

Jae-young Lee · Shahram Payandeh

Haptic Teleoperation Systems

Signal Processing Perspective



Springer

Haptic Teleoperation Systems

Jae-young Lee · Shahram Payandeh

Haptic Teleoperation Systems

Signal Processing Perspective



Springer

Jae-young Lee
Electronics and Telecommunications
Research Institute (ETRI)
Deajeon
Republic of Korea

Shahram Payandeh
Simon Fraser University
Burnaby, BC
Canada

ISBN 978-3-319-19556-8
DOI 10.1007/978-3-319-19557-5

ISBN 978-3-319-19557-5 (eBook)

Library of Congress Control Number: 2015942543

Springer Cham Heidelberg New York Dordrecht London
© Springer International Publishing Switzerland 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Preface

This book explores haptic data processing methods for teleoperation systems, including prediction, compression, and error correction. In the proposed haptic data prediction method, unreliable network conditions, such as time-varying delay and packet loss, are detected by a transport layer protocol. Given the information from the transport layer, a Bayesian approach is introduced to predict position and force data in haptic teleoperation systems. Stability of the proposed method within stochastic formalism is presented based on the notion of passivity-based control. In the proposed haptic data compression method, compression techniques based on fixed rate downsampling are presented for efficient transmission over the network. Objective and psychophysical evaluations are conducted to demonstrate the compression performance of the proposed method and the adaptive downsampling method based on human perception. By presenting the two evaluation measures, the usefulness of the objective evaluation measure for haptic data is investigated. Finally, a forward error correction method is applied to haptic data over the unreliable network. Given the psychophysical evaluation results to determine the required number of bits for haptic data quantization, the error correction performance is presented under the additive noise and packet loss behavior.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Motivation	2
1.2.1	Haptic Data Transmission Over Unreliable Network	2
1.2.2	Haptic Data Compression	3
1.3	Related Work (Literature Review)	4
1.3.1	Haptic Teleoperation Systems Over Delayed Network	4
1.3.2	Haptic Data Compression	5
1.3.3	Haptic Data Error Correction	6
1.4	Layout of the Book	6
	References	8
2	A Bayesian Approach to Haptic Teleoperation Systems	11
2.1	State-Space Representations of Haptic Data Over the Network	12
2.1.1	Uplink Transmission	12
2.1.2	Downlink Transmission	16
2.2	A Bayesian Approach for Haptic Teleoperation Systems	17
2.2.1	Uplink Transmission	17
2.2.2	Downlink Transmission	19
2.3	Bayesian Particle Filter for Haptic Teleoperation Systems	19
2.3.1	Uplink Transmission	20
2.3.2	Downlink Transmission	22
2.4	Discussion	23
	References	23

3 Stability of Haptic Teleoperation Systems	
Based on the Bayesian Approach	25
3.1 Preliminary: Passivity of Haptic Teleoperation Systems.	26
3.2 Passivity Using Generalized Scattering Transformation	27
3.2.1 Generalized Scattering Transformation	27
3.2.2 Passivity of the Proposed Haptic Teleoperation Systems Using the Scattering Transformation.	28
3.3 Stability Using the Passivity Observer/Passivity Controller	31
3.4 Discussion	32
References	33
4 A Layered Protocol Architecture for Haptic Teleoperation Systems Based on the Bayesian Approach	35
4.1 Layered Protocol Architecture for Haptic Teleoperation Systems	35
4.2 Transport Layer	36
4.2.1 Existing Transport Protocols	37
4.2.2 Proposed Transport Protocol	38
4.3 Application Layer	40
4.3.1 Bayesian Particle Filter Based on M-UDP	41
4.3.2 Prediction-Based Bayesian Particle Filter	42
4.3.3 Low-Pass Filter	45
4.4 Experimental Study	46
4.4.1 Experimental Setup	46
4.4.2 Network Emulation	48
4.4.3 Practical Convergence of Bayesian Particle Filter	50
4.4.4 Experimental Results	51
4.5 Discussion	58
References	59
5 Haptic Data Compression	61
5.1 Haptic Data Compression Based on Downsampling	62
5.2 Adaptive Downsampling (Deadband Method)	63
5.3 Haptic Data Compression Based on Fixed Rate Downsampling	65
5.3.1 Fixed Rate DownSampling	65
5.3.2 Prediction Methods	65
5.3.3 Contact Force Detection	70
5.4 Experimental Study	71
5.4.1 Experimental Setup	71
5.4.2 Objective Evaluation	73
5.4.3 Psychophysical Evaluation	77

5.5	Discussion: Psychophysical Evaluation Results Represented in PSNR	82
	References	84
6	Haptic Data Digitization and Forward Error Correction	87
6.1	Forward Error Correction for Haptic Data	88
6.2	Haptic Data Digitization	89
6.3	Forward Error Correction for Haptic Data	90
6.3.1	Symbol Mapping	91
6.3.2	Reed–Solomon Codes	91
6.4	Forward Error Correction for Packet Loss Recovery	92
6.5	Experimental Study	94
6.5.1	Psychophysical Evaluation for Haptic Data Digitization	94
6.5.2	Performance Evaluation for Forward Error Correction . . .	97
6.6	Discussion	101
	References	101
7	Conclusion and Future Work	103
7.1	Conclusion	103
7.2	Future Work	104
	Appendix A: Basics of Bayesian Filtering	107
	Appendix B: A Bayesian Approach for Multivariable Teleoperation Systems	113
	Appendix C: Non-passive Conditions in the Presence of Network Delay	117
	Appendix D: Packet Structures of Standard Transport Protocols	121
	Appendix E: Performance of Low-Pass Filter	123
	Appendix F: Basics of Data Compression	125
	Appendix G: Implementation of Virtual Slave Manipulator	127
	Appendix H: Error Correcting Codes	129

Acronyms

ACK	Acknowledgement
ARQ	Automatic repeat query
ATSC	Advanced Television Systems Committee
AWGN	Additive white Gaussian noise
BCH	Bose and Ray-Chaudhuri
BER	Bit error rate
CDF	Cumulative distribution function
CDMA	Code division multiple access
DCT	Discrete cosine transform
DoF	Degree of freedom
DPCM	Differential pulse-code modulation
DVB	Digital video broadcasting
ETP	Efficient transport protocol
FEC	Forward error correction
FIR	Finite impulse response
FPGA	Field-programmable gate array
GoS	Group of samples
HCI	Human–computer interaction
HD API	Haptic device API (Application programming interface)
HL API	Haptic Library API
IP	Internet protocol
IPG	Interpacket gap
IPTV	Internet protocol television
IRTP	Interactive real-time protocol
LAN	Local area network
LPF	Low-pass filter
MAP	Maximum a posteriori
MMSE	Minimum mean-squared error
MSE	Mean-squared error
M-UDP	Modified UDP
NACK	Unacknowledgement

PD	Proportional derivative
PSNR	Peak signal-to-noise ratio
QoS	Quality of service
RS	Reed-Solomon
RTNP	Real-time network protocol
RTP	Real-time transport protocol
RTT	Round-trip time
SCARA	Selective Compliance Assembly Robot Arm
SFU	Simon Fraser University
SMC	Sequential Monte Carlo
SNR	Signal-to-noise ratio
TCP	Transport control protocol
UDP	User datagram protocol
WiMAX	Worldwide Interoperability for Microwave Access

Chapter 1

Introduction

The word *haptic* is originally from the Greek word which means pertaining to the sense of touch. *Haptic teleoperation* is the extension of person's ability to sense and manipulate into a remote environment. In this chapter, we introduce a haptic teleoperation system, its related benefits, and existing problems. Following that, the related literature is reviewed in Sect. 1.3. The contributions and layout of this thesis are highlighted in Sect. 1.4.

1.1 Overview

In a haptic teleoperation system, a *human operator* is allowed to interact with a *remote environment* using three main components: a *master manipulator*, a *slave manipulator*, and an interactive communication medium between the master and slave manipulators. The master manipulator is directly controlled by the human operator and provides position or velocity inputs to the slave manipulator. While the slave manipulator works with the remote environment, force data are generated and then fed back into the master manipulator to be perceived by the human operator. In the interactive communication medium, *haptic data*—position, velocity, and force data—are transmitted between the master and slave manipulators in real time, and hence, the human operator is able to interact with the remote environment.

Because of the recent growth of virtual reality technology, a *haptic interface* has received wide attention, as it also allows the human operator to become immersed in a virtual environment. Typical architectures of a haptic teleoperation system and a haptic interface are shown in Fig. 1.1. As shown in these figures, the slave manipulator and remote environment in Fig. 1.1 (a) can be replaced by the virtual slave manipulator and virtual environment in Fig. 1.1 (b), and therefore, the control, mechanical, and signal-processing methods for haptic teleoperation systems can also be applied to haptic interfaces in virtual environments.

Haptic teleoperation systems can be extended to many practical applications in distant, inaccessible, or dangerous environments, such as medical applications including telesurgery and minimally invasive surgery [1–4], remote ultrasound

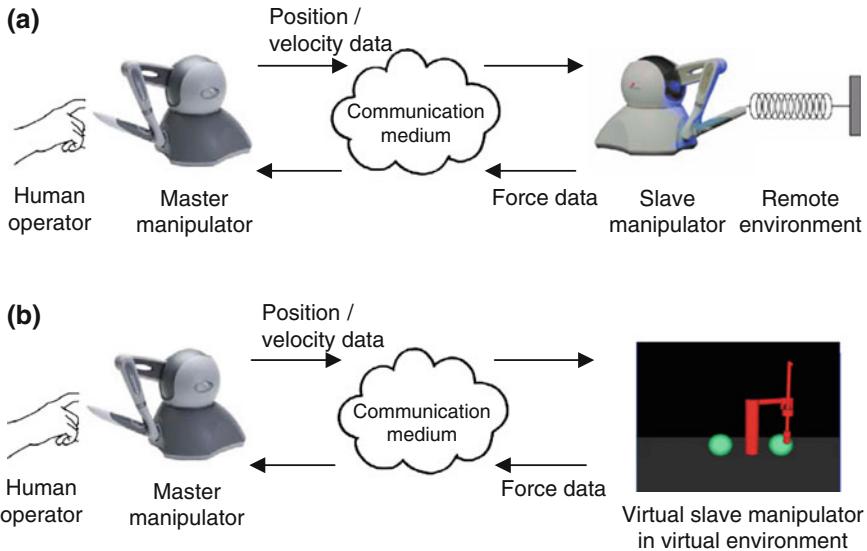


Fig. 1.1 **a** A haptic teleoperation system, and **b** a haptic interface in a virtual environment over a communication medium

controls [5], space and underwater operations [6], military tasks [7], and operations in nuclear environment [8]. While these haptic teleoperation systems are mostly designed for real environments, applications of haptic interfaces are developed to operate in artificially implemented environments, such as medical and flight training simulations [9], video game and entertainment [10], and education materials [11]. Some of these applications based on haptic teleoperation systems and haptic interfaces are shown in Fig. 1.2.

1.2 Motivation

1.2.1 Haptic Data Transmission Over Unreliable Network

In recent studies on haptic teleoperation systems, the Internet has been treated as an interactive communication medium between the master and the slave manipulators because of its availability, ease of deployment, and low cost [12]. However, the current Internet technology provides only a best-effort delivery service and does not guarantee the quality of service. Because packets are transmitted through multiple routers with different sizes of queues, packet delay through the network is generally unknown and is time varying. When the network is congested, packet delay may increase because some routers may need longer time to send out the congested packets from their queues. Furthermore, in such congested network, increasing packet traffic over limited bandwidth may overflow the queues in the routers, which

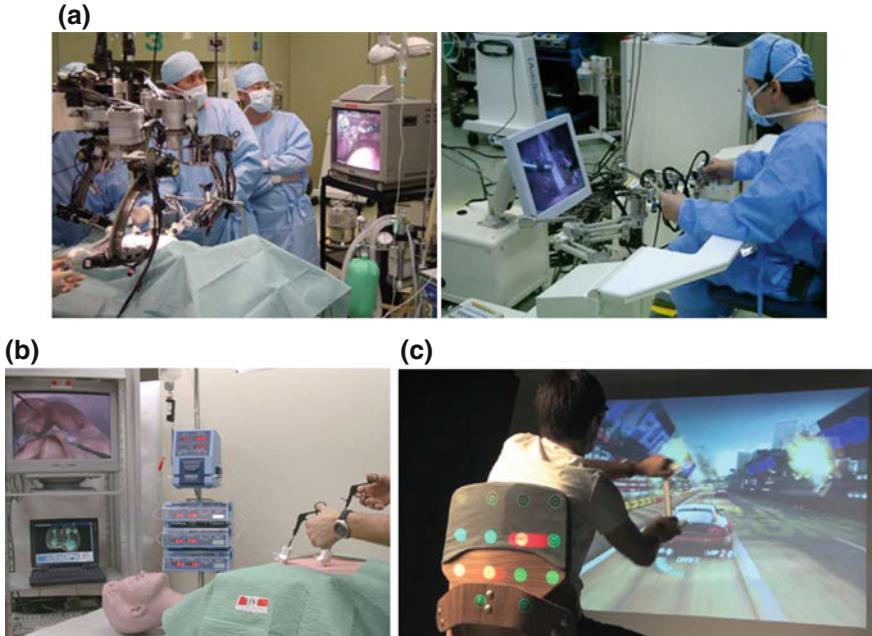


Fig. 1.2 Applications of haptic teleoperation systems: **a** A telesurgery trial conducted by University of Tokyo [1]. Applications of haptic interfaces: **b** A surgical training simulator developed by Swiss federal institute of technology [9], and **c** a haptic video game developed by Disney research [10]

eventually causes a packet loss problem. While these time-varying delay and packet loss problems are considered as the main drawbacks of the deployed Internet, it is also possible to observe other unreliable network conditions, such as out-of-ordered packets, packet corruption, and packet replication [13].

These unreliable network conditions not only degrade the performance of a haptic teleoperation system but also impair closed-loop *stability*, which is known as a fundamental requirement of haptic teleoperation systems [14]. In the area of control systems engineering, if a system is not stable, there may be random variations in the system even though a small signal is applied. Hence, an unstable teleoperation system is not operable in practice. Unreliable network conditions can also degrade *transparency*, which measures how the human operator perceives differences between the direct environment and the remote environment [15].

1.2.2 Haptic Data Compression

In recent applications of haptic teleoperation systems, haptic data are available in sampled forms to be processed through the network. Depending on the applications,

a certain minimum sampling rate is required to achieve realistic human perception and closed-loop stability [16]. In a general position—force architecture of a haptic teleoperation system, haptic data transmitted through the network are required to be sampled higher than 1 kHz. It has also been reported that significant higher sampling rates (e.g. 2–5 kHz) are required for certain applications involved with stiff environments in order to improve human perception [17]. Since haptic data sampled at such high rates can be multiple-degree of freedom (m-DoF) depending on the applications, the size of haptic data transmitted through the network can be large. However, the bandwidth of the deployed network is limited in general. Hence, when m-DoF haptic data with a high sampling rate are transmitted through unreliable network conditions, stability and transparency of overall teleoperation systems may be further impaired.

1.3 Related Work (Literature Review)

1.3.1 *Haptic Teleoperation Systems Over Delayed Network*

Haptic teleoperation systems over a delayed network have been mostly studied from the area of control systems engineering since Ferrell [14] presented that force feedback over a delayed network may introduce instability. The passivity-based control using the scattering transformation, which was first introduced by Anderson and Spong [18] and then extended by Niemeyer and Slotine [19], has been considered as one of the prominent control solutions for stability over the delayed network. In this approach, stability of an overall teleoperation system was theoretically achieved when arbitrary magnitude of constant delay was introduced between the master and the slave manipulators. The proportional derivative (PD)-based control method (Lee and Spong [20]) was also presented to address stability in the presence of known constant delay. Hannaford et al. [21] introduced the notion of time domain passivity for stability of haptic interfaces. This approach used the passivity observer to monitor energy balance in real time, and the passivity controller to dissipate any extra energy, so that stability could be guaranteed. Its extension to haptic teleoperation systems over the delayed network was also presented by Ryu et al. [22].

Since the recent growth of the Internet, unreliable network conditions, such as time-varying delay and packet loss, have been considered as the main challenges of haptic teleoperation systems. The control-based approaches, which were introduced in the presence of constant delay, have been extended in order to address such new challenges. The passivity-based control scattering transformation under the time-varying delay or packet loss was extensively studied by several researchers (e.g. Lozano et al. [24], Chopra et al. [25], and Secchi et al. [26]). Basanez et al. [27] studied the PD-based controller under the time-varying delay. Ryu et al. [23] also presented the passivity observer /passivity controller in order to address stability in the presence of time-varying delay.

Since unreliable network conditions have been considered to be random processes, signal-processing approaches, including simple signal reconstruction, interpolation, estimation, and prediction, have been employed in addition to the control architectures. Based on the scattering transformation, Niemeyer and Slotine [28] introduced a reconstruction filter in order to preserve stability in the presence of time-varying delay. Berestesky et al. [29] also used a buffer to alleviate the effect of time-varying delay and an interpolation method to reconstruct any lost data by introducing additional delay. Daly and Wang [30] presented a force estimation method based on the sliding mode control and showed its stability over the delayed network. They also presented the observer-based force estimation in the case of m-DoF nonlinear manipulations in [31]. Munir and Book [32] introduced the Smith predictor and the Kalman filter to predict transmitted scattering variables through the network. Mirfakhrai and Payandeh [47] introduced a delay prediction method over the network in order to minimize the errors of haptic data.

Besides the approaches from control systems engineering and signal processing, a few network protocol methods, which were based on the modifications from existing transport control protocol (TCP) or user datagram protocol (UDP), have been also introduced from the area of network protocols. Ping et al. [33] introduced a transport protocol based on UNIX platform in order to achieve faster transmission compared to Window-based platform. Uchimura et al. [34] presented a new configuration of packet headers in order to reduce the end-to-end delay between the master and the slave manipulators. Wirz et al. [35] introduced an efficient transport protocol modified from UDP. In this protocol, they used a gap between successive UDP packets in order to control a congested network and reduce the delay. Contributions such as real-time transport protocol that is also one of the standard protocols for real-time multimedia transmission (King et al. [36]), an adaptive buffer (Lee et al. [37]), and a wireless network-based protocol (Cen et al. [38]) have also been presented from the area of network protocols.

1.3.2 *Haptic Data Compression*

Haptic data compression has been mostly achieved by downsampling methods, which reduce the sampling rate from a sending manipulator and reconstruct the original sampling rate at a receiving manipulator in real time. Block-based methods, which may be prominently used for image or video compressions, were not significantly considered for haptic data compression because even a small processing delay introduced by such block-based methods may cause instability of an overall teleoperation system.

Shahabi et al. [39] introduced an early work on haptic data compression using downsampling methods. They also showed the use of differential pulse-code modulation, which can be achieved by similarities between consecutive haptic data. Otanez et al. [40] first introduced the deadband principle, which is based on adaptive downsampling, to network-based control systems for efficient data

transmission. Then, Hinterseer et al. [41] extended this deadband principle to a perception-oriented haptic data compression method. In this deadband principle, any changes in force data within a threshold may not be noticeable by human perception. Thereafter, the following contributions were mostly based on the deadband method. Hinterseer et al. [42] also presented a prediction scheme that is combined with the deadband method in order to achieve further data reduction. Kuschel et al. [43] used an interpolation scheme based on the deadband method, which introduced a processing delay. Because of the delay, they additionally used the passivity-based control in order to address stability. Zadeh et al. [44] conducted a psychophysical evaluation of the deadband method. In this work, they experimentally found absolute force thresholds for haptic data compression and also investigated the effect of velocity data on force perception.

1.3.3 Haptic Data Error Correction

The error correction of haptic teleoperation systems has not been significantly considered in the area of haptics yet. Cen et al. [45] presented a supermedia transport protocol, which was developed for real-time transmission of video, audio, haptic, and other types of human–computer interaction data. In this approach, since such real-time data were transmitted through multiple disjoint overlay paths, the protocol enabled transmission of duplicate packets, which provided an opportunity of error correction in the case of any packet loss. Lee et al. [46] also introduced a priority-based haptic event filtering when network problems such as packet loss occurred. This approach assigned priority to haptic data according to the network conditions when a haptic interface in a virtual environment was used.

1.4 Layout of the Book

In this book, we propose haptic data processing methods for teleoperation systems, including estimation¹/prediction,² compression, and error correction. Based on the existing control architecture, the proposed prediction/estimation and error correction methods are used to compensate for unreliable network conditions such as time-varying delay and packet loss. The proposed compression method focuses on transmitting haptic data efficiently within limited network bandwidth. The scope of this book, including haptic data processing and control architecture, is shown in

¹*Estimation* (also referred to as *filtering*) is an operation that involves the extraction of information about a quantity of interest at time k by using data measured up to and including time k [49].

²*Prediction* is an a priori form of estimation. It aims to derive information about what the quantity of interest will be at some future time $k + \tau$ ($\tau > 0$) by using data measured up to and including time k [49].

