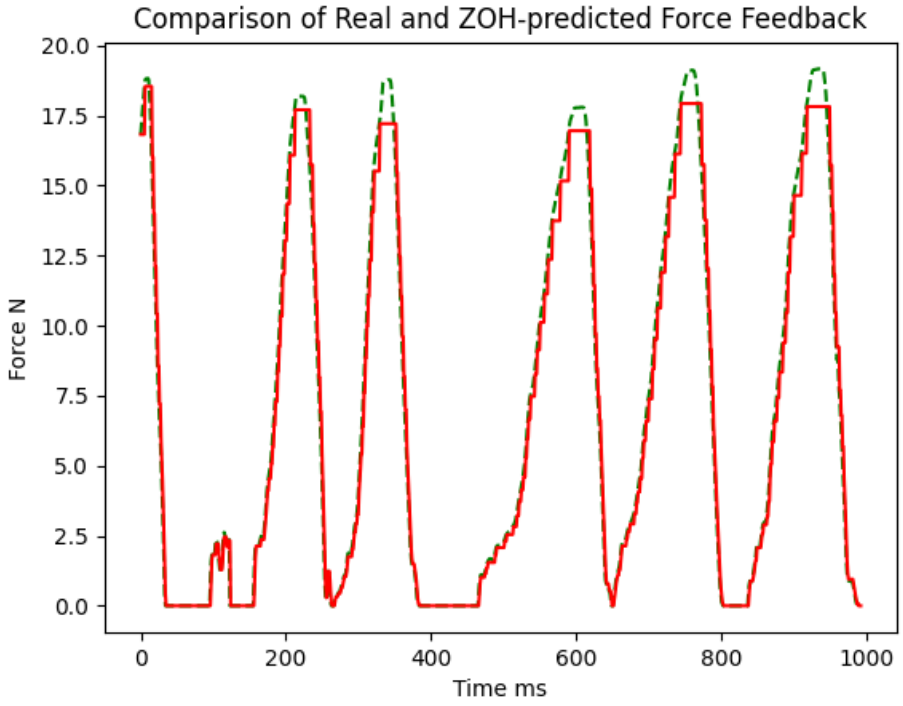


(b) Comparison of Transmission Rate (Percentage) for Force Signals When Tapping on the Foam Surface, Timestep=100 ms, $\mu = 0.8$, $K_{km} = 7$, $K_{hie} = 14$, $TC_{km} = 2$, $TC_{hie} = 4$

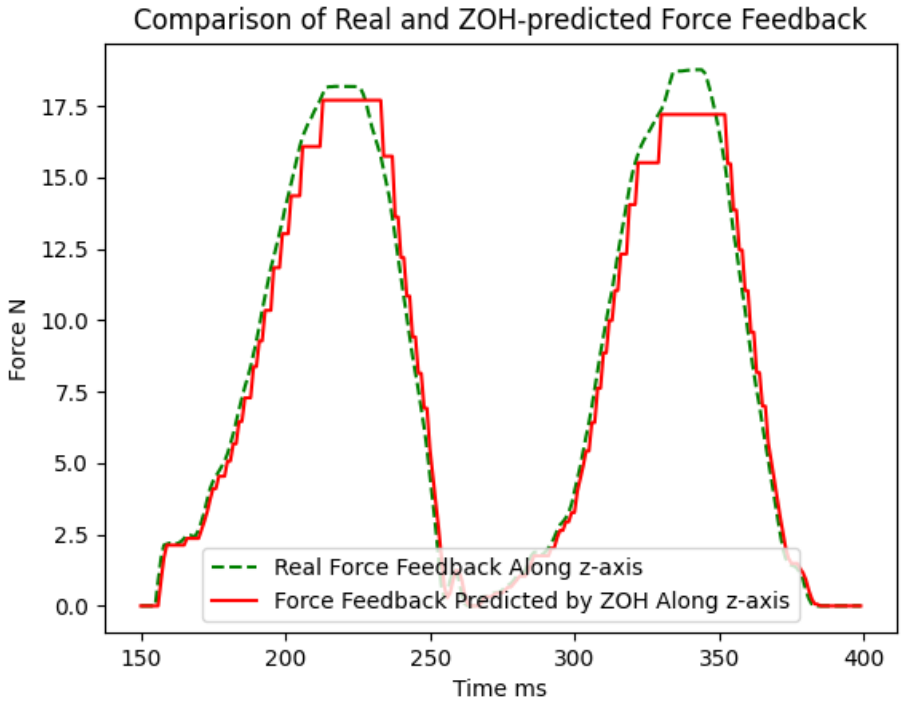
Fig. 5.7 Comparison of Transmission Rate (Percentage) When Tapping on the Foam Surface

5.5 Transparency Comparison of GBDT-Based Predictive Scheme, ZOH and FOLP Predictors

Transparency always reflects the fidelity of the predictive scheme. A predictive scheme with higher transparency can not only reduce the overall latency effectively, but increase the immersion of the users. In order to test the performance of the proposed scheme, we use the collected force data when tapping on the marble surface. There are totally 1000 sets of force data are used in this section. Before the testing process, a force data set of size 20000 has been used to train the GBDT model. After training 5000 epochs, the overall loss (MSE) is less than 10^{-3}. In this section, the proposed GBDT-based force predictive technique is compared transparency with the traditional ZOH and FOLP predictors. HPW-PSNR and predictive rate are used to measure the transparency. Moreover, whether the difference between the predicted and real data is noticeable to the human OP is also discussed.

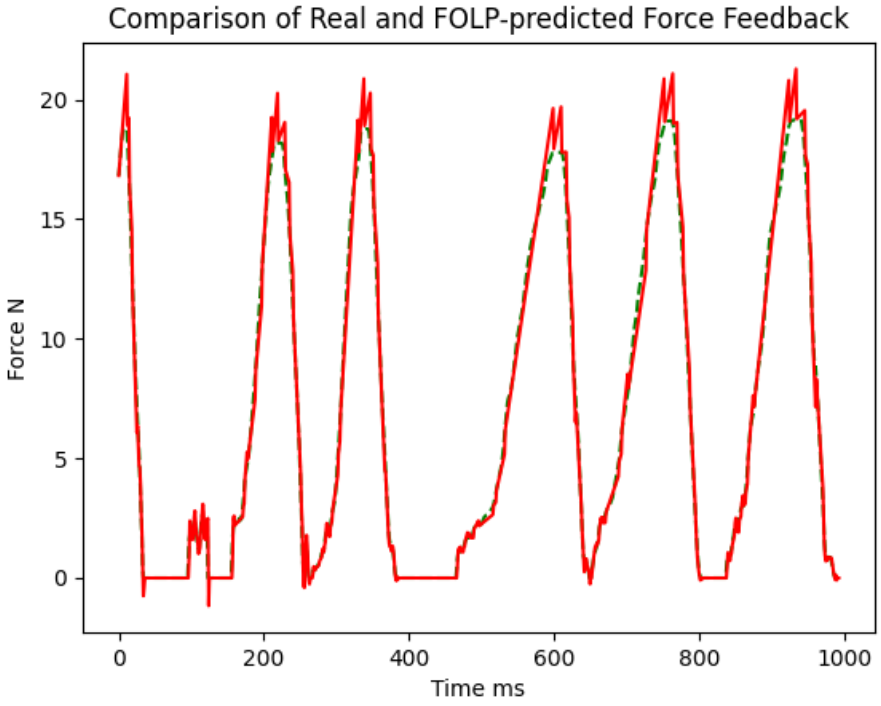


(a) Time=0-1000 ms

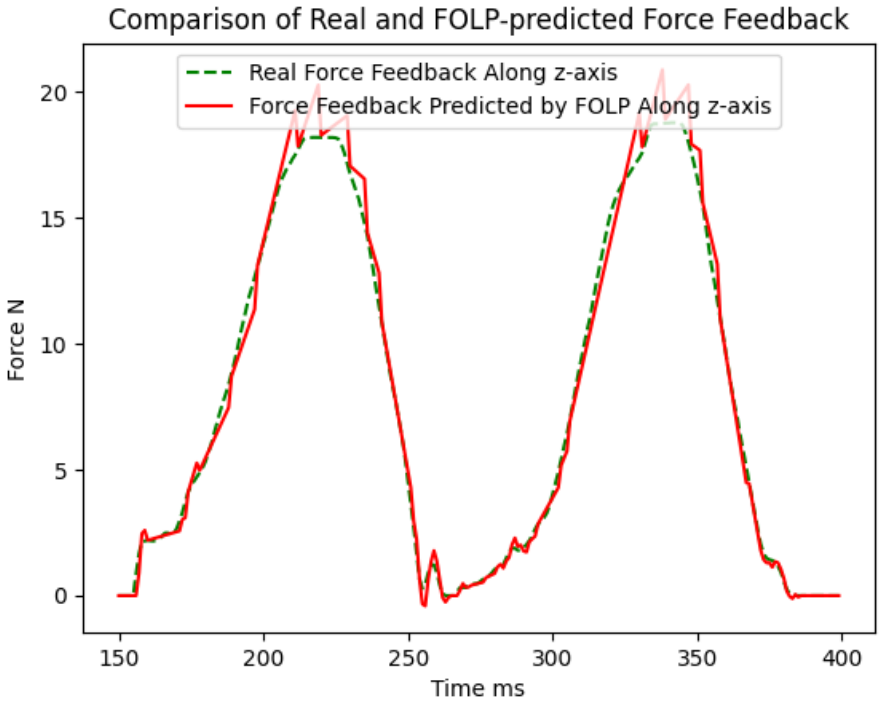


(b) Time=150-400 ms

Fig. 5.8 Comparison of Real and ZOH-predicted Force Feedback When Tapping on the Marble Surface, $\alpha_f = 0.1$

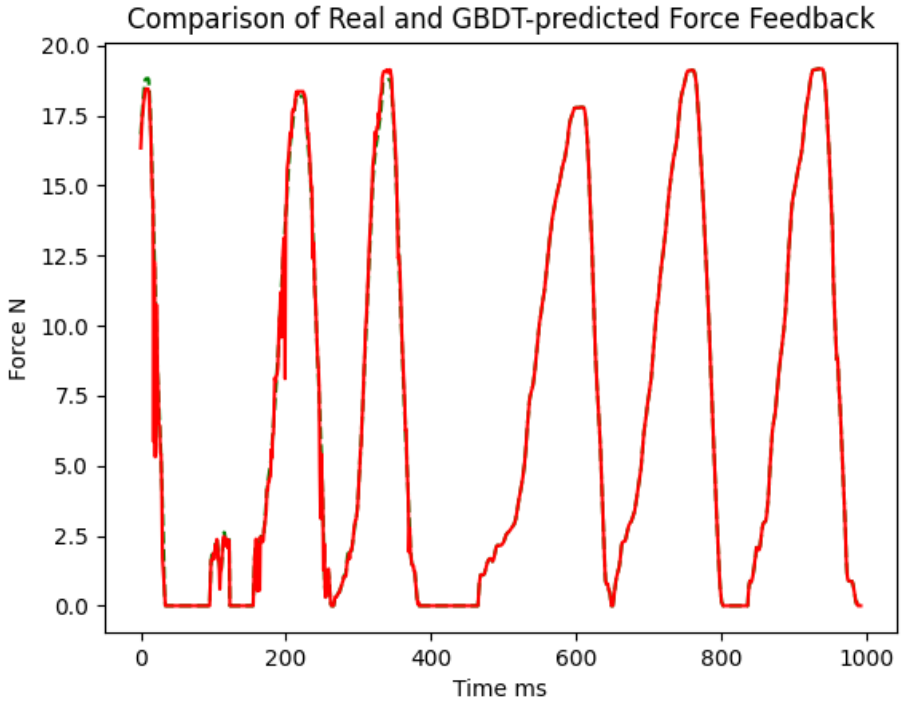


(a) Time=0-1000 ms

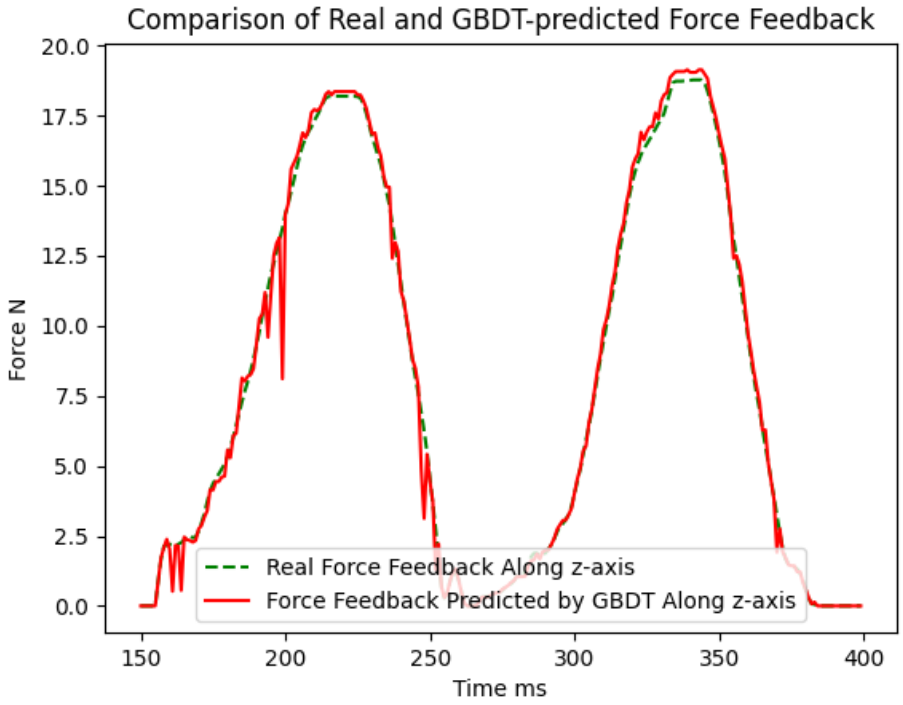


(b) Time=150-400 ms

Fig. 5.9 Comparison of Real and FOLP-predicted Force Feedback (N) When Tapping on the Marble Surface, $\alpha_f = 0.1$