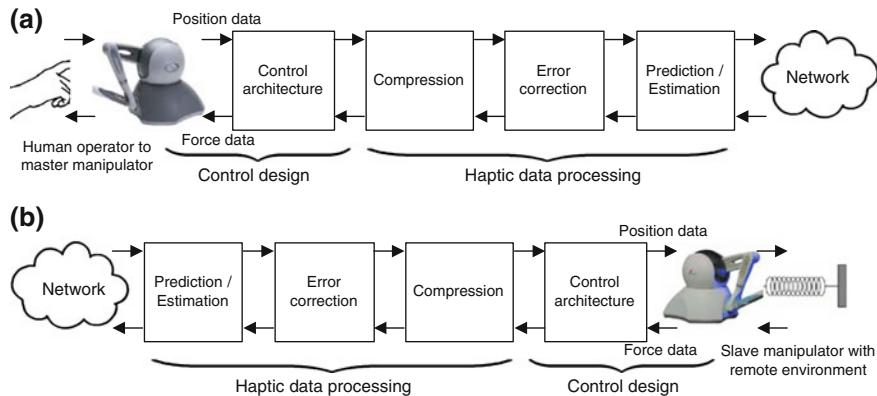


Fig. 1.3 The visualized scope of this book: **a** master side and **b** slave side. Based on the existing control architecture, compression, error correction, and prediction/estimation methods are presented for discrete haptic data through the network

Fig. 1.3. As shown in this figure, the proposed methods are based on a position–force architecture of haptic teleoperation system or haptic interface.

Some of the novel contributions of this book are as follows:

1. Haptic data prediction and estimation:

- Chapter 2—*A Bayesian approach to haptic teleoperation systems*: We present a stochastic representation of haptic data transmitted through the unreliable network. Based on the representation, we propose a Bayesian approach that tracks position and force data in order to compensate for unreliable network conditions.
- Chapter 3—*Stability of haptic teleoperation systems based on the Bayesian approach*: We present the notion of passivity-based control in order to address stability of the proposed haptic teleoperation system based on the Bayesian approach. Given the stochastic formalism introduced in Chap. 2, we present the existing control architectures, including the passivity-based scattering transformation and the passivity observer/passivity controller.
- Chapter 4—*A layered protocol architecture for haptic teleoperation systems based on the Bayesian approach*: We present a transport protocol that is able to detect unreliable network condition. Based on the information from the transport protocol, the proposed Bayesian approach performs either prediction or estimation in the application layer. An experimental study is provided using the master and slave haptic devices through the local network.

2. Haptic data compression:

- Chapter 5—We present haptic data compression methods mostly based on downsampling. The proposed compression method is based on fixed rate downsampling, and it is compared to the adaptive downsampling method

based on human perception. We conduct extensive experimental studies including objective and psychophysical evaluations for the haptic data compression methods. Given the two evaluations results, the objective evaluation measure is interpreted as the perceptual quality of haptic data, and its usefulness for haptic applications is discussed.

3 Haptic data error correction:

- Chapter 6—We first present quantization of haptic data in addition to a sampling process. Given fully digitized haptic data, we propose a forward error correction (FEC) method for haptic teleoperation systems over the unreliable network conditions. A psychophysical evaluation is conducted to determine the required number of bits for quantizing haptic data. Given the number of bits, we provide the error correction performance of the proposed FEC method under the additive noise and the packet loss behavior.

References

1. <http://www.nml.t.u-tokyo.ac.jp>
2. G. Guthart, J. Salisbury, The intuitive telesurgery system: overview and application, in *Proceedings of IEEE International Conference on Robotics Automations*, San Francisco, CA, USA, 2000, pp. 618–621
3. M. Tavakoli, R.V. Patel, M. Moallem, A. Aziminejad, *Haptic Interaction in Robotic Systems for Surgery and Telesurgery*, New Frontiers in Robotics Series (World Scientific Publishers, Singapore, 2008)
4. M. Tavakoli, R.V. Patel, M. Moallem, Design issues in a haptics-based master-slave system for minimally invasive surgery, in *IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004, pp. 371–376
5. P. Abolmaesumi, S.E. Salcudean, W.H. Zhu, M. Sirospour, S. DiMaio, Image-guided control of a robot for medical ultrasound. *IEEE Trans. Robot. Autom.* **18**, 11–23 (2002)
6. T. Sheridan, Space teleoperation through time delay: Review and prognosis. *IEEE Trans. Robot. Autom.* **9**(5), 592–606 (1993)
7. D.P Miller, Evaluation of vision systems for teleoperated land vehicles. *IEEE Control Syst. Mag.* **8**, 37–41 (1988)
8. K.A. Manocha, N. Pemalete, R. Dubey, Variable position mapping based assistance in teleoperation for nuclear cleanup, in *Proceedings of IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 374–379
9. <http://lsro.epfl.ch>
10. <http://www.disneyresearch.com>
11. P.B. Persson, M.D. Cooper, L.A.E. Tibell, S. Ainsworth, A. Ynnerman, B.H. Jonsson, Designing and evaluating a haptic system for biomolecular education, in *Proceedings of IEEE Virtual Reality*, Charlotte, NC, USA, Mar 2007, pp. 171–178
12. K. Kosuge, H. Murayama, T. Takeo, Bilateral feedback control of telemanipulators via computer network, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan, Nov 1996, pp. 1380–1385
13. V. Paxson, End-to-end Internet packet dynamics. *IEEE/ACM Trans. Netw.* **7**(3), 277–292 (1999)

14. W.R. Ferrell, Remote manipulation with transmission delay. *IEEE Trans. Hum. Factors Electron.* **6**(1), 24–32 (1965)
15. Y. Yokokohji, T. Yoshikawa, Bilateral control of master-slave manipulators for ideal kinesthetic coupling formulation and experiment. *IEEE Trans. Robot. Autom.* **10**(5), 605–619 (1994)
16. H.Z. Tan, M.A. Srinivasan, B. Eberman, B. Cheng, Human factors for the design of force-reflecting haptic interfaces, in *Proceedings of the 3rd Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1994, pp. 353–359
17. D. Kubus, I. Weidauer, F.M. Wahl, 1 kHz is not enough—how to achieve higher update rates with a bilateral teleoperation system based on commercial hardware, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robotics and Systems*, St. Louis, MO, Oct 2009, pp. 5107–5114
18. R.J. Anderson, M.W. Spong, Bilateral control of teleoperators with time delay. *IEEE Trans. Autom. Control*, **34**(5), 494–501 (1989)
19. G. Niemeyer, J.J.E. Slotine, Stable adaptive teleoperation. *IEEE J. Oceanic Eng.* **16**(1), 152–162 (1991)
20. D. Lee, M. Spong, Passive bilateral teleoperation with constant time delay. *IEEE Trans. Robot.* **22**(2), 269–281 (2006)
21. B. Hannaford, J.-H. Ryu, Time-domain passivity control of haptic interfaces. *IEEE Trans. Robot. Autom.* **18**, 1–10 (2002)
22. J.-H. Ryu, D.-S. Kwon, B. Hannaford, Stability guaranteed control: time domain passivity approach. *IEEE Trans. Control Syst. Technol.* **12**(6), 860–868 (2004)
23. J.-H. Ryu, J. Artigas, C. Preusche, A passive bilateral control scheme for a teleoperator with time-varying communication delay. *Mechatronics* **20**(7), 812–823 (2010)
24. R. Lozano, N. Chopra, M. Spong, Passivation of force reflecting bilateral teleoperators with time varying delay, in *Proceedings of 8th Mechatron. Forum*, Enschede, The Netherlands, 2002, pp. 954–962
25. N. Chopra, P. Berestesky, M.W. Spong, Bilateral teleoperation over unreliable communication networks. *IEEE Trans. Control Syst. Technol.* **16**(2), 304–313 (2008)
26. C. Secchi, S. Stramigioli, C. Fantuzzi, Digital passive geometric telemanipulation, in *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sept 2003, pp. 3290–3295
27. E. Nuno, L. Basanez, R. Ortega, M. Spong, Position tracking for non-linearteleoperators with variable time delay. *Int. J. Robot. Res.* **28**(7), 895–910 (2009)
28. G. Niemeyer, J.J.E. Slotine, Towards force-reflecting teleoperation over the Internet, in *Proceedings IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 1909–1915
29. P. Berestesky, N. Chopra, M.W. Spong, Discrete time passivity in bilateral teleoperation over the Internet, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr 2004, pp. 4557–4564
30. J.M. Daly, D.W.L. Wang, Bilateral teleoperation using unknown input observers for force estimation, in *Proceedings of American Control Conference*, St. Louis, MO, USA, June 2009, pp. 89–95
31. J.M. Daly, D.W.L. Wang, Time-delayed bilateral teleoperation with force estimation for n-DOF nonlinear manipulators, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 3911–3918
32. S. Munir, W. Book, Internet-based teleoperation using wave variables with prediction. *IEEE/ASME Trans. Mechatron.* **7**(2), 124–133 (2002)
33. L. Ping, L. Wenjuan, S. Zengqi, Transport layer protocol reconfiguration for network-based robot control system, in *Proceedings of IEEE International Conference on Networking, Sensing, and Control*, Tucson, AZ, Mar 2005, pp. 1049–1053
34. Y. Uchimura, T. Yakoh, Bilateral robot system on the real-time network structure. *IEEE Trans. Ind. Electron.* **51**(5), 940–946 (2004)

35. R. Wirz, M. Ferre, R. Marín, J. Barrio, J. Claver, J. Ortego, Efficient transport protocol for networked haptics applications, in *Proceedings of The 6th International Conference on Haptics: Perception, Devices and Scenarios*, vol. 5024, Madrid, Spain, June 2008, pp. 3–12
36. H.H. King, B. Hannaford, J. Kammerl, E. Steinbach, Establishing multimodal telepresence sessions using the session initiation protocol and advanced haptic codecs, in *Proceedings of IEEE Haptics Symposium*, Waltham, MA, USA, Mar 2010, pp. 321–325
37. S. Lee, S. Moon, J.W. Kim, A network-adaptive transport scheme for haptic-based collaborative virtual environments, in *Proceedings of ACM SIGCOMM Workshop on Network and Systems: Support for Games*, New York, NY, USA, 2006, 8 pp.
38. Z. Cen, M.W. Mutka, D. Zhu, N. Xi, Improved transport service for remote sensing and control over wireless networks, in *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*, Washington DC, USA, Nov. 2005, 8 pp.
39. C. Shahabi, A. Ortega, M.R. Kolahdouzan, A comparison of different haptic compression techniques, in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2002, pp. 657–660
40. P.G. Otanez, J.R. Moyne, D.M. Tilbury, Using deadbands to reduce communication in networked control systems, in *Proceedings of American Control Conference*, Anchorage, AK, 2002, pp. 3015–3020
41. P. Hinterseer, E. Steinbach, S. Hirche, M. Buss, A novel, psychophysically motivated transmission approach for haptic data streams in telepresence and teleaction systems, in *Proceedings of International Conference Acoustics, Speech, and Signal Processing*, vol. 2, Philadelphia, PA, Mar 2005, pp. 1097–1100
42. P. Hinterseer, E. Steinbach, Model-based data compression for 3D virtual haptic teleinteraction, in *Proceedings of IEEE Int. Conference on Consumer Electronics*, Las Vegas, NV, Jan 2006, pp. 23–24
43. M. Kuschel, P. Kremer, M. Buss, Passive haptic data-compression methods with perceptual coding for bilateral presence systems. *IEEE Trans. Syst. Man Cybern.* **39**(6), 1142–1151 (2009)
44. M.H. Zadeh, D.W.L. Wang, E. Kubica, Perception-based lossy haptic compression considerations for velocity-based interactions, in *Multimedia Systems*, vol. 13, pp. 275–282, (Springer, Berlin, 2007)
45. Z. Cen, M.W. Mutka, D. Zhu, N. Xi, Supermedia transport for teleoperations over overlay networks, in *Proceedings of Networking*, Waterloo, ON, Canada, 2005, pp. 1409–1412
46. S. Lee, J.W. Kim, Priority-based haptic event filtering for transmission and error control in networked virtual environments. *Multimedia Syst.* **15**(6), 355–367 (2009)
47. T. Mirfakhrai, S. Payandeh, On using delay prediction in controlling force reflecting teleoperation over the Internet. *Robotica* **23**(6), 809–813 (2005)

Chapter 2

A Bayesian Approach to Haptic Teleoperation Systems

Nomenclatures

t	Continuous time at the master manipulator
t_s	Continuous time at the slave manipulator
k	Discrete time of master clock
k_s	Discrete time of slave clock
k_d	Delayed time of master clock
k_{ds}	Delayed time of slave clock
n_m	Delay variable from the master
n_s	Delay variable from the slave
\mathbf{x}_m	Master position
\mathbf{x}_s	Position received at the slave
$\hat{\mathbf{x}}_m$	Estimated/predicted master position at the slave
\mathbf{f}_s	Slave force
\mathbf{f}_m	Force received at the slave
$\hat{\mathbf{f}}_m$	Estimated/predicted slave force at the master
w_m	Importance weight of the master
\tilde{w}_m	Normalized importance weight of the master
w_s	Importance weight of the slave
\tilde{w}_s	Normalized importance weight of the slave
N	Number of particles
q	Importance distribution
$\mathbf{g}_1, \mathbf{g}_2$	State transition functions
$\mathbf{h}_1, \mathbf{h}_2$	Observation transition functions
$\mathbf{l}_1, \mathbf{l}_2$	State noises
$\mathbf{r}_1, \mathbf{r}_2$	Observation noises

Unreliable network behaviors, such as time-varying delay, packet loss, and other path characteristics, are in general described by random processes. In the real network environment, delay jitter is observed since packets are transmitted through multiple intermediate nodes with random queuing delays. The delay increases randomly when packets are congested over the network with limited bandwidth.

Consequently, such congested network causes packet loss, which is also a random behavior. In this chapter, based on the stochastic nature of the network, we present a stochastic representation for haptic data, which may be time-varying delayed or lost over the network. A Bayesian approach,¹ which models uncertainty as stochastic variables, is proposed to estimate or predict haptic position and force data through unreliable network conditions. We also propose a practical solution of the Bayesian approach, which is known as a robust tracking method for nonlinear problems.

This chapter is organized as follows. In Sect. 2.1, we present a state-space representation of haptic data over the unreliable network in order to perform the stochastic estimation or prediction. Based on the state-space representation, a Bayesian approach that finds the optimal estimates of position and force data is presented in Sect. 2.2. As a practical solution, the Bayesian particle filter is proposed in Sect. 2.3 in order to estimate or predict position and force data over unreliable network conditions.

2.1 State-Space Representations of Haptic Data Over the Network

A haptic teleoperation system or a haptic interface that exchanges position and force data over the network is shown in Fig. 2.1. In the *uplink transmission* of the network, position data from a master manipulator are transmitted to a slave manipulator through the network. Based on the movements of the master manipulator, reflected force data are generated by any contact of the slave manipulator with the remote environment, and then they are fed into the master manipulator in the *downlink transmission*. The master and slave manipulators, which are interfaced with the human operator and remote environment, respectively, are continuous systems, whereas position and force data transmitted through the network are sampled data in discrete time. When the discrete position and force data are transmitted over unreliable network conditions, such as time-varying delay and packet loss, they can be formulated in nonlinear state-space models to perform the proposed stochastic estimation or prediction.

2.1.1 Uplink Transmission

In the uplink transmission of Fig. 2.1, $\mathbf{x}_m[k]$, which is sampled from a continuous movement $\mathbf{x}_h[t]$, is multiple-degree of freedom (m-DoF) true position vector in a discrete time k , where k represents a time instance of local clock at the master manipulator. The master position $\mathbf{x}_m[k]$ is transmitted through the network, and the

¹Basics of the Bayesian statistics are presented in Appendix A.

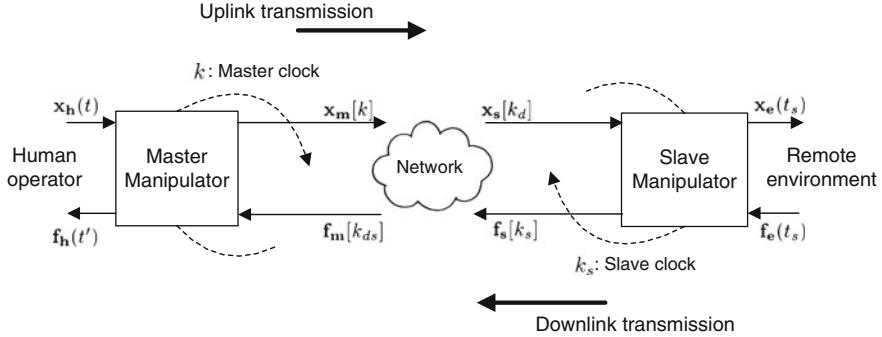


Fig. 2.1 A haptic teleoperation system that exchanges position and force data over the network

received position at the slave manipulator is represented as $\mathbf{x}_s[k_d]$, where k_d is a delayed time instance from the master manipulator. Since the network delay varies over time, k_d is given by

$$k_d = k + n_m[k] \quad (2.1)$$

where $n_m[k]$ represents a time-varying delay in the uplink transmission, and it can be expressed as a function of time k . Note that when the network delay is constant, $n_m[k]$ is not necessarily a function of time k , so (2.1) becomes $k_d = k + n_m$. Hence, the delayed position at the slave manipulator is $\mathbf{x}_s[k + n_m] = \mathbf{x}_m[k]$, which can be a deterministic signal rather than a stochastic variable. In order to perform the stochastic estimation or prediction, we first define a state-space model of position data flow through the network. The state vector is given by

$$\mathbf{x}_m[k] = \mathbf{g}_1(\mathbf{x}_m[k - 1] + \mathbf{l}_1[k - 1]) \quad (2.2)$$

where $\mathbf{x}_m[k]$ and $\mathbf{x}_m[k - 1]$ are the current and previous states, respectively. $\mathbf{l}_1[k - 1]$ is the state noise, which may be a nonnormal² distribution. $\mathbf{g}_1(\cdot)$ is the nonlinear state transition function, which represents the human movement from the master manipulator. In the stochastic estimation or prediction problem, the objective is to find the state using *observations*.³ The observation vector is given by

$$\mathbf{x}_s[k_d] = \mathbf{h}_1(\mathbf{x}_m[k] + \mathbf{r}_1[k_d]) \quad (2.3)$$

where $\mathbf{x}_s[k_d]$ is the observation vector, $\mathbf{r}_1[k_d]$ is the observation noise, and $\mathbf{h}_1(\cdot)$ is the nonlinear observation function. The state and observation processes in the uplink transmission are visualized in Fig. 2.2. In this figure, the master positions $\{\mathbf{x}_m[k - 1], \mathbf{x}_m[k], \mathbf{x}_m[k + 1]\}$ are the hidden states, which are not directly

²A random variable with a Gaussian distribution is said to be normally distributed.

³An observation is also called a measurement.

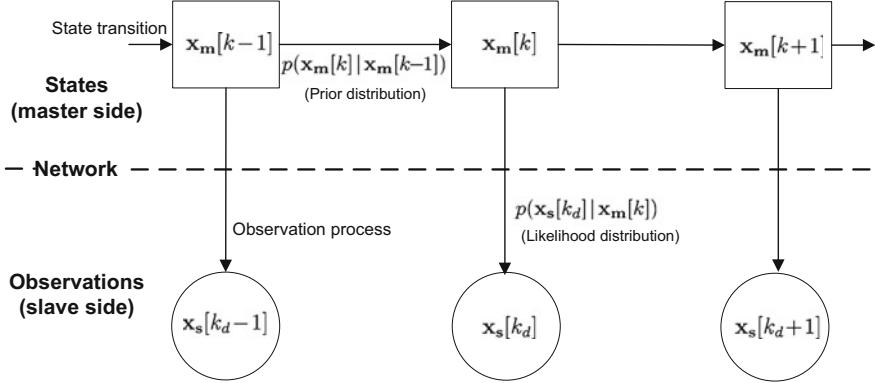


Fig. 2.2 State and observation processes of position data in the uplink transmission. The current state at the master manipulator depends on the previous state, and the current observation depends on the current state

observable from the slave manipulator. The current state depends only on the previous state⁴ so that a state transition in (2.2) is described by a *prior* distribution, which is $p(\mathbf{x}_m[k]|\mathbf{x}_m[k-1])$. The received positions $\{\mathbf{x}_s[k_d-1], \mathbf{x}_s[k_d], \mathbf{x}_s[k_d+1]\}$ are the observations at the slave manipulator, which depend on the states. This observation process in (2.3) is described by a *likelihood* distribution, which is $p(\mathbf{x}_s[k_d]|\mathbf{x}_m[k])$.⁵ Note that since this position data flow is modeled based on a Markov process with unobserved hidden states, such stochastic model is also called a *hidden Markov model* [2].

When the delayed position data are received at the slave manipulator, they need to be represented by time instances of local clock at the slave manipulator, which are $\{k_s - 1, k_s, k_s + 1\}$. Assuming that the clocks of master and slave manipulators are synchronized, the representations of delayed position data at the slave manipulator are shown in Fig. 2.3. In the presence of constant delay, which is shown in Fig. 2.3a, we simply have $k_s = k_d$ because k_d is not actually varying. Note that k_{d_0} is the delay of initial master position, which is given by $k_{d_0}k_{d_0} = n_m[0]$. When the time-varying delay is introduced, which is shown in Fig. 2.3b, the sample $\mathbf{x}_m[k + 1]$ is further delayed in this example. While the slave clock runs at its sampling rate (e.g. 1 kHz), the slave manipulator holds the last received sample $\mathbf{x}_s[k_d]$ so that $\mathbf{x}_s[k_s + 1] = \mathbf{x}_s[k_d]$ in this case. Therefore, in the presence of time-varying delay, we can have $k_s \neq k_d$. For the last case, the packet loss behavior is illustrated in Fig. 2.3c. In this example, the sample $\mathbf{x}_s[k_d]$ is lost during the network transmission.