(a) Time=0-1000 ms



(b) Time=150-400 ms

Fig. 5.10 Comparison of Real and GBDT-predicted Force Feedback (N) When Tapping on the Marble Surface, n_e=1000, lr=0.01, $\alpha_f = 0.1$

As is shown in Fig. 5.8 to Fig. 5.10, comparisons with real and predicted force feedback obtained from three predictive schemes are given. The previous explanation in Section 2.2.3 has illustrated that, when the difference between the predicted and real value has a noticeable influence to users, predictive schemes will trigger model updates. Therefore, it can be found that the trend of predicted data follows the real one in Fig. 5.8& to Fig. 5.10. As is shown in Fig. 5.8b,the predicted force feedback is stepped, as the predicted force back follows the last received data in ZOH scheme. Also, the predicted force feedback from FOLP is linear in Fig. 5.9b. However, in Fig. 5.10b, the predicted force feedback from the GBDT model is jagged since each prediction is from a leaf node of a decision tree. The value of each node is fixed. On top of that, trends of both real and predicted values are mostly the same, which benefits from trained GBDT models on both OP and TOP sides before transmission.

Furthermore, the predictive rate of GBDT-predicted force is higher than the other two schemes. In this case, we assume the noticeable threshold for force α_f is 0.1, and the corresponding predictive rate of GBDT models is 92.3%, 76.0% for ZOH schemes, and 83.4% for FOLP. Therefore, from the perspective of subjective tests, the GBDT-based predictive scheme remains the highest quality of force feedback when predictive modules are applied in bilateral teleoperation systems.

Another figure for describing the transparency of three different predictive schemes is given in Fig. 5.11. The average HPW-PSNR of GBDT-predicted data is -21.2 dB, which is higher than the other two schemes. The average HPW-PSNR of data predicted by ZOH scheme is -24.50 dB, and -23.48 dB for FOLP scheme. What's more, all of 3 predictive schemes have distortions in the time 0 to 400 ms. However, the distortion of GBDT-predicted values become smoother after 400 ms, which follows the predicted force simulation in Fig. 5.10b. In other words, GBDT predictive scheme can preserve better transparency than the other two techniques in terms of predictive rate and HPW-PSNR.

Numbers of model updates for 3 different predictive schemes are depicted in Fig. 5.12. The number of model updates reflects the stability of the entire system. The fewer model updates, the more stable the system. Model updates happen when the difference between the predicted and real value is perceivable to the user. Therefore, satisfactory predictive schemes normally require fewer model updates, which means less data to be transmitted via the backward channel. At the default perceptual threshold of $\alpha_f = 0.1$, the GBDT model needs 77 model updates, while 171 times in FOLP schemes and 271 in ZOH schemes. All of them can reduce the transmitted packets from the TOP to OP. As is concerned, only 77 model updates in GBDT, 171 in FOLP, and 271 in ZOH which are lower than 1000 (the original amount of data packets transmitted in 1000 ms). Instead, it is obvious that the number of updates for GBDT scheme is smaller than the other two schemes in terms of

different deadband parameters. FOLP scheme is second stable, while ZOH scheme is the most unstable predictive technique for bilateral teleoperation systems.

To summarize, the proposed GBDT-based predictive scheme can hold a higher accuracy pf prediction than the other two traditional schemes from two aspects: i) predictive rate; ii) transparency. Meanwhile, model updates of GBDT model is fewer, which guarantees the stability of the whole system.

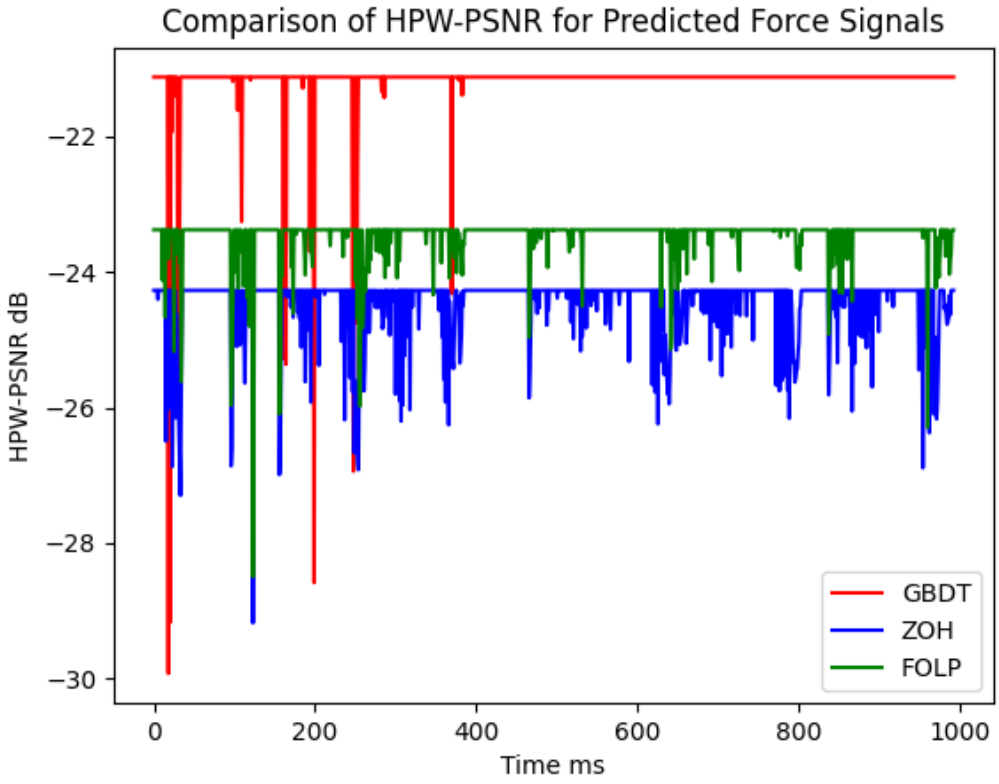


Fig. 5.11 Comparison of HPW-PSNR for Predicted Force Signals, $\alpha_f = 0.1$, $k = 1$, $C = 1$, $N = 1000$, Time=0-1000 ms

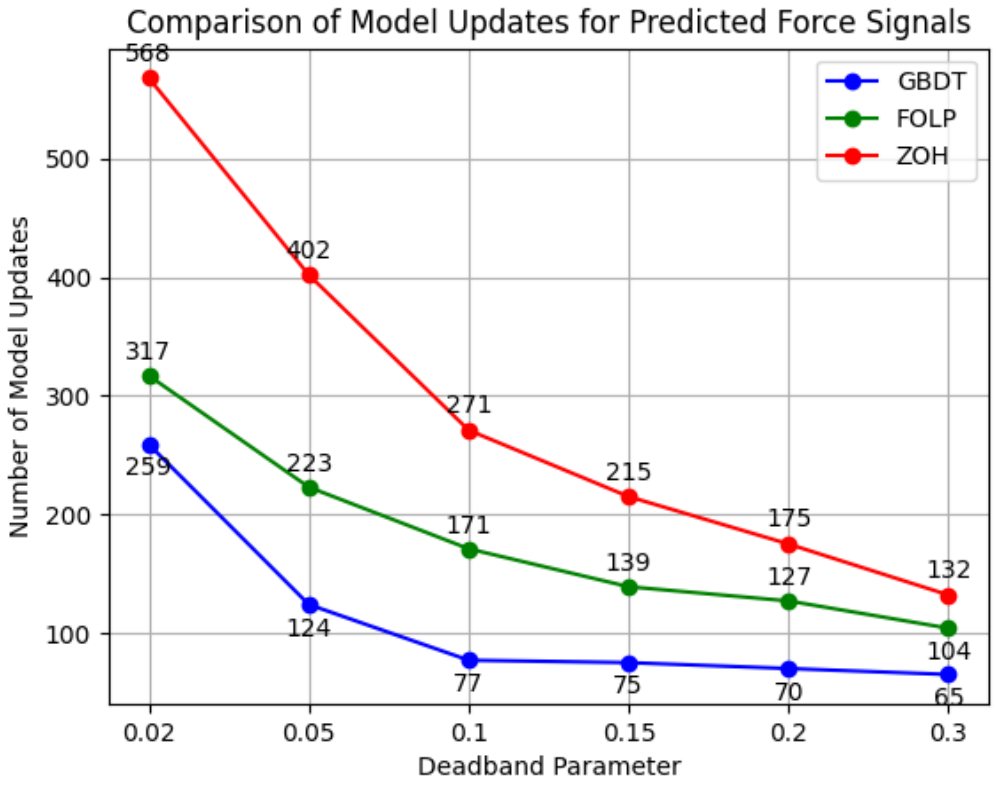


Fig. 5.12 Comparison of Model Updates for Force Signals

5.6 Conclusions

5.6.1 Discussion

In this chapter, results of the performance of two unsupervised clustering techniques are given in the first part. Also, further comparison with PD-based codecs and proposed LSTM-based mathematical model is also given. Kinesthetic data reduction module with unsupervised clustering techniques can be viewed as selecting a subset of data in clusters for transmission, and replacing the rest of the data with centroids of other clusters. Therefore, unsupervised clustering techniques are similar to LSTM-based models, they both use mathematical models to selectively transmit data in bilateral teleoperation systems.

The first step of realizing unsupervised clustering techniques is to find the optimal number of clusters K. Therefore, elbow method is introduced by finding a turning point of inertia, before which the slope decreases dramatically and after which the slope drops gently. Then, approximate ranges for two unsupervised clustering techniques can be determined. However, in Fig. 5.1 and Fig. 5.2, we can see that the average inertia of hierarchical clustering is higher than that of k-means clustering, and the slope of inertia of hierarchical clustering is not as smooth as that of k-means clustering. Thus, the range of K of hierarchical clustering is slightly greater.

Based on the obtained ranges, three metrics, i.e., CHI, SC and DBI, are used to determine the accurate K. The optimal values of these metrics are highlighted in red in Fig. 5.3 and Fig. 5.4. For k-means clustering, it is obvious that K for velocity data is 6, and 7 for force data. For hierarchical clustering, values of three metrics vary a lot with K, which is caused by the rugged slope of inertia. We must find the value of K corresponding to the closest values of the best value of the three indicators. Thus, 10 is selected as the K for velocity data in hierarchical clustering, and 14 for force data.

Furthermore, comparison of unsupervised clustering techniques, PD-based codecs and LSTM-based models is given. PD-based codecs and LSTM-based model can determine the transmission status of each data as long as it is collected, however, like DRTs, unsupervised clustering techniques can only determine the transmission status after collecting a certain amount of data ($M = 1000$). From the results, we can see that the average HPW-PSNR of LSTM-based mathematical model is 42.3 dB, which is approximately 3 dB higher than that of the other three techniques. The reason for that can be interpreted as the unsupervised clustering techniques can introduce more bias when dealing with high-frequency signals. Nevertheless, when dealing with low-frequency signals, i.e., force feedback, unsupervised clustering techniques perform better LSTM-based models and PD-based codes. When it comes to the transmission rate, both $\alpha_v = 0.1$ and $\alpha_f = 0.1$ can be regarded as critical points for judging the separation of clustering techniques and

the performance of mathematical models. When deadband parameter is greater than 0.1, LSTM-based mathematical model is preferred, otherwise, we can choose unsupervised clusterings. Both k-means and hierarchical clusterings have similar performances in terms of maintaining the transparency and data transmission rate.

The second part in this chapter analyses the performances of the proposed GBDT-based predictive scheme by comparing it with ZOH and FOLP schemes in two aspects, i)the overall predictive rate; ii)the amount of model updates. Comparing with two traditional predictive schemes, GBDT-based scheme can increase the predictive rate effectively and reduce the amount of model updates dramatically. The size of training data is 20000, which is much bigger than the testing data size. Thus, compared to 76.0% predictive rate for ZOH scheme 83.4% for FOLP, the overall predictive rate of the GBDT model is 92.3%, representing the GBDT-based predictive scheme remains the highest quality of force in bilateral teleoperation systems. Besides, the highest HPW-PSNR of GBDT-based predictive scheme also supports the above point.

As for the amount of model updates, GBDT-based predictive schemes also show good performance in maintaining the stability of the whole system. Traditional ZOH and FOLP schemes require far more model updates than GBDT-based schemes. This is because GBDT-based schemes are trained before testing, and many potential model updates occur during training. Thus, as long as a well-trained GBDT-based predictive model is derived, its performance will far exceed that of traditional ZOH and FOLP schemes. However, a large kinesthetic dataset is required prior to training.

5.6.2 Summary

In this chapter, we analysed two unsupervised clustering techniques in terms of transparency and transmission rate. Also, we discussed three metrics to analyse the performance of clustering techniques, which are SC, CHI and DBI.

As a result, by using the elbow method and comparing 3 metrics at different number of clusters K, the optimal K for velocity data in k-means clustering is 6, and K is 7 for force data. When it comes to hierarchical clustering, the optimal K for velocity data is 10, and K is 14 for force data.

Compared to LSTM-based models and PD-based codecs, clustering techniques have higher HPW-PSNRs for force feedback when $\alpha_f = 0.1$. However, the LSTM-based model has higher HPW-PSNR when dealing with velocity data, and $\alpha_v = 0.1$. In terms of the transmission rate, the LSTM-based model is more suitable for the case of high deadband parameters, while unsupervised clusterings can effectively reduce the transfer rate when the deadband parameters are low.