⁴This is a first-order Markov process. The current state does not depend on previous trajectory, but only the previous state. i.e. $p(\mathbf{x}_m[k]|\mathbf{x}_m[k-1], \mathbf{x}_m[k-2], \dots, \mathbf{x}_m[0]) = p(\mathbf{x}_m[k]|\mathbf{x}_m[k-1])$.

⁵ $p(\mathbf{x}_m[k]|\mathbf{x}_m[k-1])$ describes the distribution of the current state conditioned on the previous state whereas $p(\mathbf{x}_s[k_d]|\mathbf{x}_m[k])$ describes the distribution of the current observation conditioned on the current state.

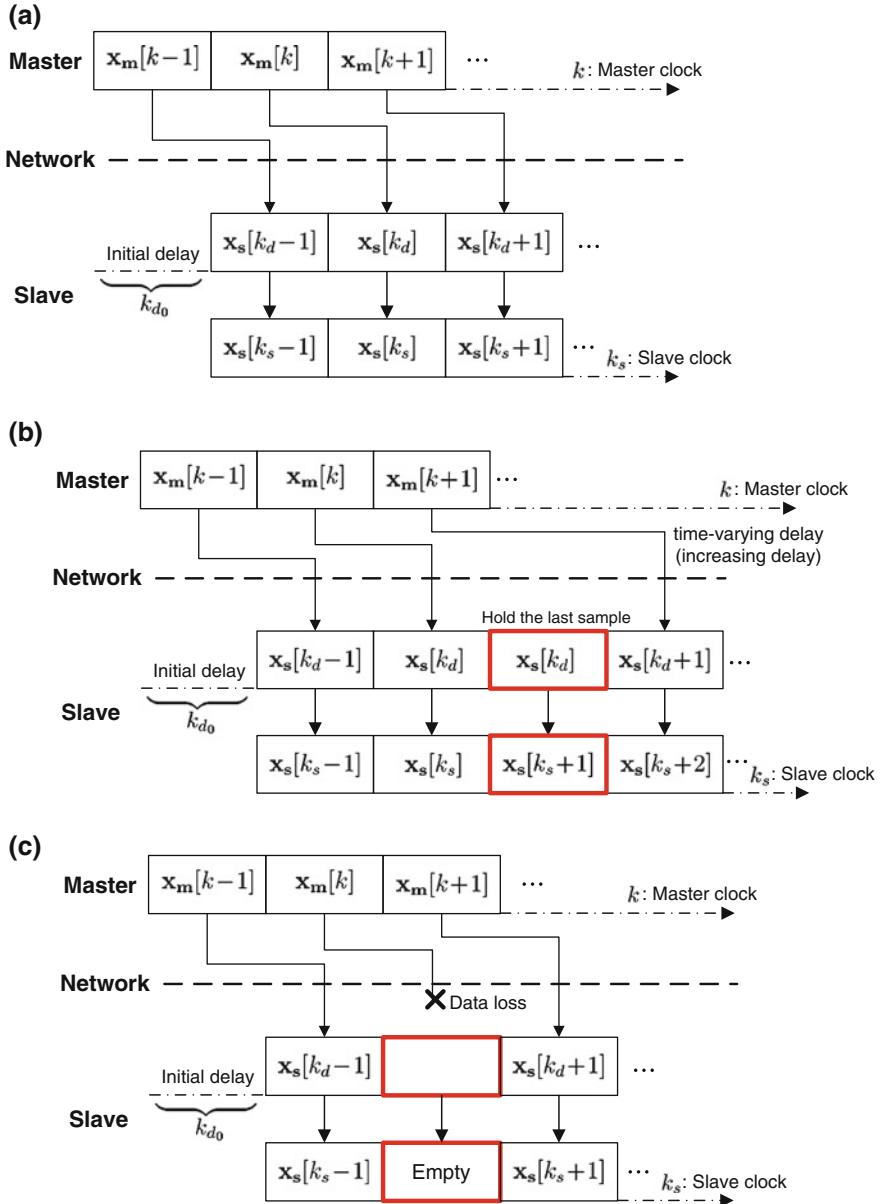


Fig. 2.3 Representations of delayed position data at the slave manipulator. **a** Constant delay case. **b** Time-varying delay (increasing delay) case. In this example, the slave manipulator holds the last received sample. **c** Packet loss case. In this example, the slave manipulator does not hold the last received sample, so that $x_s[k_s] = 0$

While the slave clock runs at its sampling rate, the slave manipulator uses the strategy that the unreceived sample becomes null data, which is $\mathbf{x}_s[k_s] = 0$. Note that in this packet loss case, it is also possible to choose the strategy used in Fig. 2.3b so that $\mathbf{x}_s[k_s] = \mathbf{x}_s[k_d - 1]$, instead of $\mathbf{x}_s[k_s] = 0$.

2.1.2 Downlink Transmission

A force data flow in the downlink transmission of the network is shown in Fig. 2.1. Once a slave position is obtained at the discrete time instance k_s , the slave manipulator interacts with the remote environment by converting to a continuous position $\mathbf{x}_e[t_s]$. When position and force data are transmitted with an equal sampling rate (e.g. 1 kHz), a continuous force $\mathbf{f}_e[t_s]$ is generated by any contact with the remote environment and is sampled as $\mathbf{f}_s[k_s]$ for transmitting over the network. The received force at the master manipulator, which is delayed over the network, is $\mathbf{f}_m[k_{ds}]$, and the delayed time instance $[k_{ds}]$ is given by

$$k_{ds} = k_s + n_s[k_s]. \quad (2.4)$$

$n_s[k_s]$ represents a time-varying delay in the downlink transmission, which is expressed as a function of time k_s . When $n_s[k_s]$ is a constant delay, $n_s[k_s]$ is not varying over time so that (2.4) becomes $k_{ds} = k_s + n_s$. Hence, the delayed force at the master manipulator is $\mathbf{f}_m[k_s + n_s] = \mathbf{f}_s[k_s]$, which can be a deterministic signal rather than a stochastic variable. In order to perform the proposed stochastic estimation or prediction, we first define a state-space model of force data flow in the downlink transmission. The state vector is given by

$$\mathbf{f}_s[k_s] = \mathbf{g}_2(\mathbf{f}_s[k_s - 1] + \mathbf{l}_2[k_s - 1]) \quad (2.5)$$

where $\mathbf{f}_s[k_s]$ and $\mathbf{f}_s[k_s - 1]$ are the current and previous states, respectively. $\mathbf{l}_2[k_s - 1]$ is the state noise that may be a nonnormal distribution. $\mathbf{g}_2(\cdot)$ is the nonlinear state transition function, which represents the force trajectory from the slave manipulator. The observation vector is given by

$$\mathbf{f}_m[k_{ds}] = \mathbf{h}_2(\mathbf{f}_s[k_s] + \mathbf{r}_2[k_{ds}]) \quad (2.6)$$

where $\mathbf{f}_m[k_{ds}]$ is the observation vector, $\mathbf{r}_2[k_{ds}]$ is the observation noise, and $\mathbf{h}_2(\cdot)$ is the nonlinear observation function. As illustrated for the state and observation processes of position data in Fig. 2.2, the slave force $\mathbf{f}_s[k_s]$ is the hidden state and is not directly observable from the master manipulator. The current state only depends on the previous state, which is described by $p(\mathbf{f}_s[k_s]|\mathbf{f}_s[k_s - 1])$. The observation vector $\mathbf{f}_m[k_{ds}]$, which is the received force at the master manipulator, depends on the state, described by $p(\mathbf{f}_m[k_{ds}]|\mathbf{f}_s[k_s])$.

In order to represent the delayed force in terms of the time instance of master clock, we use the strategies shown in Fig. 2.3. In the presence of constant delay, because k_{ds} is not varying, we simply have $k' = k_{ds}$. Note that k' is a delayed time instance of the master clock, which may be expressed as $k' = k + k_{ds_0}$, and k_{ds_0} is the delay of initial slave force such as $k_{ds_0} = n_s[0]$. In the presence of time-varying delay, we can have $k' \neq k_{ds}$ as shown in the example of Fig. 2.3b. When the packet loss occurs, as presented in Fig. 2.3c, the unreceived force sample becomes null data, which becomes zero.

2.2 A Bayesian Approach for Haptic Teleoperation Systems

Given the state-space models for the uplink transmission (2.2) and (2.3), and the downlink transmission (2.5) and (2.6), the objective is to find the states by using the stochastic estimation or prediction. In this section, a Bayesian approach is considered, in which the estimation or prediction is recursively performed by computing the states given available observations. A haptic teleoperation system based on the estimation or prediction method is shown in Fig. 2.4. In this figure, the estimates of the states in the uplink and downlink transmissions are denoted as $\hat{\mathbf{x}}_m$ and $\hat{\mathbf{f}}_s$, respectively.

2.2.1 Uplink Transmission

In a Bayesian approach, the estimate of position data $\hat{\mathbf{x}}_m[k_s]$ can be obtained using the previous state $\mathbf{x}_m[k_s - 1]$ and available observation $\mathbf{x}_s[k_{d_0}; k_d]$. Note that $\mathbf{x}_s[k_{d_0}; k_d]$ represents the received positions at the slave manipulator from the initial time k_{d_0} to k_d . The process to find the state $\hat{\mathbf{x}}_m[k_s]$ based on the state-space model of the uplink transmission is shown in Fig. 2.5.

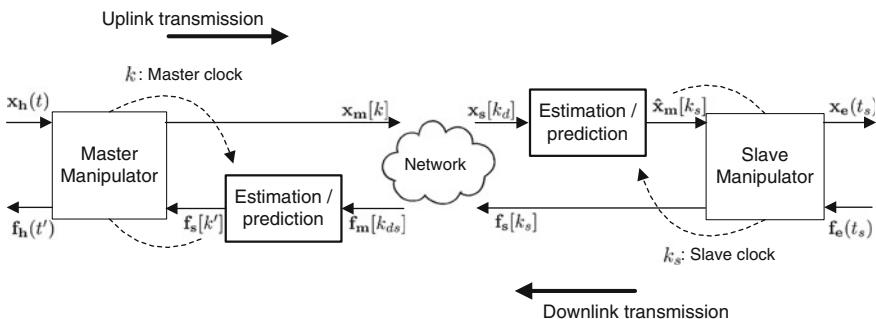


Fig. 2.4 A haptic teleoperation system based on the prediction or estimation to compensate for unreliable network conditions, such as time-varying delay or packet loss

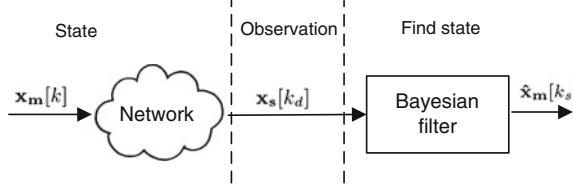


Fig. 2.5 A Bayesian approach to find the state, given the observation in the uplink transmission

In this Bayesian approach, since the state is computed given the observations, it is a problem to construct a *posterior* distribution, which is $p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d])$. A posterior distribution is a distribution of an unknown quantity, which is conditioned on an observation. The optimal solution that finds the state using the posterior distribution is given by [3, 7]

$$\mathbb{E}(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d]) = \int \mathbf{x}_m[k_s] p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d]) d\mathbf{x}_m \quad (2.7)$$

where $\mathbb{E}(\cdot)$ denotes the expectation. The posterior distribution in (2.7) is recursively computed by using the following two steps [3, 7]:

$$p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d-1]) = \int p(\mathbf{x}_m[k_s]|\mathbf{x}_m[k_s-1]) p(\mathbf{x}_m[k_s-1]|\mathbf{x}_s[k_{d_0}:k_d-1]) d\mathbf{x}_m \quad (2.8)$$

$$p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d]) = \frac{p(\mathbf{x}_s[k_d]|\mathbf{x}_m[k_s]) p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{d_0}:k_d-1])}{p(\mathbf{x}_s[k_d]|\mathbf{x}_s[k_{d_0}:k_d-1])} \quad (2.9)$$

Equations (2.8) and (2.9) are called the *prediction* and *update* steps, respectively. The prediction step (2.8) is evolved from the posterior distribution at the previous time instance, which is $p(\mathbf{x}_m[k_s-1]|\mathbf{x}_s[k_{d_0}:k_d-1])$. Then, the update step (2.9) is evolved from the prediction step (2.8), given the observation at the current time instance k_d .

In general, the computations of (2.8) and (2.9) are not directly solvable. Hence, Bayesian filters, such as the Kalman filter and the Bayesian particle filter, are used in real-world applications for the practical computation of the posterior distribution. In the case of Kalman filter, which sequentially deduces the optimal solution, we should assume that the state noise $\mathbf{l}_1[k]$ in (2.2) and observation noise $\mathbf{r}_1[k_d]$ in (2.3) have normal distributions. Furthermore, the state and observation transition functions $\mathbf{g}_1(\cdot)$ and $\mathbf{h}_1(\cdot)$ need to be assumed as linear equations. Given these assumptions, the estimates of position and force data can be obtained by sequentially deducing the distributions in (2.8) and (2.9) of the optimal solution [4].

When the transition functions are nonlinear or the noise sequences are not normally distributed, which may be observed more frequently in real-world

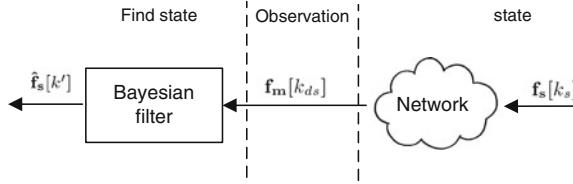


Fig. 2.6 A Bayesian approach to find the state, given the observation in the downlink transmission

applications, the Bayesian particle filter provides a robust tracking performance [5]. In this thesis, we focus our attention on the Bayesian particle filter to estimate or predict position and force data. A haptic teleoperation system using the proposed Bayesian particle filter is presented in Sect. 2.3.

2.2.2 Downlink Transmission

In the downlink transmission, the estimate of force data $\hat{\mathbf{f}}_s[k']$ can be obtained using the previous state $\mathbf{f}_s[k' - 1]$ and available observation $\mathbf{f}_m[k_{ds_0} : k_{ds}]$, where $\mathbf{f}_m[k_{ds_0} : k_{ds}]$ represents the received forces at the master manipulator from the initial time k_{ds_0} to k_{ds} . The process to find the state $\hat{\mathbf{f}}_s[k']$ based on the state-space model of the downlink transmission is shown in Fig. 2.6.

Similar to the uplink transmission, the optimal solution of force data based on the Bayesian approach is given by

$$\mathbb{E}(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds}]) = \int \mathbf{f}_s[k'] p(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds}]) d\mathbf{f}_s. \quad (2.10)$$

This optimal solution (2.10) can be obtained by recursively computing the prediction step $p(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds} - 1])$ and the update step $p(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds}])$, as presented in (2.8) and (2.9). Since we assume that the transition functions are nonlinear or noises are not necessarily normal distributions, such as (2.5) and (2.6), we focus our attention on the Bayesian particle filter for tracking force data in the downlink transmission.

2.3 Bayesian Particle Filter for Haptic Teleoperation Systems

In the Bayesian particle filter, the required posterior distributions in (2.7) and (2.10) are represented by a set of random samples with associated weights. This is a sequential Monte Carlo method that requires a number of random samples in order

to obtain the estimated or predicted states. As a number of samples increases, this Monte Carlo characteristic represents the required posterior distribution, and therefore, the Bayesian particle filter approaches to the optimal solution.

2.3.1 Uplink Transmission

Using the Bayesian particle filter, the posterior distribution in the uplink transmission (2.9) is empirically approximated by drawing a set of randomly chosen weighted samples such that [5]

$$\begin{aligned} p(\mathbf{x}_m[k_s] | \mathbf{x}_s[k_{d_0}; k_d]) &\approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_m^i[k_s] \delta(\mathbf{x}_m[k_{s_0}; k_s] - \mathbf{x}_m^i[k_{s_0}; k_s]) \\ &\triangleq \hat{p}(\mathbf{x}_m[k_s] | \mathbf{x}_s[k_{d_0}; k_d]) \end{aligned} \quad (2.11)$$

where $\delta(\cdot)$ is the Dirac delta function, N is the number of particles, and $\hat{p}(\cdot)$ is the approximation of posterior distribution. A particle refers a sample in order to represent a set of states according to probability distributions. In (2.11), the posterior distribution can be represented by the set of random samples $\{\mathbf{x}_m^i[k_{s_0}; k_s]\}_{i=1}^N$ with associated weights $\{w_m^i[k_s]\}_{i=1}^N$, where time instance k_{s_0} denotes the initial time instance of slave local clock. The weight $w_m^i[k_s]$ is called the *importance weight*, which is given by

$$w_m^i[k_s] = \frac{p(\mathbf{x}_m^i[k_{s_0}; k_s] | \mathbf{x}_s[k_{d_0}; k_d])}{q(\mathbf{x}_m^i[k_{s_0}; k_s] | \mathbf{x}_s[k_{d_0}; k_d])} \quad (2.12)$$

where $q(\cdot)$ in the denominator is called an importance distribution, and $\tilde{w}_m^i[k_s]$ in (2.11) is a normalized importance weight given by

$$\tilde{w}_m^i[k_s] = w_m^i[k_s] / \sum_{i=1}^N w_m^i[k_s]. \quad (2.13)$$

Computing the importance weight (2.12) involves the whole previous trajectory of state, which may require high computational complexity. Hence, we reformulate (2.12) in a recursive form by factorizing⁶ $q(\cdot)$ such that

$$\begin{aligned} q(\mathbf{x}_m[k_{s_0}; k_s] | \mathbf{x}_s[k_{d_0}; k_d]) &= q(\mathbf{x}_m[k_s] | \mathbf{x}_m[k_s - 1], \mathbf{x}_s[k_{d_0}; k_d]) \\ &\quad \cdot q(\mathbf{x}_m[k_{s_0}; k_s - 1] | \mathbf{x}_s[k_{d_0}; k_d - 1]) \end{aligned} \quad (2.14)$$

⁶Factorization of joint distribution is given by $p(a, b | c) = p(a | b, c)p(b | c)$.

The numerator in (2.12) can be also written in a factorized form that allows computing the posterior distribution sequentially. Based on these factorized forms, a new recursive and time-update form of the importance weight can be computed as

$$w_m^i[k_s] = \tilde{w}_m^i[k_s - 1] \frac{p(\mathbf{x}_s[k_d] | \mathbf{x}_m^i[k_s]) p(\mathbf{x}_m^i[k_s] | \mathbf{x}_m^i[k_s - 1])}{q(\mathbf{x}_m^i[k_s] | \mathbf{x}_m^i[k_s - 1]), \mathbf{x}_s[k_d]}. \quad (2.15)$$

This importance weight is associated with the state $\mathbf{x}_m^i[k_s]$ at time k_s for i th particle. After the factorization, the previous trajectory of state is not necessarily computed. Hence, the posterior distribution in (2.11) can be simplified as

$$p(\mathbf{x}_m[k_s] | \mathbf{x}_s[k_{d_0}; k_d]) \approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_m^i[k_s] \delta(\mathbf{x}_m[k_s] - \mathbf{x}_m^i[k_s]) \quad (2.16)$$

where the importance weight is (2.15). The recursive computation (2.15) depends on the choice of the importance distribution $q(\cdot)$, which is the denominator of (2.15). One of the most popular choices is to employ a prior distribution such as

$$q(\mathbf{x}_m^i[k_s] | \mathbf{x}_m^i[k_s - 1], \mathbf{x}_s[k_d]) = p(\mathbf{x}_m^i[k_s] | \mathbf{x}_m^i[k_s - 1]). \quad (2.17)$$

Substituting (2.17) into (2.15), Eq. (2.15) is simplified as

$$w_m^i[k_s] = \tilde{w}_m^i[k_s - 1] p(\mathbf{x}_s[k_d] | \mathbf{x}_m^i[k_s]). \quad (2.18)$$

Using (2.18), implementing the proposed Bayesian particle filter can be more easily achieved, since only the likelihood distribution in (2.18) needs to be computed in order to compute the approximated posterior distribution (2.16).

One of the issues using the Bayesian particle filter in practice is that the recursive computation (2.16) may diverge as iterations proceed. In order to avoid such divergence problem, the *resampling* step needs to be performed by using the associated importance weights. When cumulative distribution functions (CDFs) of the normalized weights $\{\tilde{w}_m^i[k_s]\}_{i=1}^N$ are constructed, each element of the CDF is compared with a uniformly distributed function to determine whether the weights are relatively high or low. Based on this determination, the samples with low weights are eliminated, and the samples with high weights are retained for estimating or predicting the states. Therefore, a new set of states $\{\mathbf{x}_m^i[k_s]\}_{i=1}^N$ is generated, and this procedure is repeated for the next time instance. A general process of the Bayesian particle filter including the resampling step is also presented in Appendix A.3.