In addition, three different predictive techniques are compared in terms of transparency (predictive rate and HPW-PSNR) and stability (number of model updates). GBDT is a machine learning-based algorithm which uses a group of boosting function to find the mathematical relationship between the force feedback and position through classification.

We have found that GBDT schemes perform better than traditional ZOH and FOLP schemes. From the angle of transparency, GBDT predictive schemes have a 8.9% higher predictive rate than FOLP, and 16.3% higher than ZOH when $\alpha_f = 0.1$. What's more, the average HPW-PSNR for GBDT schemes is -21.2 dB, which is 3.3 dB greater than ZOH, and 2.28 dB greater than FOLP. When it comes to the number of updates, the number of model updates for GBDT schemes is only a half of the FOLP, while a third of the ZOH, which means that GBDT schemes are much more stable than the other 3 schemes.

To summarize, the proposed bilateral teleoperation system with unsupervised clustering techniques can shows good performance when dealing with velocity and force data as long as the optimal number of clusters K is determined. As for transparency, clustering techniques have a higher HPW-PSNR than LSTM-based models and PD-based codecs in terms of force data. But for velocity data, LSTM-based models have a higher HPW-PSNR than clustering techniques and PD-based codec. As for transmission rate, the proposed bilateral teleoperation system with unsupervised clustering techniques can effectively reduce the transmission rate when deadband parameter is lower than 0.1. Therefore, instead of applying LSTM-based models, unsupervised clustering kinesthetic data reduction techniques can be applied in the situation of requiring higher transparency.

Simultaneously, the proposed bilateral teleoperation system with GBDT predictive schemes has higher performance for preserving the transparency and stability. The GBDT model is the most popular predictive model in the industry, and it is employed on the haptic communication system that can not only reduce the transmitted packets but enhance the transparency in the whole system.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

It was the aim of this thesis to pose and answer many unsolved questions relating to kinesthetic data reduction techniques in bilateral teleoperation systems. As with any scientific work, it has brought up more questions than is has solved, some of which are illuminated below. Before that, however, contributions of the thesis are glued together to give a better picture of the choice of the research conducted.

In this thesis, the idea of kinesthtic data reduction in bilateral teleoperation systems is introduced, thereby various machine learning algorithms are applied to the selective transmission or downscaling of data size. It has been demonstrated that such ideas can effectively reduce the kinesthetic data transmission over the network without significant loss of transparency. The predictive scheme has also been improved by introducing GBDT to derive a mathematical model. Thus, the performances of both data reduction and predictive modules have been improved dramatically.

Bilateral teleoperation systems with kinesthetic data reduction and predictive modules for achieving the 1-ms challenge in haptic communications are proposed in Chapter 2. It has been reiterated that the instability due to the transmission delay in bilateral teleoperation systems, proposing ideas to reduce the transmission delay is appropriate. Thus, the idea of kinesthetic data reduction techniques have been proposed. Consecutive analysis throughout the thesis therefore distinguished between three different data reduction techniques and one predictive scheme.

Although the topic of PD-based codecs has been proposed for kinesthetic data reduction, novel techniques based on machine learning algorithms have been obtained in Chapter 2. Both LSTM-based mathematical model and unsupervised clusterings reduce kinesthetic data transmission by selectively transmitting data. However, LSTM network is a supervised learning which requires output labels during the training process. Unsupervised clusterings

can train the model without any labels. Besides, DRTs can reduce the size of data transmission each time, unlike selective transfers above. For increasing the predictive rate, GBDT-based predictive scheme has been proposed, which can be derived by training a large amount of kinesthetic data.

Simulation results and analysis of the proposed LSTM-based kinesthetic data reduction techniques are given in Chapter 3. We use the Phantom Omni haptic device to collect data on the virtually built environment for collecting kinesthetic data. The kinesthetic data format and the experimental process is also given in this chapter. As for the result analysis, it is illustrated that LSTM-based mathematical models perform well in both data reduction and perceptual transparency. Compared to the PD-based codecs, LSTM-based model transmits velocity/force signals with a lower rate than PD-based codecs as well as with less degradation. And both mathematical models have similar performance for maintaining transparency, while PD-based codecs have lower degradation. Additionally, for transmitting the force feedback of dragging, a higher packet rate is required. Thefore, we can see that LSTM-based mathematical model a higher performance for velocity transmission than the conventional PD-based codecs. For force transmission, the two systems perform similarly. Also, The proposed LSTM-based perceptual threshold is more practical and maintains the transparency.

In Chapter 4, we discussed the metrics for the performance of dimensionality reduction (CON, TRU, RRE, KL ACC) and data reconstruction techniques (HPW-PSNR). As a result, DRTs can reduce kinesthetic data transmission over the network effetively without degrading the transparency of the system. Among the proposed three DRTs, UMAP outperforms the other two techniques in preservation of global and local structure of the original data, and maintains good transparency in its corresponding data reconstruction technique. Furthermore, comparison with LSTM-based mathmatical models and PD-based codecs is also give. As a result, a selection of two types of techniques is required. In general, when the deadband parameter is relatively high, LSTM-based models are recommended. When it is small, UMAP can be a good choice.

In Chapter 5, we analysed two unsupervised clustering techniques in terms of transparency and transmission rate. Also, we discussed three metrics to analyse the performance of clustering techniques, which are SC, CHI and DBI. In addition, three different predictive techniques are compared in terms of transparency (predictive rate and HPW-PSNR) and stability (number of model updates). At first, we use Elbow method to determine the optimal number of clusters in two types of clustering techniques. Compared to LSTM-based models and PD-based codecs, clustering techniques have higher HPW-PSNRs for force feedback. However, the LSTM-based model has higher HPW-PSNR when dealing with velocity data. In terms of the transmission rate, the LSTM-based model is more suitable for the case of high deadband parameters, while unsupervised clusterings can

effectively reduce the transfer rate when the deadband parameters are low. We also find that GBDT schemes perform better than traditional ZOH and FOLP schemes. From the angle of transparency, GBDT predictive schemes have a higher predictive rate than both FOLP and ZOH predictors. What's more, the average HPW-PSNR for GBDT schemes is greater. When it comes to the number of updates, the number of model updates for GBDT schemes is only a half of the FOLP, while a third of the ZOH, which means that GBDT schemes are much more stable than the other 3 schemes. Thus, it is a good choice to select unsupervised clustering techniques when dealing with velocity and force data as long as the optimal number of clusters K is determined. And unsupervised clustering kinesthetic data reduction techniques can be applied in the situation of requiring higher transparency.

Altogether, the proposed bilateral teleoperation system with various kinesthetic data reduction techniques can effectively reduce data transmission compared with conventional PD-based codecs. LSTM-based model can be viewed as a trained version of PD-based codecs, which produces higher transparency and requires lower transmission rate. Simultaneously, UMAP in DRTs has shown great performance when deadband is low. In the case of that, UMAp could be a good choice. Besides, unsupervised clustering techniques have higher transparency than PD-based codecs and LSTM-based models. When users need a stable immersion to the remote environment, unsupervised clustering kinesthetic data reduction techniques can be applied. Human operators can select types of data reduction techniques to satisfy their requirements.

6.2 Future Work

In a further step, a bilateral teleoperation system with the cooperation of proposed kinesthetic data reduction and predictive techniques can be analysed and discussed. In current stage, for simplifying the system model and emphasizing kinesthetic data reduction, all of the predictions from predictive scheme are from ZOH predictors. But the author has illustrated that the novel GBDT-based predictive scheme has higher accuracy than ZOH scheme. Analysis of the performance of one system with two different algorithms can be discussed in the future.

Another issue is complexity of the model update in GBDT-based predictive schemes. GBDT is an offline predictive strategy, and once the model changes, all of the parameters in the model will change, which increase the potential offload of the network. An online predictive scheme can be found in the future.

Other machine learning algorithms can be applied in dealing with time-series kinesthetic data. For instance, the most popular transformer model, which is used to train Chat Generative Pre-trained Transformer(ChatGPT). The transformer model is a powerful neural

network architecture which captures long-range dependencies and models sequential data more efficiently than previous models. Thus, a further discussion of transformer model can be given in the future.

Additionally, the kinesthetic data collection experiment can be extended by moving the cursor on different objects. Current data we collect is on a surface, however, effects of the proposed models on different shape of surfaces can be discussed.

In conclusion, the ultimate purpose of this thesis was to positively contribute to kinesthetic data reduction, proposed various bilateral teleoperation systems with different data reduction techniques, and hopefully, to pose many questions that may catch the imagination of future researchers.

References

[1] Kron, A., Schmidt, G., Petzold, B., Zah, M., Hinterseer, P. & Steinbach, E. Disposal of explosive ordnances by use of a bimanual haptic telepresence system. *IEEE International Conference On Robotics And Automation, 2004. Proceedings. ICRA '04. 2004*. **2** pp. 1968-1973 Vol.2 (2004)

[2] Lawrence, D. Stability and transparency in bilateral teleoperation. *IEEE Transactions On Robotics And Automation*. **9**, 624-637 (1993)

[3] Radcliffe, J., Ga, B., Tan, H., Eberman, B., Srinivasan, M. & Cheng, B. Human Factors For The Design Of Force-Reflecting Haptic Interfaces. *Proc. ASME WAM*. **55** (1996,12)

[4] Colgate, J. & Brown, J. Factors affecting the Z-Width of a haptic display. *Proceedings Of The 1994 IEEE International Conference On Robotics And Automation*. pp. 3205-3210 vol.4 (1994)

[5] Buss, M. & Schmidt, G. Control Problems in Multi-Modal Telepresence Systems. (1999)

[6] Guo, F., Zhang, C. & He, Y. Haptic Data Compression Based on a Linear Prediction Model and Quadratic Curve Reconstruction. *Journal Of Software*. **9** (2014,11)

[7] Brandi, F. & Steinbach, E. Prediction techniques for haptic communication and their vulnerability to packet losses. *2013 IEEE International Symposium On Haptic Audio Visual Environments And Games (HAVE)*. pp. 63-68 (2013)

[8] Steinbach, E., Hirche, S., Ernst, M., Brandi, F., Chaudhari, R., Kammerl, J. & Vittorias, I. Haptic Communications. *Proceedings Of The IEEE*. **100**, 937-956 (2012)

[9] Hirche, S., Ferre, M., Barrio, J., Melchiorri, C. & Buss, M. Bilateral Control Architectures for Telerobotics. (2007,10)

[10] El Saddik, A. The Potential of Haptics Technologies. *Instrumentation & Measurement Magazine, IEEE*. **10** pp. 10 - 17 (2007,3)

[11] Anderson, R. & Spong, M. Bilateral control of teleoperators with time delay. *IEEE Transactions On Automatic Control*. **34**, 494-501 (1989)

[12] Ryu, J., Artigas, J. & Preusche, C. A passive bilateral control scheme for a teleoperator with time-varying communication delay. *Mechatronics*. **20** pp. 812-823 (2010)

[13] Ryu, J. Bilateral Control with Time Domain Passivity Approach Under Time-varying Communication Delay. *RO-MAN 2007 - The 16th IEEE International Symposium On Robot And Human Interactive Communication*. pp. 986-991 (2007)

[14] Lederman, S. & Klatzky, R. Haptic perception: a tutorial. *Attention, Perception & Psychophysics*. **71**, 1439-1459 (2009,10), https://doi.org/10.3758/APP.71.7.1439

[15] Verrillo, R. Vibration Sensation in Humans. *Music Perception*. pp. 281-302 (1992)

[16] Haigh, A., Apthorp, D. & Bizo, L. The role of Weber's law in human time perception. *Attention, Perception, & Psychophysics*. **83** (2020,10)

[17] Heath, M. & Manzone, J. Manual estimations of functionally graspable target objects adhere to Weber's law. *Experimental Brain Research*. **235** (2017,6)

[18] Bhardwaj, A., Chaudhuri, S. & Dabeer, O. Deadzone analysis of 2D kinesthetic perception. *2014 IEEE Haptics Symposium (HAPTICS)*. pp. 473-478 (2014)

[19] Bhardwaj, A., Dabeer, O. & Chaudhuri, S. Can we improve over weber sampling of haptic signals?. *2013 Information Theory And Applications Workshop (ITA)*. pp. 1-6 (2013)

[20] Tsumaki, Y. & Uchiyama, M. A model-based space teleoperation system with robustness against modeling errors. *Proceedings Of International Conference On Robotics And Automation*. **2** pp. 1594-1599 vol.2 (1997)

[21] Nasir, Q. & Khalil, E. Perception based adaptive haptic communication protocol (PAHCP). *2012 International Conference On Computer Systems And Industrial Informatics*. pp. 1-6 (2012)

[22] Tanaka, H. & Ohnishi, K. Haptic data compression/decompression using DCT for motion copy system. *2009 IEEE International Conference On Mechatronics*. pp. 1-6 (2009)

[23] Kuzu, A., Baran, E., Bogosyan, S., Gokasan, M. & Sabanovic, A. Wavelet packet transform-based compression for teleoperation. *Proceedings Of The Institution Of Mechanical Engineers, Part I: Journal Of Systems And Control Engineering*. **229**, 639-651 (2015)

[24] Otanez, P., Moyne, J. & Tilbury, D. Using deadbands to reduce communication in networked control systems. *Proceedings Of The 2002 American Control Conference (IEEE Cat. No.CH37301)*. **4** pp. 3015-3020 vol.4 (2002)

[25] Weber, E. De pulsu, resorptione, Auditu et tactu. Annotationes Anatomicae et physiologicae. (Prostat apud C.F. Koehler,1834)