2.3.2 Downlink Transmission

Using the Bayesian particle filter, the posterior distribution of the downlink transmission in (2.10) is also empirically approximated by drawing a set of randomly chosen weighted samples such that

$$\hat{p}(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0}; k_{ds}]) = \frac{1}{N} \sum_{i=1}^N \tilde{w}_s^i[k'] \delta(\mathbf{f}_s[k'] - \mathbf{f}_s^i[k']). \quad (2.19)$$

The importance weight in (2.19) can be computed recursively by selecting the importance distribution $q(\cdot)$ as a prior distribution. Hence, the simplified form of the importance weight for the downlink transmission is given by

$$w_s^i[k'] = \tilde{w}_s^i[k' - 1] p(\mathbf{f}_m[k_{ds}] | \mathbf{f}_s^i[k']). \quad (2.20)$$

The implementation to obtain the estimate of force data can be relatively easy in practice, since we only need to compute the likelihood distribution in (2.20). After normalizing the importance weight as $\tilde{w}_s^i[k'] = w_s^i[k'] / \sum_{i=1}^N w_s^i[k']$, the resampling step needs to be performed to regenerate a new set of states $\{\mathbf{f}_s^i[k']\}_{i=1}^N$ based on the associated importance weights. In addition to the uplink transmission, the entire processes of the Bayesian particle filter, including the initialization, prediction, update, resampling, and iteration steps, are described in Table 2.1.

Table 2.1 Bayesian particle filters for the uplink and downlink transmissions

<p><i>1. Initialization step</i></p> <hr/> <p>Initialize states $\{\mathbf{x}_m^i[k_{s_0}]\}_{i=1}^N$, $\{\mathbf{f}_s^i[k'_0]\}_{i=1}^N$ by initial distributions, which may be known or chosen randomly</p> <hr/>
<p><i>2. Prediction step</i></p> <hr/> <p>Predict states $\mathbf{x}_m[k_s + 1]$, $\mathbf{f}_s[k' + 1]$, by distributions $p(\mathbf{x}_m[k_s + 1] \mathbf{x}_m^i[k_s], p(\mathbf{f}_s[k' + 1] \mathbf{f}_s^i[k'])$</p> <hr/>
<p><i>3. Update step</i></p> <hr/> <p>Compute importance weights $\{\tilde{w}_m^i[k_s]\}_{i=1}^N$ (2.18) and $\{\tilde{w}_s^i[k']\}_{i=1}^N$ (2.20)</p> <hr/>
<p><i>4. Resampling step</i></p> <hr/> <p>Draw N new states according to associated importance weights as $\{\mathbf{x}_m^i[k_s]\}_{i=1}^N$ and $\{\mathbf{f}_s^i[k']\}_{i=1}^N$</p> <hr/>
<p><i>5. Iteration step</i></p> <hr/> <p>Set $k = k + 1$, $k_s = k_s + 1$ and repeat from Step (2)</p> <hr/>

2.4 Discussion

The Bayesian particle filter introduced in Sect. 2.3 can be used in either prediction or estimation depending on the availability of observation. When a sample is lost during the network transmission, the observation at the corresponding time instance is not available at the receiving manipulator. In this case, the proposed Bayesian particle filter performs the prediction using the observations up to the past time instances. This practical approach of the Bayesian particle filter in the case of the packet loss and time-varying delay is presented in Chap. 4.

In Sect. 2.3, the Bayesian particle filter is applied only to find the estimates of position and force data. In the current network technology based on the Internet, the delay, which is time varying depending on network traffic conditions, is often expected to be estimated or predicted for further improvement in performance [1, 6]. Using the Bayesian particle filter, the estimate of network delay in addition to position and force data is considered and presented in Appendix B. In the uplink transmission, a concatenated vector, which consists of the position vector and the time-varying delay $n_m[k]$, is constructed and processed by the Bayesian particle filter in order to solve a multivariable estimation problem. A similar process is performed for the downlink transmission by constructing a concatenated vector of the time-varying delay $n_s[k_s]$ and force data.

References

1. T. Mirfakhrai, S. Payandeh, On using delay prediction in controlling force reflecting teleoperation over the Internet. *Robotica* **23**(6), 809–813 (2005)
2. C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2007)
3. Z. Chen, Bayesian filtering: From Kalman filters to particle filters, and beyond. Adaptive Systems Lab., McMaster Univ., Hamilton, ON, Canada [Online]. Available: http://soma.crl.mcmaster.ca/zhechen/download/ieee_bayesian.ps
4. B.D.O. Anderson, J.B. Moore, *Optimal Filtering* (Prentice-Hall, Englewood Cliffs, 1979)
5. A. Doucet, N. de Freitas, N. Gordon, *Sequential Monte Carlo Methods in Practice* (Springer, New York, 2001)
6. M.J. Coates, R.D. Nowak, Sequential Monte Carlo inference of internal delays in nonstationary communication networks. *IEEE Trans. Signal Process.* **50**, 366–376 (2002)
7. X.-L. Hu, T.B. Schon, L. Ljung, A basic convergence result for particle filtering. *IEEE Trans. Signal Process.* **56**(4), 1337–1348 (2008)

Chapter 3

Stability of Haptic Teleoperation Systems Based on the Bayesian Approach

Nomenclatures

P_{tot}	Total power flow in the teleoperation system
P_{net}	Power flow through the network
P_h	Entering power to the master
P_m	Leaving power from the master
P_s	Entering power to the slave
P_e	Leaving power from the slave
P_D	Dissipated power
H	Stored energy
S	Scattering transform matrix
v_m, u_m	Scattering variables at the master
v_s, u_s	Scattering variables at the slave
R	Rotation matrix
b_1, b_2	Impedance parameters
\dot{x}_m, \dot{x}_s	Velocity variables
T_s	Sampling rate

Stability is a fundamental requirement for control systems. If a system is not stable, there may be random variations in the system even though a small signal is applied. Hence, an unstable teleoperation system is not operable in practice. In this chapter, the notion of passivity, which represents a sufficient condition for stability of a system, is introduced in order to address stability of the proposed haptic teleoperation system based on the Bayesian approach. Given the stochastic formalism presented in Chap. 2, we use the control architectures, including the passivity-based scattering transformation [1] and the passivity observer/passivity controller [2]. The remainder of the chapter is organized as follows. In Sect. 3.1, we review the notion of passivity for haptic teleoperation systems. In Sect. 3.2, the scattering transformation is presented to address a passivity condition of the proposed system within the stochastic formalism of haptic data. In Sect. 3.3, stability of the proposed teleoperation system is also addressed using the passivity observer/passivity controller for the proposed teleoperation system.

3.1 Preliminary: Passivity of Haptic Teleoperation Systems

Before we show stability of the proposed haptic teleoperation system, we first present an overview of passivity formalism in a general one-port system and its extension to a haptic teleoperation system. A simple one-port system (manipulator) which has the input velocity vector $\dot{\mathbf{x}}$ and output force vector \mathbf{f} is shown in Fig. 3.1. Note that we use velocity and force vectors to introduce power and energy of a physical system and to define its closed-loop passivity. In this figure, \mathbf{P} is the entering power into the manipulator as the product of the input and output vectors of the manipulator. By definition, the one-port manipulator is said to be passive if it obeys

$$\mathbf{P}(t) = \dot{\mathbf{x}}^T(t)\mathbf{f}(t) = \dot{\mathbf{H}}(t) + \mathbf{P}_D(t). \quad (3.1)$$

$\mathbf{H}(t)$ is a lower bounded stored energy and $\mathbf{P}_D(t)$ is a nonnegative dissipative power at a continuous time t . Equation (3.1) shows that the entering power into the manipulator is either stored or dissipated. It also implies that the total energy supplied by the manipulator up to time t is limited to the initial stored energy such that

$$\begin{aligned} \int_0^t \mathbf{P}(\tau) d\tau &= \int_0^t \mathbf{f}(\tau) \dot{\mathbf{x}}(\tau) d\tau \\ &= \mathbf{H}(t) - \mathbf{H}(0) + \int_0^t \mathbf{P}_D(\tau) d\tau \geq -\mathbf{H}(0), \quad \forall t \geq 0 \end{aligned} \quad (3.2)$$

This shows that the energy transfer is lower bounded by the negative initial energy, which is $-\mathbf{H}(0)$. When the initial energy is $\mathbf{H}(0) = 0$, the energy supplied to a passive manipulator must be positive for all time. This is called a strictly passive condition as long as the stored energy does not reach the lower bound. If the power dissipation is zero for all time, the system is lossless [1].

A haptic teleoperation system consisting of master and slave manipulators can be modeled as an interconnected two-port system, which is shown in Fig. 3.2. In this figure, each individual manipulator is a two-port system that has entering and

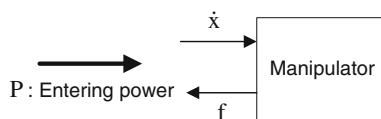


Fig. 3.1 Passivity of a one-port manipulator

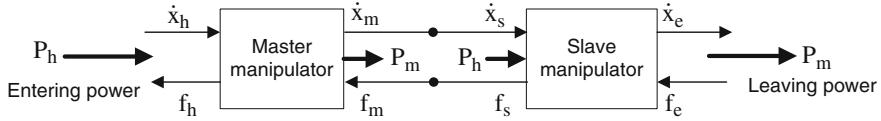


Fig. 3.2 Passivity of an interconnected two-port teleoperation system

leaving power flows. In the case that the delay between the master and the slave manipulators is negligible, the total power flow \mathbf{P}_{tot} of the interconnected system is

$$\begin{aligned}\mathbf{P}_{\text{tot}}(t) &= \mathbf{P}_h(t) - \mathbf{P}_m[k] + \mathbf{P}_s[k] - \mathbf{P}_e(t) \\ &= \dot{\mathbf{x}}_h^T(t)\mathbf{f}_h(t) - \dot{\mathbf{x}}_m^T[k]\mathbf{f}_m[k] + \dot{\mathbf{x}}_s^T[k]\mathbf{f}_s[k] - \dot{\mathbf{x}}_e^T(t)\mathbf{f}_e(t).\end{aligned}\quad (3.3)$$

\mathbf{P}_h and \mathbf{P}_m are the entering and leaving powers at the master manipulator, respectively. \mathbf{P}_s and \mathbf{P}_e are the entering and leaving powers at the slave manipulator, respectively. Note that for the discrete power flows, which are $\mathbf{P}_m[k]$ and $\mathbf{P}_s[k]$, we assume that the sampling process does not introduce any extra energy to the overall system so that the power flows between the continuous and the discrete data are preserved [3]. In the case of negligible delay, the interconnected two-port system is passive if (3.3) obeys

$$\mathbf{P}_{\text{tot}}(t) = \dot{\mathbf{H}}_1(t) + \mathbf{P}_{D1}(t) + \dot{\mathbf{H}}_2(t) + \mathbf{P}_{D2}(t). \quad (3.4)$$

$\mathbf{H}_1(t)$ and $\mathbf{H}_2(t)$ are the lower bounded stored energies at the master and slave manipulators, respectively. $\mathbf{P}_{D1}(t)$ and $\mathbf{P}_{D2}(t)$ are nonnegative dissipative powers at the master and slave manipulators, respectively. Equation (3.4) implies that the interconnected two-port teleoperation system with negligible delay is passive when the individual master and slave manipulators are passive elements.

3.2 Passivity Using Generalized Scattering Transformation

3.2.1 Generalized Scattering Transformation

When the network delay is presented between the master and slave manipulators, the power variables introduced by the network delay may have negative power flow, which may destabilize an overall teleoperation system [1]. A nonpassive condition of teleoperation system over constant network delay is shown in Appendix C.1. In the case of constant delay, the scattering transformation has been considered as one of prominent control approaches to ensure passivity of an overall teleoperation system [1, 4]. A generalized form of the scattering transformation, which reconfigures the velocity and force variables in (3.3), is given by [5]

$$\begin{bmatrix} \mathbf{v}_m & \mathbf{v}_s \\ \mathbf{u}_m & \mathbf{u}_s \end{bmatrix} = \mathbf{S} \begin{bmatrix} \dot{\mathbf{x}}_m & \dot{\mathbf{x}}_s \\ \mathbf{f}_m & \mathbf{f}_s \end{bmatrix} \quad (3.5)$$

where \mathbf{v}_m , \mathbf{u}_m , \mathbf{v}_s , and \mathbf{u}_s are the scattering variables. \mathbf{S} is the scattering transformation matrix, which consists of a rotation matrix \mathbf{R} and a scaling matrix \mathbf{B} such that

$$\mathbf{S} = \mathbf{RB} = \begin{bmatrix} \cos \theta \mathbf{I} & -\sin \theta \mathbf{I} \\ \sin \theta \mathbf{I} & \cos \theta \mathbf{I} \end{bmatrix} \begin{bmatrix} b_1 \mathbf{I} & 0 \\ 0 & b_2 \mathbf{I} \end{bmatrix} \quad (3.6)$$

where \mathbf{I} is an identity matrix with dimensions of the power variables. The transformation angle θ and impedance parameters, b_1 and b_2 , can be empirically chosen from $\theta \in [-\pi/2, \pi/2]$ and $b_1, b_2 > 0$. Note that specific choices of transformation angle and parameters result in a simplified form of scattering transformation, which is also known as the wave variables [4]. Using this scattering transformation, the power flow within the network becomes independent of any magnitude of a constant delay. Therefore, passivity of an overall teleoperation system can be guaranteed in the presence of constant delay [1, 4]. Such passive condition in the presence of constant delay is presented in Appendix C.2.

In the presence of time-varying delay, however, it is known that passivity may not be guaranteed even though the scattering transformation is used. For instance, if an increasing delay occurs due to network congestion, a receiving manipulator may hold the last received packet because the current packet is not received yet as presented in Fig. 2.3b of Chap. 2. In the case that the manipulator holds the old packet that causes extra energy to the overall teleoperation system, passivity may not be maintained [6, 7]. This nonpassive condition in the presence of time-varying delay is shown in Appendix C.3.

In the case of packet loss problem, when we choose the strategy that introduces a null packet instead of the unreceived packet as presented in Fig. 2.3c of Chap. 2, a null packet basically does not introduce extra energy to an overall teleoperation system, and therefore, passivity may be maintained [8]. This passive condition in the case of packet loss problem is presented in Appendix C.4. However, if the receiving manipulator holds the last received packet in this packet loss problem (Fig. 2.3b), it is also possible to introduce extra energy to the overall teleoperation system, which may be nonpassive.

3.2.2 Passivity of the Proposed Haptic Teleoperation Systems Using the Scattering Transformation

Based on the Bayesian approach presented in Chap. 2, transmitting haptic data packets that may be time-varying delayed or lost through the network can be obtained, given the previous state and available observations. The proposed haptic

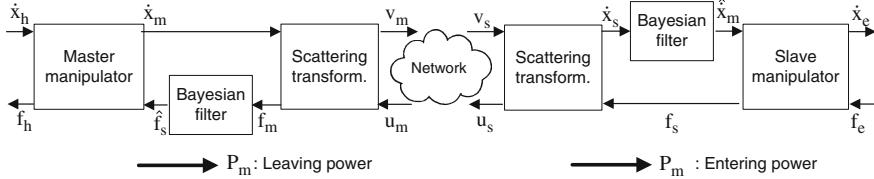


Fig. 3.3 A haptic teleoperation system using the proposed Bayesian particle filter and the scattering transformation

teleoperation system using the Bayesian approach and the generalized scattering transformation is shown in Fig. 3.3.

In order to show a passive condition of the proposed haptic teleoperation system, we consider the interconnected two-port system in Sect. 3.1. Assuming that the individual master and slave manipulators are passive elements, we are only interested in the power flow through the network \mathbf{P}_{net} , which consists of the leaving power from the master manipulator \mathbf{P}_m and the entering power into the slave manipulator \mathbf{P}_s . From the total power flow of an overall teleoperation system (3.3), the power flow through the network, which is based on the estimates of power variables, is given by

$$\mathbf{P}_{\text{net}}[k] = -\dot{\mathbf{x}}_s^T[k]\hat{\mathbf{f}}_m[k'] + \dot{\mathbf{x}}_m^T[k_s]\mathbf{f}_s[k_s] \quad (3.7)$$

where $\dot{\mathbf{x}}_m[k_s]$ is the estimate of velocity at the slave manipulator and $\hat{\mathbf{f}}_m[k']$ is the estimate of force at the master manipulator. Note that from Chap. 2, k_s and k' are the time instances of slave and master clocks, respectively, and k' can be initiated by $k + k_{ds_0}$. Ideally, the optimal solutions, which are shown in (2.7) and (2.10), are desired to obtain the estimates. However, since these optimal solutions are not computationally tractable, the Bayesian particle filters are used to provide the approximated solutions.

According to the convergence results of the Bayesian particle filters [9, 10], the approximations, (2.16) and (2.19), approach to the optimal solutions, (2.7) and (2.10), respectively when the number of particles tends to infinity. In other words, as $N \rightarrow \infty$,

$$\begin{cases} \left\{ \widehat{\dot{\mathbf{x}}}_m^i[k_s] \right\}_{i=1}^N \xrightarrow{\text{a.s.}} \mathbb{E}(\dot{\mathbf{x}}_m[k_s] | \dot{\mathbf{x}}_s[k_{d_0} : k_d]) \\ \left\{ \widehat{\mathbf{f}}_s^i[k'] \right\}_{i=1}^N \xrightarrow{\text{a.s.}} \mathbb{E}(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{ds_0} : k_{ds}]) \end{cases}. \quad (3.8)$$

In (3.8), the approximations obtained by the Bayesian particle filters *almost surely* (a.s.) converge to the optimal solutions, as the number of particles tends to infinity. Note that in probability theory, an event occurs almost surely if it happens with probability one [11]. The conditional expectations in (3.8) can be further

simplified because the expected variables in the optimal solutions are subsets of the observations, which are

$$\begin{cases} p(\dot{\mathbf{x}}_m[k_s]) \subset p(\dot{\mathbf{x}}_s[k_{d_0} : k_d]) \\ p(\mathbf{f}_s[k']) \subset p(\mathbf{f}_m[k_{ds_0} : k_{ds}]) \end{cases}. \quad (3.9)$$

Then, using the generalized form of scattering transformation in (3.5) and (3.6), the expectations of power variables can be represented in terms of the scattering variables such that

$$\begin{cases} \mathbb{E}(\dot{\mathbf{x}}_m[k_s]) = \frac{1}{b_1} \mathbb{E}(\cos \theta \mathbf{u}_s[k_s] + \sin \theta \mathbf{v}_s[k_s]) \\ \mathbb{E}(\mathbf{f}_s[k']) = \frac{1}{b_2} \mathbb{E}(\cos \theta \mathbf{u}_m[k'] - \sin \theta \mathbf{v}_m[k']) \end{cases}. \quad (3.10)$$

By solving the matrix in (3.5), every scattering variable consists of one of the stochastic variables, which is $\mathbb{E}(\dot{\mathbf{x}}_s[k_s])$ or $\mathbb{E}(\mathbf{f}_m[k'])$. Therefore, the power variables $\dot{\mathbf{x}}_m$ and \mathbf{f}_s in (3.7) can also be represented in stochastic forms such that

$$\begin{cases} \dot{\mathbf{x}}_m[k] = \frac{1}{b_1} \mathbb{E}(\cos \theta \mathbf{u}_m[k] + \sin \theta \mathbf{v}_m[k]) \\ \mathbf{f}_s[k_s] = \frac{1}{b_2} \mathbb{E}(\cos \theta \mathbf{u}_s[k_s] - \sin \theta \mathbf{v}_s[k_s]) \end{cases}. \quad (3.11)$$

Substituting (3.10) and (3.11) into (3.7), the power flow in the network can be transformed to

$$\begin{aligned} \mathbf{P}_{\text{net}}[k] = & \frac{1}{b_1 b_2} \{ \mathbb{E}(\cos \theta \mathbf{v}_s[k_s] + \sin \theta \mathbf{u}_s[k_s]) \mathbb{E}(\cos \theta \mathbf{u}_s[k_s] - \sin \theta \mathbf{v}_s[k_s]) \\ & - \mathbb{E}(\cos \theta \mathbf{v}_m[k'] + \sin \theta \mathbf{u}_m[k']) \mathbb{E}(\cos \theta \mathbf{u}_m[k'] - \sin \theta \mathbf{v}_m[k']) \} \end{aligned} \quad (3.12)$$

In (3.12), the scattering variables at the master manipulator, \mathbf{v}_m and \mathbf{u}_m , need to be computed at time k' for synchronization. Using the linearity property of the expected value, which is given by $\mathbb{E}(\cos \theta \mathbf{v}_m[k] + \sin \theta \mathbf{u}_m[k]) = \cos \theta \mathbb{E}(\mathbf{v}_m[k]) + \sin \theta \mathbb{E}(\mathbf{u}_m[k])$, Eq. (3.12) can be computed as

$$\begin{aligned} \mathbf{P}_{\text{net}}[k] = & \frac{\sin 2\theta}{2b_1 b_2} \left\{ \mathbb{E}(\mathbf{v}_m[k'])^2 + \mathbb{E}(\mathbf{u}_s[k_s])^2 - \mathbb{E}(\mathbf{v}_s[k_s])^2 - \mathbb{E}(\mathbf{u}_m[k'])^2 \right. \\ & \left. + 2 \cot 2\theta \mathbb{E}(\mathbf{v}_s[k_s] \mathbf{u}_s[k_s]) - \mathbf{v}_m[k'] (\mathbf{u}_m[k']) \right\}. \end{aligned} \quad (3.13)$$

Equation (3.13) is the power flow using the generalized form of scattering transformation. To obtain a specific form of the scattering transformation, a transformation angle can be selected as $\theta = \theta_c$ that has the solutions of $\sin 2\theta_c \geq 0$ and $\cot 2\theta_c = 0$. When the expected scattering variables are achieved by the optimal solutions, the delayed scattering variables, $\mathbb{E}(\mathbf{v}_s[k_s])$ and $\mathbb{E}(\mathbf{u}_m[k'])$, can be

replaced by the scattering variables from the sending manipulators. Therefore, the power flow in the network (3.13) can be further simplified such that

$$\mathbf{P}_{\text{net}}[k] = \frac{\sin 2\theta_c}{2b_1 b_2} \left\{ \mathbb{E}(\mathbf{v}_m[k'])^2 - \mathbb{E}(\mathbf{v}_m[k])^2 \right\}. \quad (3.14)$$

The energy stored in the network can be obtained by integrating the power (3.14) as

$$\begin{aligned} \mathbf{H}_{\text{net}}[k_c] &= \frac{T_s \sin 2\theta_c}{2b_1 b_2} \sum_{k=0}^{k_c} \left\{ \mathbb{E}(\mathbf{v}_m[k'])^2 - \mathbb{E}(\mathbf{v}_m[k])^2 \right\} \\ &= \frac{T_s \sin 2\theta_c}{2b_1 b_2} \sum_{k=0}^{k_c+k_{ds_0}} \mathbb{E}(\mathbf{v}_m[k])^2 \end{aligned} \quad (3.15)$$

where T_s is a sampling rate to obtain sampled data in discrete time from the master and the slave manipulators. In (3.15), since the stored energy up to the time instance k_c is $\mathbf{H}_{\text{net}}[k_c] \geq 0$, based on the notion of passivity, the overall haptic teleoperation system can be stable.