[26] Steinbach, E., Hirche, S., Kammerl, J., Vittorias, I. & Chaudhari, R. Haptic Data Compression and Communication. *IEEE Signal Processing Magazine*. **28**, 87-96 (2011)

[27] Hinterseer, P., Steibach, E. & Chaudhuri, S. Model based data compression for 3D virtual haptic teleinteraction. *2006 Digest Of Technical Papers International Conference On Consumer Electronics*. pp. 23-24 (2006)

[28] Hinterseer, P., Hirche, S., Chaudhuri, S., Steinbach, E. & Buss, M. Perception-Based Data Reduction and Transmission of Haptic Data in Telepresence and Teleaction Systems. *IEEE Transactions On Signal Processing*. **56**, 588-597 (2008)

[29] Sakr, N., Georganas, N., Zhao, J. & Shen, X. Motion and Force Prediction in Haptic Media. *2007 IEEE International Conference On Multimedia And Expo*. pp. 2242-2245 (2007)

[30] Liu, X., Dohler, M. & Deng, Y. Vibrotactile Quality Assessment: Hybrid Metric Design Based on SNR and SSIM. *IEEE Transactions On Multimedia*. **22**, 921-933 (2020)

[31] Liu, X. & Dohler, M. Vibrotactile Alphabets: Time and Frequency Patterns to Encode Information. *IEEE Transactions On Haptics*. **14**, 161-173 (2021)

[32] Graves, A., Mohamed, A. & Hinton, G. Speech Recognition with Deep Recurrent Neural Networks. (arXiv,2013), https://arxiv.org/abs/1303.5778

[33] Sutskever, I., Vinyals, O. & Le, Q. Sequence to Sequence Learning with Neural Networks. (arXiv,2014), https://arxiv.org/abs/1409.3215

[34] Donahue, J., Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K. & Darrell, T. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. (arXiv,2014), https://arxiv.org/abs/1411.4389

[35] Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation*. **9**, 1735-1780 (1997,11), https://doi.org/10.1162/neco.1997.9.8.1735

[36] Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, C. & Courville, A. Towards end-to-end speech recognition with deep convolutional neural networks. *ArXiv Preprint ArXiv:1701.02720*. (2017)

[37] Cho, K., Van Merriënboer, B., Bahdanau, D. & Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *ArXiv Preprint ArXiv:1409.1259*. (2014)

[38] Yang, X., Sun, D., Zhu, R., Deng, T., Guo, Z., Ding, Z., Qin, S. & Zhu, Y. Aiads: Automated and intelligent advertising system for sponsored search. *Proceedings Of The 25th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining*. pp. 1881-1890 (2019)

[39] Hu, Z., Wang, Y., Peng, Q. & Li, H. Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm. (2019,3)

[40] Jiang, S., Mao, H., Ding, Z. & Fu, Y. Deep Decision Tree Transfer Boosting. *IEEE Transactions On Neural Networks And Learning Systems*. **31** pp. 383-395 (2020)

[41] Breiman, L., Friedman, J., Stone, C. & Olshen, R. Classification and Regression Trees. (Taylor & Francis,1984)

[42] Maaten, L., Postma, E. & Herik, H. Dimensionality Reduction: A Comparative Review. *Journal Of Machine Learning Research - JMLR*. **10** (2007,1)

[43] Hu, H. & Zahorian, S. Dimensionality reduction methods for HMM phonetic recognition. *2010 IEEE International Conference On Acoustics, Speech And Signal Processing*. pp. 4854-4857 (2010)

[44] Soni, A., Upadhyay, R. & Kumar, A. Dimensionality Reduction in Wireless Physical Layer Key Generation. *2019 IEEE 16th India Council International Conference (INDICON)*. pp. 1-4 (2019)

[45] Wang, G., Lauri, F. & Hajjam El Hassani, A. A Study of Dimensionality Reduction's Influence on Heart Disease Prediction. *2021 12th International Conference On Information, Intelligence, Systems & Applications (IISA)*. pp. 1-6 (2021)

[46] Salem, N. & Hussein, S. Data dimensional reduction and principal components analysis. *Procedia Computer Science*. **163** pp. 292-299 (2019), https://www.sciencedirect.com/science/article/pii/S1877050919321507, 16th Learning and Technology Conference 2019Artificial Intelligence and Machine Learning: Embedding the Intelligence

[47] Hotelling, H. Analysis of a complex of statistical variables into principal components.. *Journal Of Educational Psychology*. **24** pp. 498-520 (1933)

[48] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.*. **11** pp. 3371-3408 (2010,12)

[49] McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv Preprint ArXiv:1802.03426*. (2018)

[50] Myasnikov, E. Using UMAP for Dimensionality Reduction of Hyperspectral Data. *2020 International Multi-Conference On Industrial Engineering And Modern Technologies (FarEastCon)*. pp. 1-5 (2020)

[51] Peng, I., McFadden, M., Green, E., Iwabuchi, K., Wu, K., Li, D., Pearce, R. & Gokhale, M. UMap: Enabling Application-driven Optimizations for Page Management. *2019 IEEE/ACM Workshop On Memory Centric High Performance Computing (MCHPC)*. pp. 71-78 (2019)

[52] Umap, P., Chaudhari, K. & Joshi, M. Unsupervised singing voice separation from music accompaniment using robust principal componenet analysis. *2015 International Conference On Industrial Instrumentation And Control (ICIC)*. pp. 1433-1436 (2015)

[53] Hirche, S. & Buss, M. Human-Oriented Control for Haptic Teleoperation. *Proceedings Of The IEEE*. **100**, 623-647 (2012)

[54] Hirche, S., Bauer, A. & Buss, M. Transparency of haptic telepresence systems with constant time delay. *Proceedings Of 2005 IEEE Conference On Control Applications, 2005. CCA 2005.*. pp. 328-333 (2005)

[55] Steinbach, E., Strese, M., Eid, M., Liu, X., Bhardwaj, A., Liu, Q., Al-Ja'afreh, M., Mahmoodi, T., Hassen, R., El Saddik, A. & Holland, O. Haptic Codecs for the Tactile Internet. *Proceedings Of The IEEE*. **107**, 447-470 (2019)

[56] Hinterseer, P., Steinbach, E., Hirche, S. & Buss, M. A Novel, Psychophysically Motivated Transmission Approach For Haptic Data Streams in Telepresence and Teleaction Systems. *IN PROC. OF THE IEEE INT. CONF. ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING*. pp. 1097-1100 (2005)

[57] Hirche, S., Hinterseer, P., Steinbach, E. & Buss, M. Network traffic reduction in haptic telepresence systems by deadband control. *IFAC Proceedings Volumes*. **38**, 77-82 (2005)

[58] Bizo, L., Chu, J., Sanabria, F. & Killeen, P. The failure of Weber's law in time perception and production. *Behavioural Processes*. **71**, 201-210 (2006), Interval Timing: The Current Status