3.3 Stability Using the Passivity Observer/Passivity Controller

Stability of a haptic teleoperation system based on the proposed Bayesian approach can be also addressed by using the passivity observer/passivity controller, which is also one of control architectures to ensure passivity [2, 12]. When observed haptic data, which are transmitted over the network and received at the manipulators, are estimated or predicted by the proposed Bayesian approach, the expected power variables are $\hat{\mathbf{f}}_s[k']$ at the master manipulator and $\hat{\mathbf{x}}_m[k_s]$ at the slave manipulator. Then, the energy supplied by the master manipulator up to the time instance k_c is given by

$$\mathbf{H}_m[k_c] = T_s \sum_{k=0}^{k_c} \dot{\mathbf{x}}_m[k] \hat{\mathbf{f}}_s[k']. \quad (3.16)$$

As defined in Sect. 3.1, with the initial energy storage $\mathbf{H}_m[0] = 0$, the master manipulator is passive if $\mathbf{H}_m[k_c] \geq 0$ for every $k \geq 0$, and this implies that the system power is either stored or dissipated. When $\mathbf{H}_m[k_c] < 0$ at any time instance, the system generates extra energy, which may cause instability. In the case of (3.16), since the time instance k' contains the delay k_{ds_0} , this may result in negative energy as presented in Appendix C.1. In order to maintain passivity, we use the passivity observer that monitors the time instances when the energy becomes negative and

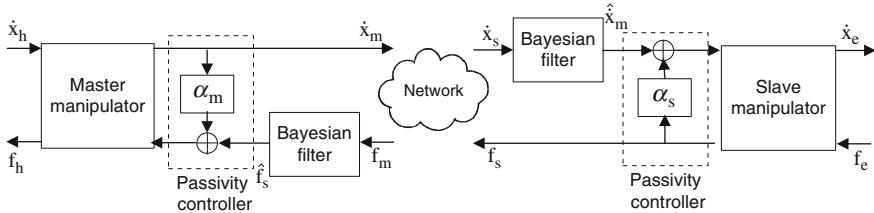


Fig. 3.4 A haptic teleoperation system based on the proposed Bayesian approach and the passivity observer/passivity controller

also measures the exact amount of the extra energy in real time. Given the information from the passivity observer, the passivity controller, which is a software-implemented damping element, is applied to dissipate any extra energy at each time instance [2]. Based on the proposed Bayesian approach, the use of passivity observer/passivity controller in both the master and the slave manipulators is shown in Fig. 3.4. In this figure, α_m and α_s are the adjustable damping elements at the master and slave manipulators, respectively. The slave manipulator also uses the passivity observer/passivity controller to dissipate any extra energy in real time if the supplied energy computed by the power variables $\hat{x}_m[k_s]$, $f_s[k_s]$ is negative.

3.4 Discussion

In this chapter, we present two different control architectures, which are the scattering transformation and the passivity observer/passivity controller, in order to address stability of the haptic teleoperation system based on the Bayesian approach. In the case of scattering transformation, the passivity condition (3.15) highly depends on the performance of the estimation or prediction method. In other words, passivity is ensured if the estimates of power variables are achieved by the optimal solutions. When the proposed Bayesian particle filter is used, the convergence to optimal solutions is theoretically achieved with an infinitely large number of particles [9, 10]. Since such large number of particles introduces computational complexity, a limited number of particles should be chosen for practical convergence. In Chap. 4, we conduct a preliminary experimental study in order to determine the number of particles for obtaining the practical convergence while performing in real time.

On the other hand, when the passivity observer/passivity controller is used for the proposed teleoperation system, the processes of monitoring a nonpassive condition and dissipating any extra energy are independent of the processes of estimating or prediction the power variables. Therefore, by using additional damping elements at the master and slave manipulators, passivity of an overall teleoperation system may be achieved regardless of the performance of the proposed estimation

or prediction method. However, such additional use of damping elements may deteriorate haptic data, which eventually degrade the transparency performance of an overall teleoperation system.

References

1. R.J. Anderson, M.W. Spong, Bilateral control of teleoperators with time delay. *IEEE Trans. Autom. Control* **34**(5), 494–501 (1989)
2. B. Hannaford, J.-H. Ryu, Time-domain passivity control of haptic interfaces. *IEEE Trans. Robot. Autom.* **18**, 1–10 (2002)
3. S. Stramigioli, C. Secchi, A.J. van der Schaft, C. Fantuzzi, A novel theory for sample data system passivity, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp. 1936–1941
4. G. Niemeyer, J.J.E. Slotine, Stable adaptive teleoperation. *IEEE J. Oceanic Eng.* **16**(1), 152–162 (1991)
5. S. Hirche, T. Matiakis, M. Buss, A distributed controller approach for delay-independent stability of networked control systems. *Automatica* **45**(8), 1828–1836 (2009)
6. R. Lozano, N. Chopra, M. Spong, Passivation of force reflecting bilateral teleoperators with time varying delay, in *Proceedings 8th Mechatronics Forum*, Enschede, The Netherlands (2002), pp. 954–962
7. N. Chopra, P. Berestesky, M.W. Spong, Bilateral teleoperation over unreliable communication networks. *IEEE Trans. Control Syst. Technol.* **16**(2), 304–313 (2008)
8. C. Secchi, S. Stramigioli, C. Fantuzzi, Digital passive geometric telemanipulation, in *Proceedings IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sept 2003, pp. 3290–3295
9. D. Crisan, A. Doucet, A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Sig. Process.* **50**(3), 736–746 (2002)
10. X.-L. Hu, T.B. Schon, L. Ljung, A basic convergence result for particle filtering. *IEEE Trans. Sig. Process.* **56**(4), 1337–1348 (2008)
11. A. Papoulis, S.U. Pillai, *Probability, Random Variables and Stochastic Processes* (McGraw-Hill, New York, 2002)
12. J.-H. Ryu, D.-S. Kwon, B. Hannaford, Stability guaranteed control: time domain passivity approach. *IEEE Trans. Control Syst. Technol.* **12**(6), 860–868 (2004)

Chapter 4

A Layered Protocol Architecture for Haptic Teleoperation Systems Based on the Bayesian Approach

Nomenclatures

\dot{f}_m, \dot{f}_s	Rates of force change
c	Filter coefficient

In this chapter, we present a layered protocol architecture for haptic teleoperation systems over unreliable network conditions, such as time-varying delay, packet loss, and out-of-ordered packets. In the transport layer of the network, we propose a modified transport protocol that provides information about unreliable network conditions to the application layer. In the application layer, we use the Bayesian approach proposed in Chap. 2. Given the information about unreliable network conditions from the transport layer, the proposed Bayesian approach effectively performs either prediction or estimation.

This chapter is organized as follows. In Sect. 4.1, we review the layered protocol architecture of the network for haptic teleoperation systems. In Sect. 4.2, after reviewing the existing transport protocols for haptic teleoperation systems, we propose a modified packet structure of the standard transport protocol. In Sect. 4.3, we present the detailed operation of the application layer, which is incorporated with the proposed method in the transport layer. In Sect. 4.4, an experimental study is provided using two haptic devices through the local network.

4.1 Layered Protocol Architecture for Haptic Teleoperation Systems

In the current Internet, a typically used layered protocol architecture is the *Internet protocol (IP) suite* that consists of the following four layers: the application, transport, network, and link layers [1]. Haptic position and force data transmitted through this four-layer architecture is shown in Fig. 4.1.

The application layer is the highest layer that is interacted by the human operator. Since the human operator is able to process his or her own approach with the software application, haptic teleoperation systems over unreliable network

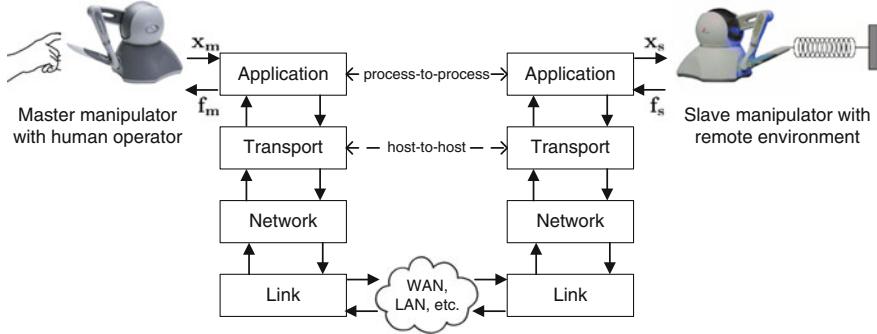


Fig. 4.1 Haptic data transmission through the four-layer architecture of the Internet protocol suite

conditions have been mostly studied within the application layer. Approaches based on the control architectures (e.g. scattering transformation [2], proportional derivative [PD] control [3], and passivity observer/passivity controller [4]) and the signal processing algorithms (e.g. Kalman filter [5] and Bayesian particle filter) can be implemented in the application layer by using the software application.

The transport layer provides host-to-host communication to the application layer. In other words, this layer handles transferring haptic data between two computers that directly operate the master and slave manipulators. For haptic teleoperation systems, the modified transport protocols based on the standard transport control protocol (TCP) and user datagram protocol (UDP) [6, 7], the real-time transport protocol designed (RTP) for multimedia systems [8], the adaptive buffer [9], and the transport protocol for wireless transmission [10] can be implemented in the transport layer by using the software application.

The network layer is responsible for sending transport layer packets by defining the IP addresses and routing structures. The link layer is implemented in a hardware-based network interface card, such as Ethernet or WiFi, in order to perform the physical transmission of haptic data. Since these two lower layers are mostly implemented in hardware, modifications to these layers are difficult to be achieved by the human operator. On the other hand, the two higher layers, which are the application and transport layers, can be effectively modified in software in order to improve the performance of haptic teleoperation systems. Therefore, in this chapter, we focus our attention on the two higher layers for the proposed layered protocol architecture of haptic teleoperation systems.

4.2 Transport Layer

The transport layer for a haptic teleoperation system provides connectivity between the master and the slave manipulators. A transport protocol in this layer is responsible for packetizing and transmitting haptic data to the application layer.

In this section, we review the existing standard protocols, including TCP, UDP, and RTP, with their applicability to haptic teleoperation systems. We also review the modified transport protocols developed for haptic teleoperation systems. Then, we propose a modified packet structure of UDP, which provides the information about unreliable network conditions to the application layer.

4.2.1 Existing Transport Protocols

For the existing transport protocols, we review the three standard protocols, TCP, UDP, and RTP, and the three modified protocols for haptic teleoperation systems in the literature: the real-time network protocol (RTNP), the interactive real-time protocol (IRTP), and the efficient transport protocol (ETP). The detailed packet structures of the existing protocols are also shown in Appendix D.

- **Transport Control Protocol:** TCP is a transport layer network protocol that provides a reliable and connection-oriented service to various network applications. This protocol employs the packet retransmission mechanism for reliable transmission and performs congestion control to regulate the transmission rate through the network. In haptic teleoperation systems, TCP may be useful for establishing an initial connection between the master and the slave manipulators and for delivering crucial data. However, it is not generally recommended because the retransmission mechanism and congestion control algorithm may cause relatively high variations in delay and delay jitter.
- **User Datagram Protocol:** UDP packages application data and immediately passes the packets to the network layer. Unlike TCP, it does not have a retransmission mechanism, and it does not perform any congestion control either. Therefore, the data transfer using UDP can be accomplished without significant delay and variations. As UDP is widely used for many real-time applications over the network, such as streaming multimedia and voice over IP, it is also preferred for haptic teleoperation system [11]. However, since it provides an unreliable and connectionless service, the packet loss, packet replication, and out-of-order delivery problems are more significant in general [1].
- **Real-Time Transport Protocol:** RTP has been developed as a standard transport protocol for recent real-time network applications, such as video and audio streaming services. This protocol employs an intermediate buffer in order to deal with the variations in delay, and it is suitable for real-time video and audio services [1]. In the case of haptic teleoperation systems, however, such intermediate buffer introduces processing delay, which eventually causes additional constant delay between the master and the slave manipulators.
- **Real-Time Network Protocol:** As a transport protocol for haptic teleoperation systems, RTNP has been introduced for UNIX environments because the end-to-end delay depends not only on the network condition but also on the software supported by the operating system. Due to such limitation, this

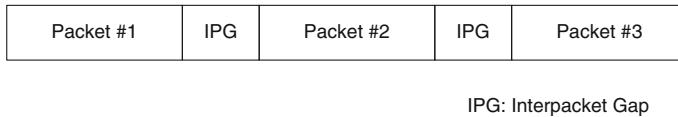


Fig. 4.2 Interpacket gaps between data packets

protocol is not available for other types of platforms, such as Windows-based operation systems [6].

- **Interactive Real-Time Protocol:** IRTP reconfigures and simplifies packet headers of haptic data in order to reduce the end-to-end delay between the master and the slave manipulators. This protocol employs TCP for the connection establishment and the transmission of crucial data. It then uses UDP to transmit the remaining haptic data that need to be transmitted without significant variations in delay [7].
- **Efficient Transport Protocol:** ETP focuses on reducing the end-to-end delay between the master and the slave manipulators by introducing a gap between two successive data packets [12, 13]. This is called the interpacket gap (IPG) and a simple illustration is shown in Fig. 4.2. The amount of IPG can be controlled depending on the network condition, and such IPG control may be used as a congestion control that TCP performs with window size. In the case of network congestion, for example, the amount of IPG increases in order to reduce the data rate within available bandwidth. Since UDP alone does not provide any congestion control mechanism, the IPG control is recommended based on UDP protocol.

4.2.2 Proposed Transport Protocol

As various transport protocols are reviewed in Sect. 4.2.1, the role of transport protocols is to provide connectivity between the master and the slave manipulators and deliver data packets to the application layer. Most approaches based on control architectures and signal-processing algorithms are basically performed within the application layer. In this study, we propose a transport protocol based on UDP in order to achieve relatively faster transmission of haptic data and to maintain lower variations in delay. Then, the packet structure of UDP is modified to provide additional information about unreliable network conditions to the application layer. The modified packet structure based on UDP is shown in Fig. 4.3.

In a standard UDP packet, a header consists of the four fields and each field is assigned by two bytes: source and destination port numbers, length of the packet, and checksum that allows a receiver to verify the incoming packet. The details of UDP packet structure are also shown in Appendix D.1. As shown in Fig. 4.3, a four-byte *packet number field*, which is one of the main features of TCP, is

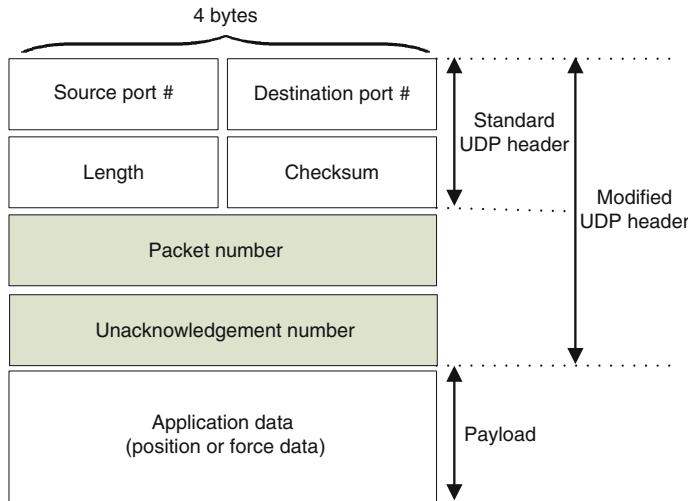


Fig. 4.3 The proposed packet structure based on *user datagram protocol* (UDP; multiple UDP [M-UDP]). In the packet header, the packet number field is assigned to the sending manipulator and the unacknowledgment number field is assigned to the receiving manipulator

additionally assigned to the header of UDP packet. In TCP, the packet number field is used at a sender to send sequential packets with unique numbers. When the packets are received, a receiver is able to detect whether packets are received or not by observing these successive packet numbers. If packets are received in order, the receiver sends back *acknowledgment* (ACK) to the sender. Otherwise, it sends back unacknowledgment (NACK) for requesting retransmission to the sender if any missing or out-of-ordered packets are detected. In the proposed modified UDP (M-UDP), the sending manipulator uses the packet number field to assign unique numbers to every successive packet of haptic data. After M-UDP packets are transmitted over the network, the receiving manipulator examines the NACK *number field* in order to detect any missing or time-varying delayed packets. The NACK number is only used to deliver the information about unreliable network conditions to the application layer rather than to send back to the sending manipulator. An operation example of the proposed M-UDP between the master and the slave manipulators is shown in Fig. 4.4.

As shown in Fig. 4.4, each sample of position and force data is assigned by a unique packet number and transmitted over the network. While the slave manipulator runs by its own local clock (e.g. 1 kHz), the position packet #3 is not received yet due to packet loss, time-varying delay, or out-of-ordered packets. Hence, the NACK number field at the slave manipulator detects this unreceived packet and informs to the application layer. Given the NACK number from the transport layer, the application layer has the knowledge of which packet needs to be predicted, and therefore, the proposed Bayesian approach can be performed as the prediction-based method.

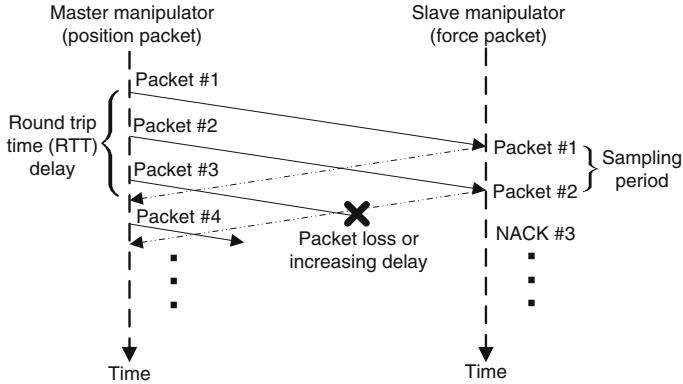


Fig. 4.4 A haptic teleoperation using the M-UDP

4.3 Application Layer

A haptic teleoperation system based on the proposed methods in the transport and application layers is shown in Fig. 4.5. As presented in Sect. 4.2, the M-UDP send and receive protocols are in the transport layer in order to detect unreliable network conditions. Given the information from the transport layer, the proposed Bayesian particle filter is performed in the application layer. In this section, we first present the detailed operation of the Bayesian particle filter based on the M-UDP. We also present the prediction-based Bayesian particle filter using available observations. For further postprocessing after the prediction, a low-pass filter is optionally used.

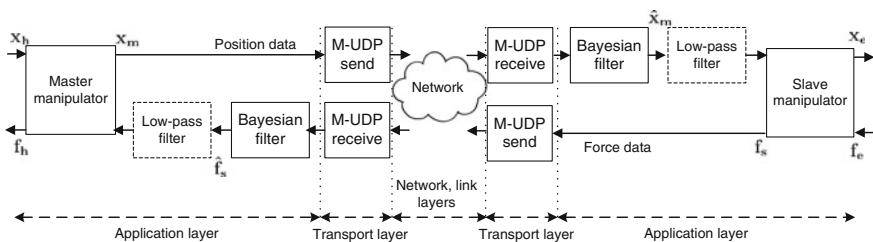


Fig. 4.5 A haptic teleoperation based on the proposed methods in the transport and application layers. The M-UDP send and receive protocols are in the transport layer. Given the information from the M-UDP, the Bayesian particle filter is used in the application layer. The low-pass filter is optionally used

4.3.1 Bayesian Particle Filter Based on M-UDP

The Bayesian particle filter presented in Chap. 2 can be used as either prediction or estimation depending on the observation availability. Although it can be performed as the estimation-based method to compensate for unreliable network conditions, its performance without the information about observation availability may be limited. To describe more specifically, let us recall the posterior distributions of position and force data that are computed by the Bayesian particle filter (2.16) and (2.19):

$$\begin{cases} p(\mathbf{x}_m[k_s] | \mathbf{x}_s[k_{d_0} : k_d]) \approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_m^i[k_s] \delta(\mathbf{x}_m[k_s] - \mathbf{x}_m^i[k_s]) \\ p(\mathbf{f}_s[k'] | \mathbf{f}_m[k_{d_{s_0}} : k_{d_s}]) \approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_s^i[k'] \delta(\mathbf{f}_s[k'] - \mathbf{f}_s^i[k']) \end{cases}. \quad (4.1)$$

In the uplink transmission, if the observation $\mathbf{x}_s[k_d]$ is available up to the current time instance of the slave local clock k_s , the Bayesian particle filter performs the estimation. This example is shown in Fig. 4.6a. If the current observation is not available due to the packet loss or time-varying delay, the Bayesian particle filter performs the prediction using the previous observation $\mathbf{x}_s[k_d - 1]$ as shown in Fig. 4.6b. Similarly, in the downlink transmission, the estimation is performed when the current observation $\mathbf{f}_m[k_{d_s}]$ is available. Otherwise, the prediction is performed using the previous observation $\mathbf{f}_m[k_{d_s} - 1]$.

As presented in Sect. 4.2.2, the M-UDP in the transport layer provides the information about observation availability to the application layer, and hence, the Bayesian particle filter can be switched between the prediction and the estimation, which not only enables an efficient use of the proposed method but also improves the tracking performance. Further detailed flowcharts of operating the M-UDP-based Bayesian particle filter are presented in Fig. 4.7. In the case of position data compensation, which is shown in Fig. 4.7a, when the slave manipulator receives a packet, the M-UDP receive protocol in the transport layer uses the NACK number field to detect the observation availability. Given the current time instance k_s of the slave local clock, if the received packet is the current observation, which is $k_s = k_d$, the Bayesian particle filter in the application layer performs the estimation to obtain the state $\hat{\mathbf{x}}_m[k_s]$ by computing the posterior distribution $p(\mathbf{x}_m[k_s] | \mathbf{x}_s[k_{s_0} : k_s])$. If the current observation is not available because of any packet loss or time-varying delay, which is $k_s > k_d$, the proposed Bayesian particle filter is performed as the prediction method by computing the prior distribution $p(\mathbf{x}_m[k_s] | \mathbf{x}_s[k_{s_0} : k_s - 1])$, with the previous observation $\mathbf{x}_s[k_s - 1]$. When the state is obtained at the current time instance k_s , the entire process is repeated for the next time instance $k_s + 1$, which evolves at the sampling rate of the slave local clock (e.g. 1 kHz).

In the case of force data compensation, which is shown in Fig. 4.7b, a similar process is performed to obtain the state of force data. When the current observation is available, which is $k' = k_{d_s}$, the Bayesian particle filter performs the estimation by computing the posterior distribution $P(\mathbf{f}_s[k'] | \mathbf{f}_m[k'_0 : k'])$. If the current

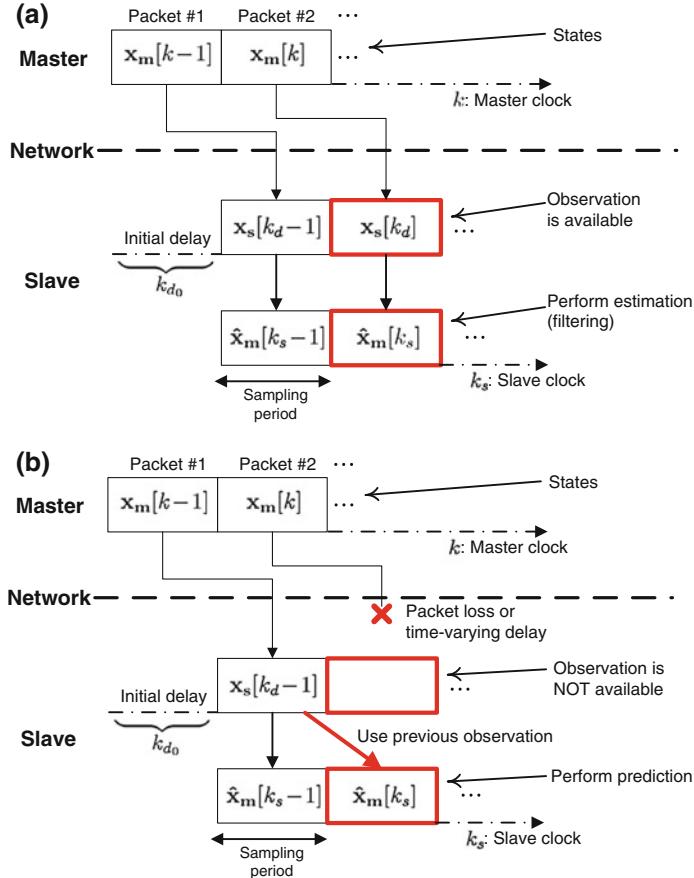


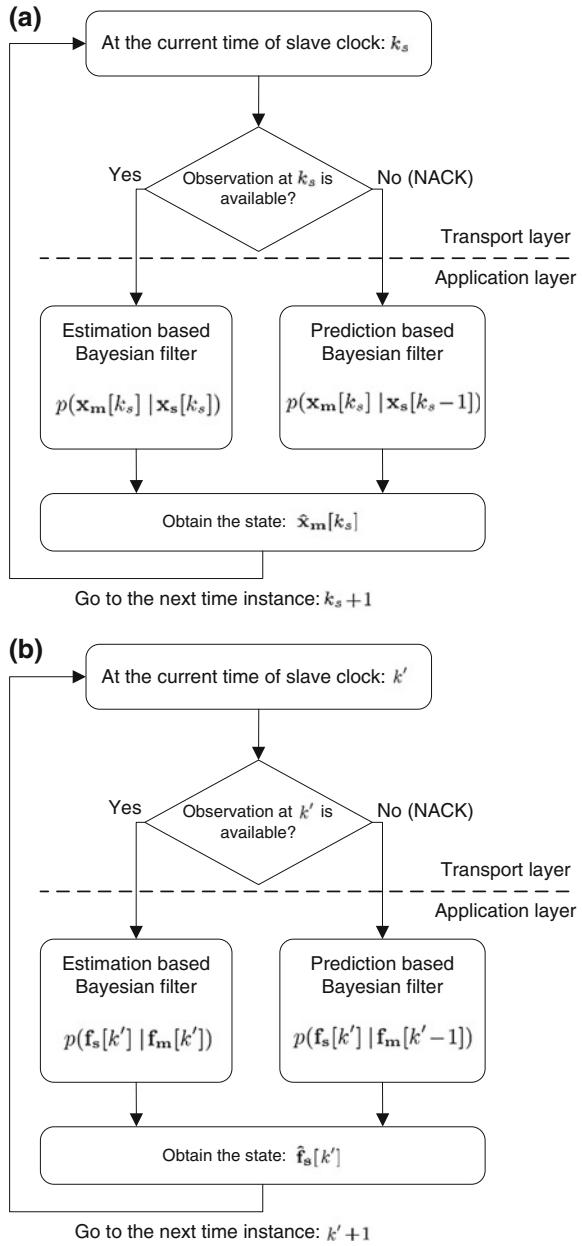
Fig. 4.6 The use of Bayesian particle filter depending on the observation availability: **a** The estimation and **b** prediction methods

observation is not available, which is $k' > k_{ds}$, the prediction is performed by computing the prior distribution $P(\mathbf{f}_s[k']|\mathbf{f}_m[k'_0 : k'-1])$.

4.3.2 Prediction-Based Bayesian Particle Filter

In Sect. 2.3, the Bayesian particle filter is presented in the estimation-based method in order to compute the posterior distributions, $P(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{s_0} : k_s])$ and $P(\mathbf{f}_s[k']|\mathbf{f}_m[k'_0 : k'])$. When the current observations are not available, the Bayesian particle filter is required to solve the prior distributions, $P(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{s_0} : k_s - 1])$ and $P(\mathbf{f}_s[k']|\mathbf{f}_m[k'_0 : k'-1])$. In this section, we present the prediction-based Bayesian particle filter by introducing a new state vector of position and force data.

Fig. 4.7 Flowcharts of the M-UDP-based Bayesian particle filter for **a** position data compensation, and **b** force data compensation



4.3.2.1 Uplink Transmission

In order to perform the proposed prediction-based method, we first redefine the general state-space model (2.2) as the prediction model of position data. A new

position state vector, which is not directly observable from the slave manipulator, is given by

$$\mathbf{x}_m[k+1] = \mathbf{g}_1(\mathbf{x}_m[k] + \dot{\mathbf{x}}_m[k]) + \mathbf{l}_1[k], \quad (4.2)$$

where $\dot{\mathbf{x}}_m[k]$ is the velocity vector, $\mathbf{l}_1[k]$ is the state noise sequence, and $\mathbf{g}_1(\cdot)$ is the state transition function. In this state vector, the position vector at the next time instance $k+1$ is described by the current position vector, the current velocity vector, and the noise sequence. Note that the state vector (4.2) is modeled when we assume the human operator performs unspecified teleoperation tasks to generate continuous hand movement. When the human operator does not move the master manipulator, which is at a stationary position, the velocity vector $\dot{\mathbf{x}}_m[k]$ becomes zero, and hence, the state vector (4.2) depends only on the current position and the state noise.

When the current observation of position data is not available in the uplink transmission, the prior distribution $p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{s_0} : k_s-1])$ using the Bayesian particle filter is given by

$$p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{s_0} : k_s-1]) \approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_m^i[k_s] \delta(\mathbf{x}_m[k_s] - \mathbf{x}_m^i[k_s]). \quad (4.3)$$

The importance weight is also given by

$$w_m^i[k_s] = \tilde{w}_m^i[k_s-1] p(\mathbf{x}_s[k_s-1] | \mathbf{x}_m^i[k_s]). \quad (4.4)$$

Note that for the estimation-based method, the posterior distribution and importance weights are given in (2.16) and (2.18), respectively. Since the previous state $\mathbf{x}_m[k-1]$ is known at the time instance k_s , the distribution in (4.4) can be solved using the previous state and the prediction model in (4.2). For example, assuming a normal distribution with zero mean and unit variance, Eq. (4.4) can be solved as

$$w_m^i[k] = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_s[k_s-1] - \mathbf{x}_m^i[k_s]) \right\}. \quad (4.5)$$

In order to compute $\mathbf{x}_m^i[k_s]$ in (4.5), we use the prediction model (4.2) such that

$$\mathbf{x}_m^i[k_s] = \mathbf{g}_1(\mathbf{x}_m^i[k_s-1] + \dot{\mathbf{x}}_m^i[k_s-1] + \mathbf{l}_1^i[k_s-1]). \quad (4.6)$$

The computation of $\mathbf{x}_m^i[k_s]$ needs to be performed for every i number of particles.

Alternatively, the prior distribution $p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{s_0} : k_s-1])$ can also be obtained directly from the estimation result of $p(\mathbf{x}_m[k_s-1]|\mathbf{x}_s[k_{s_0} : k_s-1])$. Since the distribution $p(\mathbf{x}_m[k_s]|\mathbf{x}_s[k_{s_0} : k_s-1])$ is evolved from $p(\mathbf{x}_m[k_s-1]|\mathbf{x}_s[k_{s_0} : k_s-1])$, we first compute $p(\mathbf{x}_m[k_s-1]|\mathbf{x}_s[k_{s_0} : k_s-1])$ using the estimation-based method in (2.16). After $\dot{\mathbf{x}}_m^i[k_s-1]$ is obtained from the posterior distribution

$p(\mathbf{x}_m[k_s - 1] | \mathbf{x}_s[k_{s_0} : k_s - 1])$. $\hat{\mathbf{x}}_m^i[k_s]$ can be directly computed by using the prediction model (4.2).

4.3.2.2 Downlink Transmission

To perform the proposed prediction-based method in the downlink transmission, we also redefine the general state-space model (2.5) as the prediction model of force data. A new force state vector, which is not directly observable from the master manipulator, is given by

$$\mathbf{f}_s[k_s + 1] = \mathbf{g}_2(\mathbf{f}_s[k_s] + \dot{\mathbf{f}}_s[k_s] + \mathbf{l}_2[k_s]), \quad (4.7)$$

where $\dot{\mathbf{f}}_s[k_s]$ is the difference computed by the last two force vectors, $\mathbf{l}_2[k_s]$ is the state noise sequence, and $\mathbf{g}_2(\cdot)$ is the state transition function. In this state vector, the force vector at the next time instance $k_s + 1$ is described by the current force vector, the current force difference, and the noise sequence.

When the current observation of force data is not available in the downlink transmission, the prior distribution $p(\mathbf{f}_s[k'] | \mathbf{f}_m[k'_0 : k' - 1])$ using the Bayesian particle filter is given by

$$p(\mathbf{f}_s[k'] | \mathbf{f}_m[k'_0 : k' - 1]) \approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_s^i[k'] \delta(\mathbf{f}_s[k'] - \mathbf{f}_s^i[k']). \quad (4.8)$$

The importance weight is also given by

$$w_s^i[k'] = \tilde{w}_s^i[k' - 1] p(\mathbf{f}_m[k' - 1] | \mathbf{f}_s^i[k']). \quad (4.9)$$

At the time instance k' , the previous state $\mathbf{f}_s[k' - 1]$ is obtained so that the likelihood distribution in (4.9) can be solved using the previous state and the prediction model of force data (4.7). When the estimation result of $p(\mathbf{f}_s[k' - 1] | \mathbf{f}_m[k'_0 : k' - 1])$ is solved, $\mathbf{f}_s^i[k']$ can also be directly obtained by computing the force prediction model (4.7) for every i number of particles.

4.3.3 Low-Pass Filter

A low-pass filter is a widely used signal-processing technique that passes low-frequency signals and reduces amplitudes of high-frequency signals based on a predefined cutoff frequency [14]. Depending on the prediction performance of the proposed Bayesian particle filter, the resulting waveforms of position and force data may contain jitter or distortion. Using the low-pass filter after the proposed method

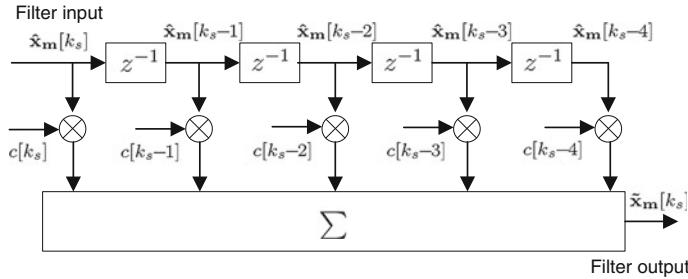


Fig. 4.8 A process of finite impulse response (FIR) low-pass filter with a window size of 5 after the position data compensation by the Bayesian particle filter

as shown in Fig. 4.5, short-term fluctuation or jitter, which is a high-frequency element, can be reduced and hence, smooth waveforms can be obtained.

In this section, we introduce a finite impulse response (FIR) low-pass filter that is suited for discrete-time implementation of haptic data. Since this FIR type of filter uses only finite input signals and does not require any feedback from filter output, it is also suited for any system that requires stability [14]. An example of FIR low-pass filter to process predicted or estimated position data is shown in Fig. 4.8. In this figure, after a finite window size is defined as 5, filter coefficients $c[k_s] \dots c[k_s - 4]$, which result in the amplitude of output waveform, need to be calculated based on the window size.

In general, a low-pass filter is used to improve the quality of signals by reducing noise or jitter. Since such filtering process is the process to compute a moving average, it inevitably introduces a group delay, which causes an additional delay to an overall teleoperation system. If the low-pass filters are used for the master and slave manipulators with window size of 5, the additional round trip time (RTT) delay is about 5 ms, given the local sampling rate of 1 kHz.

4.4 Experimental Study

4.4.1 Experimental Setup

The experimental setup to demonstrate a haptic teleoperation system based on the proposed methods in the transport and application layers is shown in Fig. 4.9. Two identical SensAble PHANToM Omni haptic devices were used as master and slave manipulators. The master manipulator was operated by the PC #1, and the slave manipulator was operated by the PC #2. A local area network (LAN) cable was directly connected between these two Window-based computers. The hardware and software specification for this experimental setup are described in Table 4.1.

We implemented that the master manipulator provided a single degree of freedom (DoF) positioning inputs to the slave manipulator through the LAN cable.

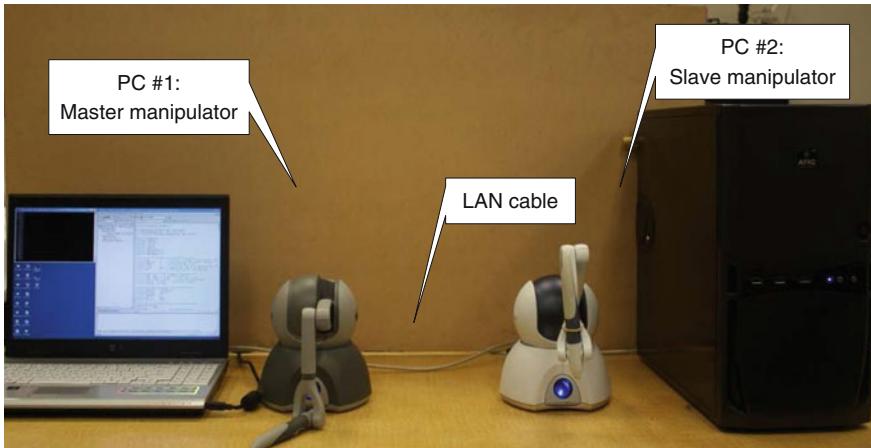


Fig. 4.9 Experimental setup: Two PHANToM Omni haptic devices, which are connected via a LAN cable, are used as master and slave manipulators. Spring force data attached to a stiff wall are generated from the slave manipulator and transmitted to the master manipulator

Table 4.1 Hardware and software specifications for the experimental setup

Hardware	PC #1: Dual core CPU 2.4 GHz, 2.0 GB RAM PC #2: Dual core CPU 2.8 GHz, 3.25 GB RAM
Operating systems	MS Windows XP Service Pack 3
Software	MS Visual C++ ver. 6.0
UDP socket library	Winsock ver. 2.0
Haptics library	OpenHaptics ver. 2.0

A conventional PD controller was implemented for robust control of slave positions. At the slave side, 1-DoF nonlinear spring was connected to a stiff wall in a virtual environment.¹ Based on the movements of the slave manipulator, spring force data were generated and transmitted to the master manipulator through the LAN cable. In the transport layer, the proposed M-UDP send and receive protocols were implemented at both PCs for interactive real-time transmission of position and force data. In the application layer, the Bayesian particle filter was implemented, and it was processed within a sampling period for real-time execution. The low-pass filter was also implemented for optional use. The sampling rate of position and force data was 1 kHz, and each sample was packetized to be sent via the M-UDP

¹As the current technology of virtual reality is closely related with teleoperation systems, real slave forces obtained from a force/torque sensor can be replaced by force data implemented in a virtual environment. The forces obtained from the sensor generally contain additive noise whereas the virtual forces do not contain any noise.

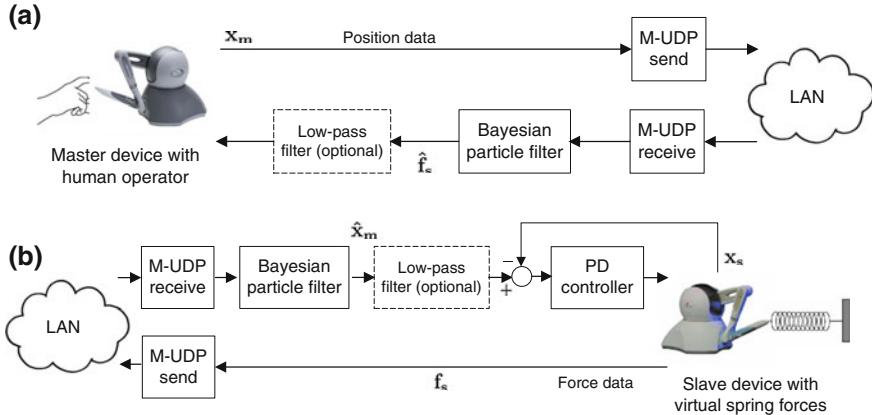


Fig. 4.10 Implementation of the proposed transport and application layers in the experimental setup: **a** The master manipulator, and **b** the slave manipulator

protocols. The implementation details for the master and slave manipulators are shown in Fig. 4.10.