[59] Le, X., Ho, H., Lee, G. & Jung, S. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water*. **11** pp. 1387 (2019,7)

[60] Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation*. **9**, 1735-1780 (1997,11), https://doi.org/10.1162/neco.1997.9.8.1735

[61] Van Der Maaten, L., Postma, E., Herik, J. & Others Dimensionality reduction: a comparative. *J Mach Learn Res*. **10**, 13 (2009)

[62] Jolliffe, I. & Cadima, J. Principal component analysis: A review and recent developments. *Philosophical Transactions Of The Royal Society A: Mathematical, Physical And Engineering Sciences*. **374** pp. 20150202 (2016,4)

[63] Ruder, S. An overview of gradient descent optimization algorithms. *CoRR*. **abs/1609.04747** (2016), http://arxiv.org/abs/1609.04747

[64] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M. & Others Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *ArXiv Preprint ArXiv:1603.04467*. (2016)

[65] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances In Neural Information Processing Systems*. **30** (2017)

[66] Sakr, N., Georganas, N. & Zhao, J. A Perceptual Quality Metric for Haptic Signals. *2007 IEEE International Workshop On Haptic, Audio And Visual Environments And Games*. pp. 27-32 (2007)

[67] Li, S., Zang, Z. & Wu, L. Deep Manifold Transformation for Nonlinear Dimensionality Reduction. *ArXiv Preprint ArXiv:2010.14831*. (2020)

[68] Venna, J. & Kaski, S. Visualizing Gene Interaction Graphs with Local Multidimensional Scaling. (2006,1)

[69] Lee, J. & Verleysen, M. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*. **72**, 1431-1443 (2009), Advances in Machine Learning and Computational Intelligence

[70] Kullback, S. & Leibler, R. On information and sufficiency. *The Annals Of Mathematical Statistics*. **22**, 79-86 (1951)

[71] Shahabi, C., Ortega, A. & Kolahdouzan, M. A comparison of different haptic compression techniques. *Proceedings. IEEE International Conference On Multimedia And Expo*. **1** pp. 657-660 vol.1 (2002)

[72] You, Y. & Sung, M. Haptic Data Transmission Based on the Prediction and Compression. *2008 IEEE International Conference On Communications*. pp. 1824-1828 (2008)

[73] Wei, C. & Chiu, I. Turning telecommunications call details to churn prediction: a data mining approach. *Expert Systems With Applications*. **23**, 103-112 (2002)

[74] Pizzuti, C. & Procopio, N. A K-means Based Genetic Algorithm for Data Clustering. *International Joint Conference SOCO'16-CISIS'16-ICEUTE'16*. pp. 211-222 (2017)

[75] Alshareef, A., Wan Ishak, W., Md Norwawi, N. & Alkilany, A. Pattern discovery using k-means algorithm. (2014,10)

[76] Zheng, X., Lei, Q., Yao, R., Gong, Y. & Yin, Q. Image segmentation based on adaptive K-means algorithm. *EURASIP Journal On Image And Video Processing*. **2018** (2018,8)

[77] Toman, S., Abed, M. & Toman, Z. Cluster-Based Information Retrieval by using (K-means)- Hierarchical Parallel Genetic Algorithms Approach. (arXiv,2020)

[78] Bustamam, A., Tasman, H., Yuniarti, N., Frisca & Mursidah, I. Application of k-means clustering algorithm in grouping the DNA sequences of hepatitis B virus (HBV). *AIP Conference Proceedings*. **1862** pp. 030134 (2017,7)

[79] Tan, L. & Sun, J. K-means clustering based compression algorithm for the high-throughput DNA sequence. *2014 International Conference On Audio, Language And Image Processing*. pp. 952-955 (2014)

[80] Sathya, R. & Abraham, A. Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal Of Advanced Research In Artificial Intelligence*. **2** (2013,2)

[81] Park, S. & Park, Y. Photovoltaic power data analysis using hierarchical clustering. *2018 International Conference On Information Networking (ICOIN)*. pp. 727-731 (2018)

[82] Rousseeuw, P. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal Of Computational And Applied Mathematics*. **20** pp. 53-65 (1987)

[83] Kaufman, L. & Rousseeuw, P. Finding Groups in Data: An Introduction to Cluster Analysis. (Wiley,2009)

[84] Caliński, T. & J Harabasz A dendrite method for cluster analysis. *Communications In Statistics*. **3**, 1-27 (1974)

[85] Davies, D. & Bouldin, D. A Cluster Separation Measure. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. **PAMI-1**, 224-227 (1979)

[86] Thorndike, R. Who belongs in the family?. *Psychometrika*. **18** pp. 267-276 (1953)

[87] Macqueen, J. Some methods for classification and analysis of multivariate observations. *In 5-th Berkeley Symposium On Mathematical Statistics And Probability*. pp. 281-297 (1967)

[88] Shah, S. & Singh, M. Comparison of a Time Efficient Modified K-mean Algorithm with K-Mean and K-Medoid Algorithm. *2012 International Conference On Communication Systems And Network Technologies*. pp. 435-437 (2012)

[89] Dehariya, V., Shrivastava, S. & Jain, R. Clustering of Image Data Set Using K-Means and Fuzzy K-Means Algorithms. *2010 International Conference On Computational Intelligence And Communication Networks*. pp. 386-391 (2010)

[90] Joe H. Ward Jr. Hierarchical Grouping to Optimize an Objective Function. *Journal Of The American Statistical Association*. **58**, 236-244 (1963)

[91] Tomasini, R. & Van Wassenhove, L. From Preparedness to Partnerships: Case Study Research on Humanitarian Logistics. *International Transactions In Operational Research*. **16** pp. 549 - 559 (2009,9)

[92] Özdamar, L. & Demir, O. A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transportation Research Part E: Logistics And Transportation Review*. **48**, 591-602 (2012)

[93] Lee, K., Jo, J., Lee, B. & Kim, J. Design of tele-commanding system using asynchronous message service based on delay tolerant network for space communication. *2017 International Conference On Information And Communication Technology Convergence (ICTC)*. pp. 1253-1256 (2017)

[94] Farzad, A., Mashayekhi, H. & Hassanpour, H. A comparative performance analysis of different activation functions in LSTM networks for classification. *Neural Computing And Applications*. **31** (2019,7)

[95] Deng, Q., Faghanimakrani, T. & Aghvami, A. GBDT-Based Modules for Force Prediction in a Model-Mediated Teleoperation System. *2020 27th International Conference On Telecommunications (ICT)*. pp. 1-6 (2020)

[96] Deng, Q., Mahmoodi, T. & Aghvami, A. A Long-Short-Term Memory-Based Model for Kinesthetic Data Reduction. *IEEE Internet Of Things Journal*. pp. 1-1 (2023)

[97] Deng, Q., Mahmoodi, T., H. & Aghvami, A Bilateral Teleoperation System with Dimensionality Reduction and Data Reconstruction Techniques *IEEE Transactions on Haptics*. submitted on 08/05/2022