4.4.2 Network Emulation

In order to demonstrate the performance of the proposed teleoperation system, we emulated the network delay between the master and the slave manipulators. In a real network environment, delay jitter inevitably occurs because sequential packets travel through multiple hops with different random queuing delays. An increasing delay is also typically observed when the traffic is congested over limited network bandwidth. When the network is congested, transmitting packets may not reach a destination, which results in a packet loss eventually. In this experimental study, we designed two different network delay models, including delay jitter, increasing delay, and packet loss. The details of one-way delay profiles are described in Table 4.2 and shown in Fig. 4.11. The delay profile #1 represents a relatively mild network condition, and the delay profile #2 represents a relatively severe network

Table 4.2 One-way delay profiles for the experimental study

	Delay profile #1	Delay profile #2
Sampling rate	100 Hz	200 Hz
Maximum delay	75 ms	150 ms
Minimum delay	50 ms	100 ms
Average delay	59.6 ms	122.0 ms
Standard deviation	7.6 ms	12.8 ms
Packet loss	5 %	10 %

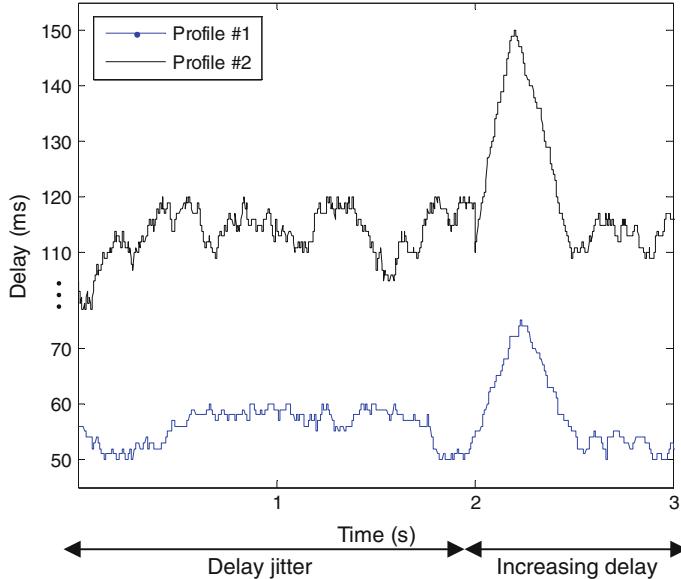


Fig. 4.11 One-way delay models including delay jitter and increasing delay

condition. These two delay profiles were emulated in real time in the uplink and downlink transmission in order to demonstrate the proposed teleoperation system.

For the delay profile #1, which represented a mild network condition, we modeled that the delay jitter varied as ± 5 ms when the one-way transmission delay between the master and the slave manipulators was 55 ms. The increasing delay was also modeled to reach up to 75 ms. In general, delay distributions over the network can be described by one of the stochastic models, such as normal, Pareto, or Pareto-normal distributions [15–17]. It is also known that a distributed delay sample depends on the previous delay sample rather than is totally random. Hence, for modeling the delay jitter and increasing delay, we designed that the current delay sample was distributed within 25 % of the previous delay sample based on a normal distribution. In addition, a new delay sample was updated at every 10 ms (100 Hz sampling rate) in order to represent a mild network condition. Since packet losses were inevitable over such delay model, the rate of packet losses was about 5 %. We used this delay model for both the uplink and the downlink transmission, and hence, the range of RTT delay was 100–150 ms with 119.2 ms average RTT delay. The standard deviation, which represented the degree of variations, was 7.6 ms.

In the case of delay profile #2, which represented a severe network condition, we modeled that the delay jitter varied as ± 10 ms, given the one-way transmission delay of 110 ms. The increasing delay was also modeled to reach up to 150 ms. For modeling these delay jitter and increasing delay, a new delay sample was updated at every 5 ms (200 Hz sampling rate) to represent relatively frequent variations, and it

was distributed within 25 % of the previous delay sample based on a normal distribution. The rate of packet losses was about 10 %, and the range of RTT delay was 200–300 ms with 244 ms average RTT delay. The standard deviation of this severe network condition was 12.8 ms.

The network emulation introduced in this section is often more preferred than the real Internet because one can understand the underlying performance of the proposed method in specific network conditions. Comparison with different types of methods can also be effectively conducted under a specific delay model [11]. Based on the experimental setup with the network emulation, we conducted (1) the preliminary experimental study to determine the number of particles and (2) the experimental study to demonstrate the performance of the proposed methods under the delay profiles.

4.4.3 Practical Convergence of Bayesian Particle Filter

As discussed in Sect. 3.4, the theoretical convergence of the Bayesian particle filter, which requires an infinitely large number of particles, is not tractable in practice. Hence, the number of particles needs to be chosen empirically while avoiding the computational overload for real-time execution. In this study, we conducted a preliminary experiment in order to determine such optimal number of particles in practice.

By conducting the preliminary experiment, the noise variances of position and force data were chosen as 1 mm and 0.001 N, respectively. Then, the Bayesian particle filter was used to estimate position data transmitted from the master manipulator through a 100-ms constant delay. The initial master position was not known to the slave manipulator so that the Bayesian particle filter was initialized from zero position. While different numbers of particles were used, we measured the times when the Bayesian particle filter converged to the master positions. The measured times with different numbers of particles from 50 to 1000 are shown in Table 4.3. When 200 particles were used, a convergence result is also shown in Fig. 4.12. As shown in this example, although the initial master position was unknown to the slave manipulator, the Bayesian particle filter was able to converge and track the master positions in 334 ms. According the results in Table 4.3, as the number of particles increases, the convergence time was reduced. Since we

Table 4.3 Number of particles versus convergence time

Number of particles	Convergence time (ms)
50	502
100	448
200	334
400	244
1000	242

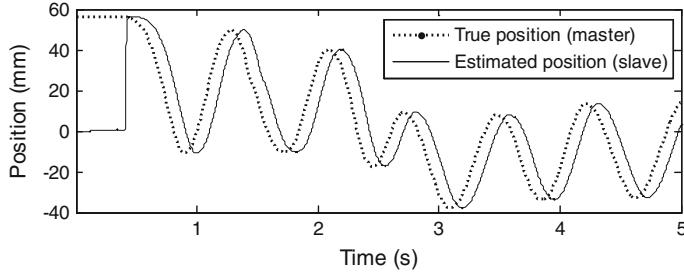


Fig. 4.12 Convergence of the Bayesian particle filter in practice with 200 number of particles. The Bayesian particle filter converged to the master positions in 334 ms excluding the constant transmission delay

observed that the convergence time was not significantly reduced with the number of particles larger than 400, we chose 400 particles as an optimal value, and it was used for the following experimental study in Sect. 4.4.4.

4.4.4 Experimental Results

In this section, we provide the performance results of the proposed teleoperation system under the delay profiles #1 and #2. After we conducted a bilateral teleoperation task in real time, we collected haptic data from the master and slave manipulators to represent the performance. Given the collected haptic data, we also measured the errors between the original haptic data and the compensated haptic data.

4.4.4.1 Performance Under Network Delay Profile #1

In the case of delay profile #1, which represented a mild network condition, the experimental results of 1-DoF position and force data are shown in Figs. 4.13 and 4.14, respectively. True master positions and true slave forces are shown in Figs. 4.13a and 4.14a, respectively. After true haptic data were transmitted through the delay profile #1, the delayed positions at the slave manipulator and the delayed forces at the master manipulator are shown in Figs. 4.13b and 4.14b, respectively. It was observed that the delayed positions and forces were impaired due to the network delay model. Position and force compensations using the estimation-based Bayesian particle filters are shown in Figs. 4.13c and 4.14c, respectively. In these cases, the M-UDP protocols were not used so that the Bayesian particle filters were performed without the information about unreliable network conditions. Finally, position and force compensations using the M-UDP-based Bayesian particle filters are shown in Figs. 4.13d and 4.14d, respectively. In this mild network condition

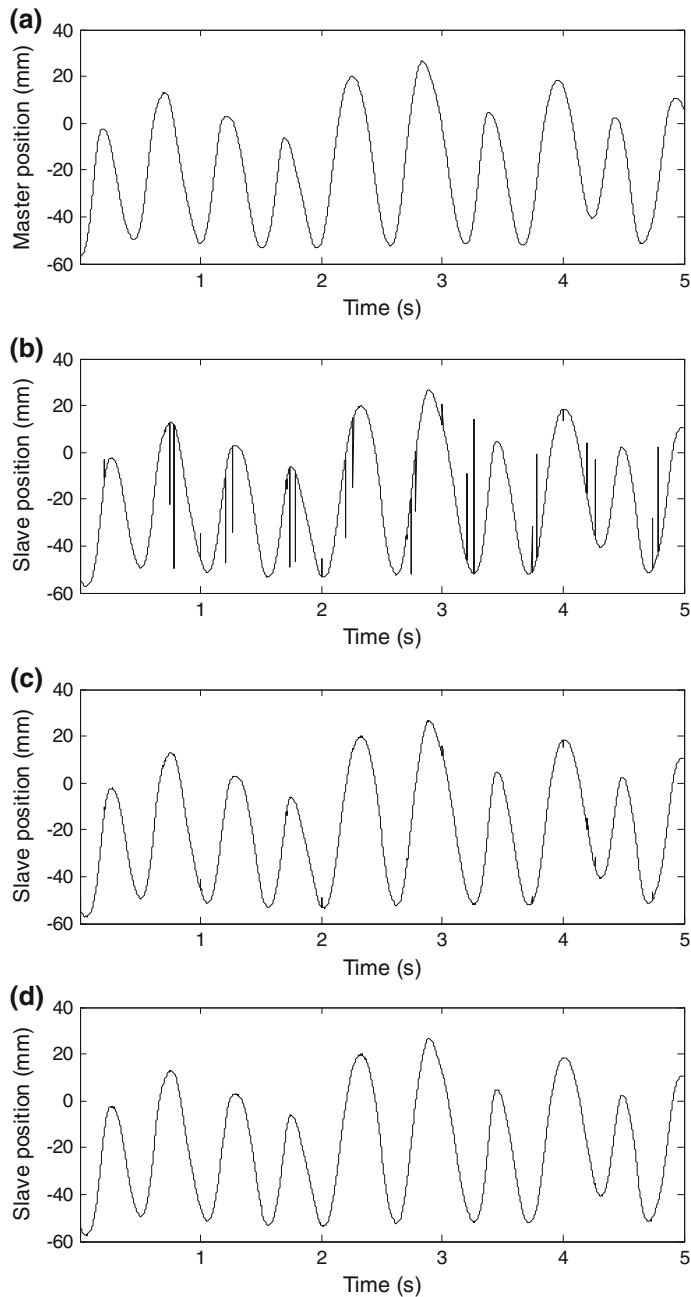


Fig. 4.13 Position compensation over delay profile #1: **a** Master positions, **b** slave positions through the delay model, **c** slave positions compensated by the estimation-based Bayesian particle filter, **d** slave positions compensated by the M-UDP-based Bayesian particle filter

case (profile #1), the estimation-based method without using the M-UDP compensated positions and forces quite successful, but resulting waveforms contained relatively small errors as shown in Figs. 4.13c and 4.14c. On the other hand, when the information about unreliable network conditions was available using the M-UDP, the quality of resulting waveforms was improved as shown in Figs. 4.13d and 4.14d.

4.4.4.2 Performance Under Network Delay Profile #2

In the case of delay profile #2, which represented a severe network condition, the experimental results of 1-DoF position and force data are shown in Figs. 4.15 and 4.16, respectively. True master positions and true slave forces are shown in Figs. 4.15a and 4.16a, respectively. After true haptic data were transmitted through the delay profile #2, the delayed positions at the slave manipulator and the delayed forces at the master manipulator are shown in Figs. 4.15b and 4.16b, respectively. Compared to profile #1, it was observed that the delayed position and force data were severely impaired because of the emulated network model. Position and force compensations using the estimation-based Bayesian particle filters are shown in Figs. 4.15c and 4.16c, respectively. As shown in these figures, position and force data were compensated with noticeable errors because the estimation-based Bayesian particle filters were performed without the information of observation availability. Finally, position and force compensations using the M-UDP-based Bayesian particle filters are shown in Figs. 4.15d and 4.16d, respectively. As shown in these figures, the quality of resulting waveforms was improved when the Bayesian particle filters were performed using the information about observation availability from the transport layer. Therefore, the proposed M-UDP-based Bayesian particle filters clearly outperformed the estimation-based Bayesian particle filters in this severe network condition.

4.4.4.3 Performance Results of Low-Pass Filter

After we obtained the performance results using the Bayesian particle filters, it was observed that relatively low jitter still remained depending on the prediction performance and the degree of unreliability over the network. This is because the Bayesian particle filter is a sample-based method that computes the states by averaging over a number of weighted samples (particles). In the case that such low jitter or prediction error caused an unstable operation, we additionally employed the low-pass filters at the master and slave manipulators.

In this study, we used the FIR low-pass filter with a window size of 20. Since the use of window size 20 at both the master and the slave manipulators already introduced additional RTT delay of 20 ms, we did not use the window size larger than 20. We applied the low-pass filter after the M-UDP-based Bayesian particle filter was performed. In the case of delay profile #2, the position compensation

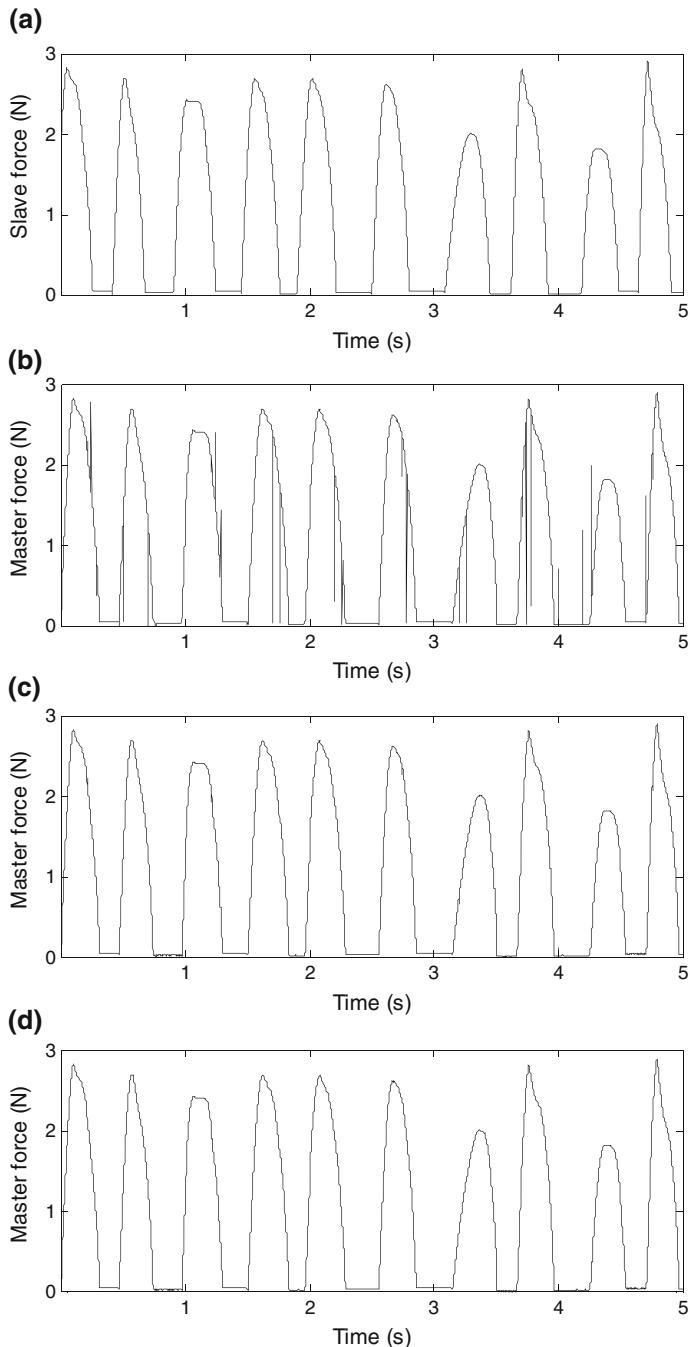


Fig. 4.14 Force compensation over delay profile #1: **a** Slave forces, **b** master forces through the delay model, **c** master forces compensated by the estimation-based Bayesian particle filter, and **d** master forces compensated by the M-UDP-based Bayesian particle filter

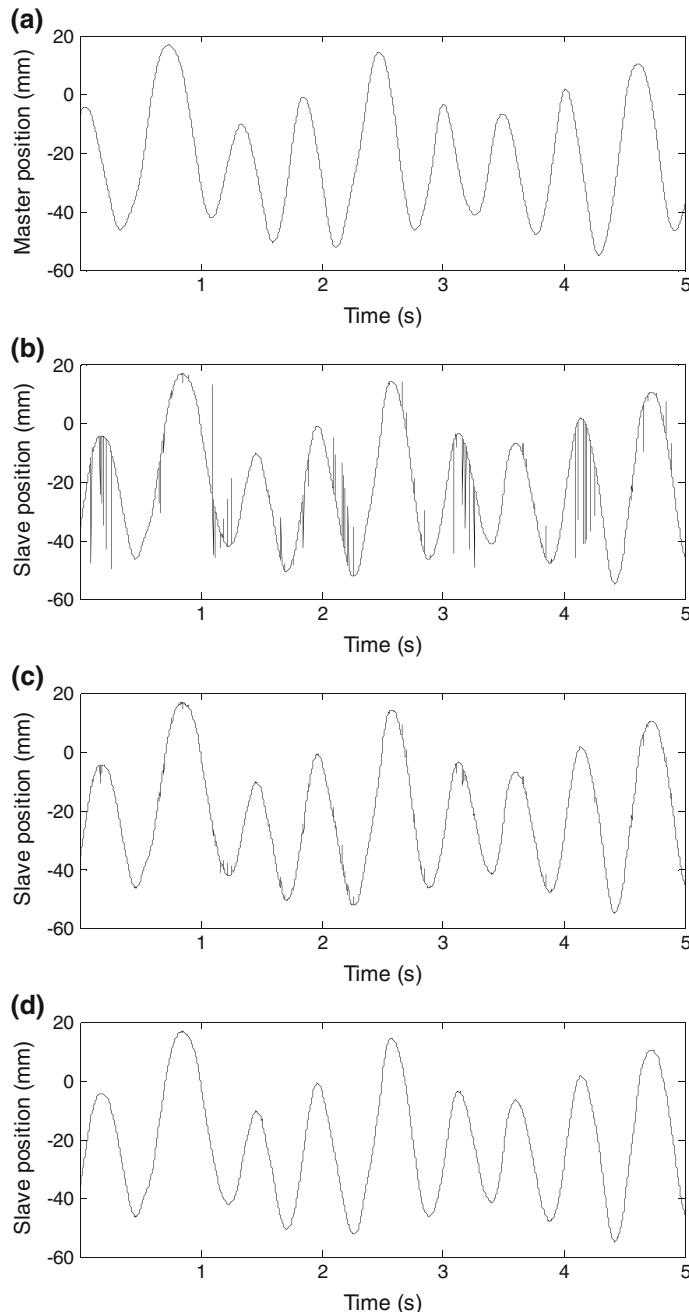


Fig. 4.15 Position compensation over the delay profile #2: **a** Master positions, **b** slave positions through the delay, **c** slave positions compensated by the estimation-based Bayesian particle filter, and **d** slave positions compensated by the M-UDP-based Bayesian particle filter

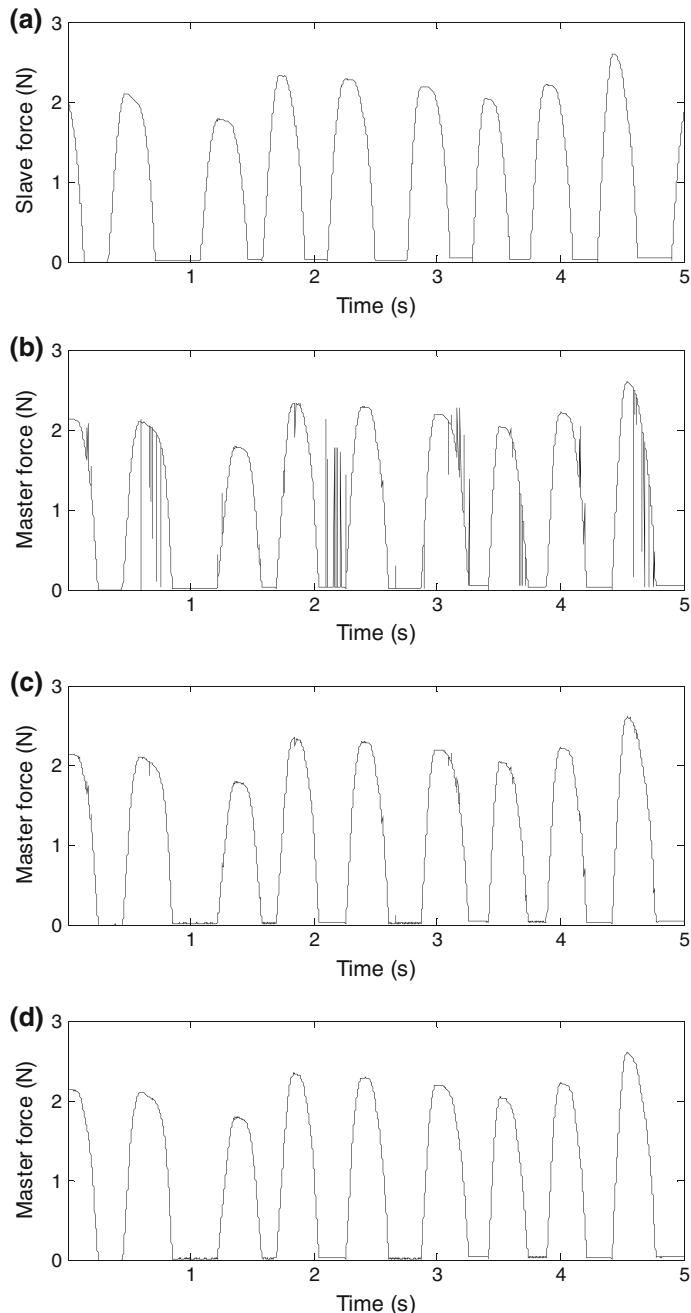


Fig. 4.16 Force compensation over the delay profile #2: **a** Slave forces, **b** master forces through the delay model, **c** master forces compensated by the estimation-based Bayesian particle filter, **d** master forces compensated by the M-UDP-based Bayesian particle filter

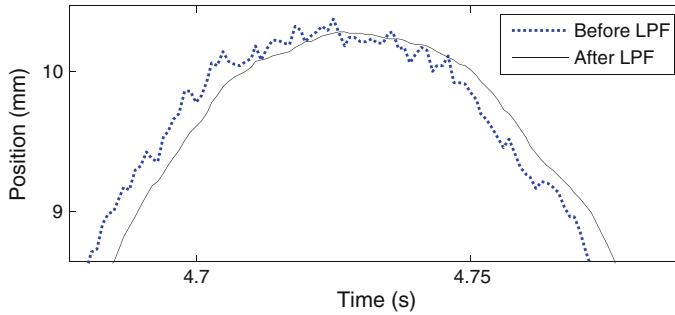


Fig. 4.17 1-DoF position data when low-pass filter is additionally used after the M-UDP-based Bayesian particle filter under the severe network condition (profile #2)

before and after the low-pass filter is shown in Fig. 4.17. As shown in this figure, the quality of resulting waveform was further improved with about 10 ms additional group delay. In this experiment, we also tested the low-pass filter when the Bayesian particle filter or the M-UDP was not performed. These additional performance results of the low-pass filter are shown in Appendix E.

4.4.4.4 Error Performance of the Proposed Methods

When the network delay profiles #1 and #2 were emulated, we measured the errors between the master and the slave haptic data. In the case of position data, we collected the following four data sets from the slave manipulator for a 10-s interval: (1) the impaired positions due to the delay profiles, (2) the compensated positions by the estimation-based Bayesian particle filter, (3) the compensated positions by the M-UDP-based Bayesian particle filter, and (4) the compensated positions by the M-UDP-based Bayesian particle filter and low-pass filter. After collecting the true positions from the master manipulator, we measured the mean squared errors (MSEs) between the master positions and the four slave positions. In the case of force data, the same experiment was conducted by collecting true slave force data and four master force data. Note that these data sets were collected after the Bayesian particle filters converged. In the case of delay profile #1, the MSE performances of position and force data are shown in Table 4.4. The MSE performances under the

Table 4.4 Mean squared error (MSE) performance of the proposed methods under the mild network condition (profile #1). Errors were measured for a 10-s operation

	Position error (mm)	Force error (mN)
Impaired data	9.26	22.58
PF	3.39	12.54
PF + M-UDP	3.29	12.46
PF + M-UDP + LPF	3.28	12.36

*PF Bayesian particle filter, LPF low-pass filter

Table 4.5 Mean squared error (MSE) performance of the proposed methods under the severe network condition (profile #2)

	Position error (mm)	Force error (mN)
Impaired data	11.19	46.35
PF	6.36	25.23
PF + M-UDP	6.26	25.16
PF + M-UDP + LPF	6.24	25.07

*PF: Bayesian particle filter, LPF: low-pass filter

Errors were measured for a 10-s operation

delay profile #2 are also shown in Table 4.5. As shown in these tables, the M-UDP-based Bayesian particle filter outperformed the estimation-based Bayesian particle filter. The low-pass filter also provided additional improvement.

4.5 Discussion

One of the concerns regarding the Bayesian particle filter may be the computational complexity. The Bayesian particle filter is a sequential Monte Carlo method that may introduce computational complexity while estimating or predicting a sample for a given number of particles. In the experimental setup, we also performed an experiment when 3-DoF position and force data were transmitted. Then, we verified that 3-DoF haptic data can be estimated and predicted in real time when the number of particles increased up to 500. However, given the resources of the experimental setup (i.e., CPU speeds of the PC #1 and PC #2), the computations of 3-DoF data did not work in real time when the number of particles exceeded 500. In order to reduce computational complexity of the Bayesian particle filter, haptic data compression methods, which down-sample transmitting haptic data, may be additionally used. The combination with the haptic data compression methods may provide not only the computational efficiency but also the further robustness over unreliable network conditions.

In this study, we do not consider any specific application of haptic teleoperation system, and hence, position and force models are generated by an arbitrary task. Since such arbitrary task involves highly uncertain states, a relatively large number of particles may be required for the Bayesian particle filter. On the other hand, when we consider a haptic teleoperation system that performs a specific task, such as telesurgery, position and force models can be task specific rather than totally random. These task-specific models generally have low uncertainty, which are more predictable. Therefore, in the task-specific haptic teleoperation system, a relatively small number of particles may be necessary when the proposed Bayesian particle filter is used.

References

1. J.F. Kurose, K.W. Ross, *Computer Networking: A Top-Down Approach* (Pearson Education, Boston, MA, 2008)
2. R.J. Anderson, M.W. Spong, Bilateral control of teleoperators with time delay. *IEEE Trans. Automatic Control* **34**(5), 494–501 (1989)
3. D. Lee, M. Spong, Passive bilateral teleoperation with constant time delay. *IEEE Trans. Robotics* **22**(2), 269–281 (2006)
4. B. Hannaford, J.-H. Ryu, Time-domain passivity control of haptic interfaces. *IEEE Trans. Robot. Autom.* **18**, 1–10 (2002)
5. S. Munir, W. Book, Internet-based teleoperation using wave variables with prediction. *IEEE/ASME Trans. Mechatron.* **7**(2), 124–133 (2002)
6. L. Ping, L. Wenjuan, S. Zengqi, Transport layer protocol reconfiguration for network-based robot control system, in *Networking, Sensing, and Control, 2005. Proceedings. 2005 IEEE*. Tucson, AZ, pp. 1049–1053 (2005)
7. Y. Uchimura, T. Yakoh, Bilateral robot system on the real-time network structure. *IEEE Trans. Ind. Electron.* **51**(5), 940–946 (2004)
8. H.H. King, B. Hannaford, J. Kammerl, E. Steinbach, Establishing multimodal telepresence sessions using the session initiation protocol and advanced haptic codecs, in *Proceedings IEEE Haptics Symposium*, Waltham, MA (2010), pp. 321–325
9. S. Lee, S. Moon, J.W. Kim, A network-adaptive transport scheme for haptic-based collaborative virtual environments, in *Proceedings of ACM SIGCOMM Workshop on Network and Sysystems: Support for Games*, New York, 2006. 8 pp
10. Z. Cen, M.W. Mutka, D. Zhu, N. Xi, Improved transport service for remote sensing and control over wireless networks, in *Proceedings IEEE International Conference on Mobile Adhoc and Sensor Systems*, Washington, DC, 2005, 8 pp
11. E.J. Rodríguez-Seda, D.J. Lee, M.W. Spong, Experimental comparison study of control architectures for bilateral teleoperators. *IEEE Trans. Robot.* **25**(6), 1304–1318 (2009)
12. R. Wirz, M. Ferre, R. Marín, J. Barrio, J. Claver, J. Ortego, Efficient transport protocol for networked haptics applications, in *Proceedings of The 6th International Conference on Haptics: Perception, Devices and Scenarios*, vol. 5024. Madrid, 2008, pp. 3–12
13. R. Rejaie, M. Handley, D. Estrin, RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet, in *Proceedings IEEE INFOCOM*, 1999, pp. 1337–1345
14. A.V. Oppenheim, R.W. Schafer, *Discrete-Time Signal Processing* (Prentice-Hall, Upper Saddle River, NJ, 1998)
15. V. Paxson, End-to-end Internet packet dynamics. *IEEE/ACM Trans. Networking* **7**(3), 277–292 (1999)
16. T. Mirfakhrai, S. Payandeh, On using delay prediction in controlling force reflecting teleoperation over the Internet. *Robotica* **23**(6), 809–813 (2005)
17. M.J. Coates, R.D. Nowak, Sequential Monte Carlo inference of internal delays in nonstationary communication networks. *IEEE Trans. Sig. Proc.* **50**, 366–376 (2002)

Chapter 5

Haptic Data Compression

Nomenclatures

d	Deadband threshold
n_1, n_2	Positive integers
f_c	Most recently transmitted force sample
\hat{x}_p, \hat{f}_p	Position and force samples computed by the prediction mode
\hat{x}_B, \hat{f}_B	Position and force samples computed by the bidirectional prediction mode
x_I, f_I	Intra-position and force samples
D_f	Downsampling factor
L	Number of data sequence
f_{\max}	Maximum possible force

In this chapter, we present signal-processing techniques that are suitable for haptic data compression in teleoperation systems. The presented haptic data compression methods mostly focus on downsampling methods, including the adaptive downsampling method introduced in [1, 2] as well as the fixed rate downsampling method. We propose the prediction methods and force data processing, which are based on the fixed rate downsampling. In the experimental study, we introduce an objective evaluation measure for haptic data, which may provide insight into the performance of various haptic data compression methods. A psychophysical evaluation is also conducted with a haptic interface in a virtual environment. Given the results of psychophysical and objective evaluations, we discuss how the objective evaluation measure can be interpreted with regard to the perceptual quality of haptic applications.

The remaining chapter is as follows. In Sect. 5.1, we present an overview of haptic data compression methods based on downsampling and their performance measures. In Sect. 5.2, we review the adaptive downsampling method, which is based on the deadband principle. The proposed signal-processing techniques based on the fixed rate downsampling are presented in Sect. 5.3. In Sect. 5.4, we provide extensive experimental studies, including the objective and psychophysical evaluations of the haptic data compression methods. Finally, we discuss the usefulness of the objective evaluation measure for haptic data in Sect. 5.5.

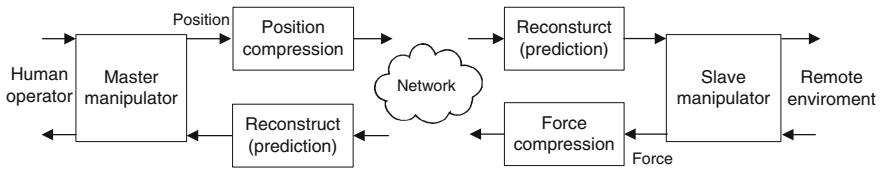


Fig. 5.1 Haptic data compression based on downsampling

5.1 Haptic Data Compression Based on Downsampling

In signal processing,¹ downsampling is the process that reduces the sampling rate in order to reduce the size of data or bit rate. For haptic data compression, position data in the uplink transmission are downsampled at the master manipulator. After the downsampled position data are transmitted, the original sampling rate (e.g., 1 kHz) is reconstructed at the slave manipulator. Force data in the downlink transmission are also downsampled at the slave manipulator and reconstructed at the master manipulator. In order to reduce the errors between the original haptic data and the reconstructed haptic data, a prediction method can be additionally used at the receiving manipulators. This bilateral process of haptic data compression is shown in Fig. 5.1. Downsampling can be performed with a fixed rate, in which the sampling rate is divided by an integer value. It can be also performed adaptively depending on the human perception [1, 2].

The performance of a haptic data compression method can be evaluated by two performance measures in general: The *compression ratio* is a fundamental measurement, which is evaluated by the ratio between the compressed data size and the uncompressed data size. In order to be processed through the network, it is commonly represented as a bit rate, such as bits per second (bps). When the reconstruction process is performed, the *quality measure* needs to be introduced for evaluating the perceptual quality of the reconstructed data compared to the uncompressed data. In the area of haptics, a psychophysical experimental study is mostly preferred in order to evaluate the perceptual quality [3, 4]. In this psychophysical evaluation, the proposed haptic data compression method is implemented in a teleoperation system or a haptic interface. While the system runs in real time, a number of participants conduct experiments to evaluate the proposed method by comparing it with the uncompressed haptic data. A psychophysical evaluation using a haptic interface in a virtual environment is shown in Fig. 5.2. In multimedia data compression, on the other hand, an objective measure is also widely used to evaluate the quality of the reconstructed data. In this objective evaluation, metric measures, such as signal-to-noise ratio (SNR) or peak SNR (PSNR), are used by computing the errors between the reconstructed data and the uncompressed data.

¹Basics of data compression methods are presented in Appendix F.

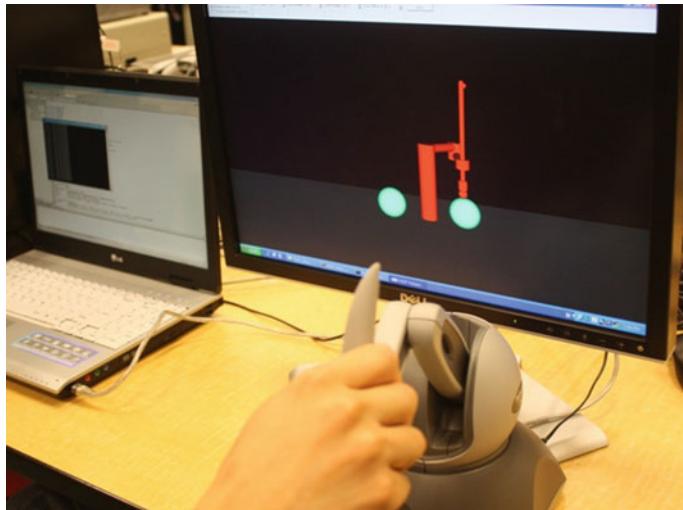


Fig. 5.2 A psychophysical evaluation using a haptic interface in a virtual environment

In this chapter, we demonstrate both the psychophysical and the objective evaluations, and then discuss the usefulness of the objective measures to evaluate the perceptual quality of haptic data.

5.2 Adaptive Downsampling (Deadband Method)

The adaptive downsampling method introduced in [1, 2] is based on the deadband principle that defines unnoticeable ranges of haptic data by human perception. Given a predefined *deadband threshold*, a sample within the threshold is discarded while a sample that exceeds the threshold is transmitted. In other words, a sample is chosen and transmitted over the network only if the absolute difference between the current sample and the most recently transmitted sample exceeds the deadband threshold. In the case of force data, a compression process from the slave manipulator is achieved by transmitting the current force sample $f[k]$ at the time instance k if the following condition holds:

$$|f[k] - f_c[k - n_1]| > d \quad (5.1)$$

where $f_c[k - n_1]$ is the most recently transmitted force sample at the time instance $k - n_1$, and n_1 is a positive integer with $k > n_1 > 0$. d is a predefined deadband threshold that may be represented in percentage. For example, when the deadband threshold is defined as 1 % and the most recently transmitted force sample is 1.0 N, the compression process selects only the next force sample that is higher than 1.01

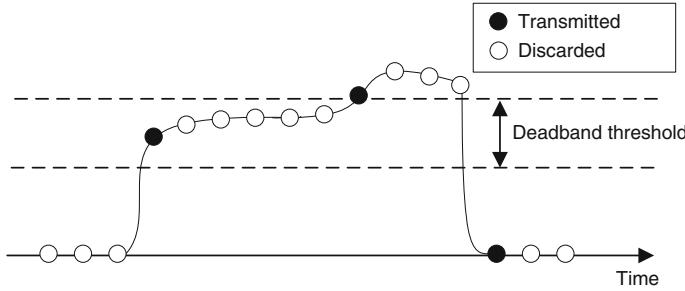


Fig. 5.3 1-degree of freedom (DoF) force data compression based on the adaptive downsampling method. *Black* samples are transmitted, and *white* samples within a deadband threshold are discarded

or lower than 0.99 N. After the selected force samples are transmitted, the reconstruction process is performed at the master manipulator. Once a force sample is received, it is resampled with the sampling rate of local clocks (e.g., 1 kHz) until a new force sample is updated. A 1-degree of freedom (DoF) example of deadband method to achieve a force data compression is shown in Fig. 5.3.

The deadband method can be combined with a prediction scheme for performance improvement both in the compression ratio and in the quality measure [5]. For the compression process of the prediction-based deadband method, a predicted force sample $\hat{f}[k]$ at the time instance k is computed by using the following linear predictor:

$$\hat{f}[k] = f_c[k - n_1] + \frac{f_c[k - n_1] - f_c[k - n_2]}{n_2 - n_1} \quad (5.2)$$

where $f_c[k - n_1]$ and $f_c[k - n_2]$ are the last two transmitted force samples at the time instances $k - n_1$ and $k - n_2$, respectively. n_1 and n_2 are positive integers with $k > n_2 > n_1 > 0$. The current force sample $f[k]$ is then a new transmitted sample only if the following condition holds:

$$|f[k] - \hat{f}[k]| > d. \quad (5.3)$$

In this prediction-based deadband method, the current force sample $f[k]$ is compared with the predicted force sample $\hat{f}[k]$ rather than compared with the most recently transmitted sample. For the reconstruction process at the master manipulator, the identical predictor that is used for the compression process (5.2) should be used. The predicted force samples are generated at the sampling rate of local clocks (e.g., 1 kHz) until a new force sample is received.

5.3 Haptic Data Compression Based on Fixed Rate Downsampling

In this section, we propose the haptic data compression method based on fixed rate downsampling. We present prediction methods that can be suitable for fixed rate downsampling and contact force data detection scheme for further improvement in performance. We also present the control architecture in order to address stability of the method that may introduce a processing delay.

5.3.1 Fixed Rate DownSampling

Fixed rate downsampling is one of the simple approaches to compress haptic position and force data. From the sending manipulators, the *downsampling factor*, which is an integer value greater than unity, is chosen in order to divide the original sampling rate (e.g., 1 kHz) by the integer value. After downsampled position and force data are transmitted, the original sampling rate is reconstructed at the receiving manipulators by the upsampling process. A 1-DoF example of a compression process based on fixed rate downsampling is shown in Fig. 5.4. In this example, the downsampling factor is chosen as 4. Hence, given the original sampling rate of 1 kHz, the sampling rate is reduced to 250 Hz. After the transmission, the up-sampling process is performed at the receiving manipulators to reconstruct 1 kHz.

5.3.2 Prediction Methods

In data compression, a prediction method, which is also called predictive coding, is widely used to improve the compression ratio and to reduce the errors between the

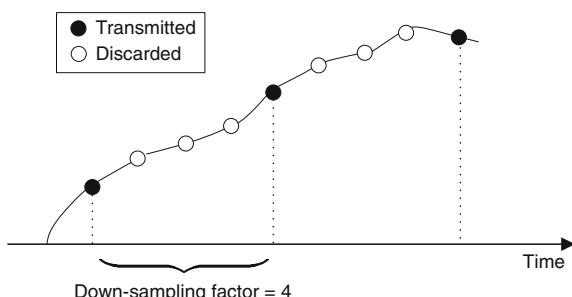


Fig. 5.4 1-DoF example of haptic data compression based on fixed rate downsampling. In this example, the downsampling factor is chosen as 4 so that the sampling rate is reduced to 250 Hz, given 1 kHz original sampling rate

uncompressed and the reconstructed data. When a downsampling method is used for compressing haptic data, a prediction method is performed with a reconstruction process at the receiving manipulator as shown in Fig. 5.1.

In this section, we present two prediction methods based on the fixed rate downsampling. As a compression process, samples are selected at a fixed rate from the sending manipulators. When the selected samples are transmitted to the receiving manipulators, the proposed prediction methods use the transmitted samples to compute predicted samples with the sampling rate of local clocks (e.g., 1 kHz). The prediction method using only past samples is called the *prediction mode*, whereas the prediction method using both past and next samples is called the *bidirectional prediction mode*. Before we present these prediction methods based on the fixed rate downsampling, we first define the following terms:

- *Intrasample (I-sample)*: A sample that is selected at a fixed rate from the sending manipulators and transmitted to the receiving manipulators is defined as *I-sample*. This refers the information within an original sample and not relative to any other samples.
- *Predicted sample (P-sample)*: When the prediction method is performed at the receiving manipulators, a predicted sample computed only using previous *I*-samples is defined as a *P*-sample.
- *Bidirectionally predicted sample (B-sample)*: A predicted sample computed by using both the previous *I*-sample and the next *I*-sample is defined as a *B*-sample.
- *Group of samples (GoS)*: A group of samples is determined by a downsampling factor. It consists of one *I*-sample and a number of *P*-samples or *B*-samples.

5.3.2.1 Prediction Mode

In the prediction mode, *P*-samples are computed using the last two transmitted *I*-samples. In the case of position data, the initial position sample needs to be transmitted first in order to compute the *P*-samples of the first GoS. Then, *I*-samples are chosen by the downsampling factor and then transmitted with a fixed rate. At the slave manipulator, the *P*-samples \hat{x}_P within the current GoS at the time instance of slave local clock k_s can be computed using the following equations:

$$\hat{x}_P[k_s + i] = x_I[k_s] + \frac{x_I[k_s] - x_I[k_s - n_1]}{n_1} \cdot \frac{i}{D_f} \quad (5.4)$$

for $i = 1, 2, \dots, D_f - 1$, where D_f is a downsampling factor. $x_I[k_s]$ is the *I*-sample within the current GoS and $x_I[k_s - n_1]$ is the *I*-sample in the previous GoS or the initial position sample. If $x_I[k_s - n_1]$ is the *I*-sample in the previous GoS, then $n_1 = 1$. If $x_I[k_s - n_1]$ is the initial position sample, then $n_1 = D_f$, which is the positive integer greater than unity.

A 1-DoF example of position data reconstruction process based on the prediction mode is shown in Fig. 5.5. In this example, after the initial position sample is

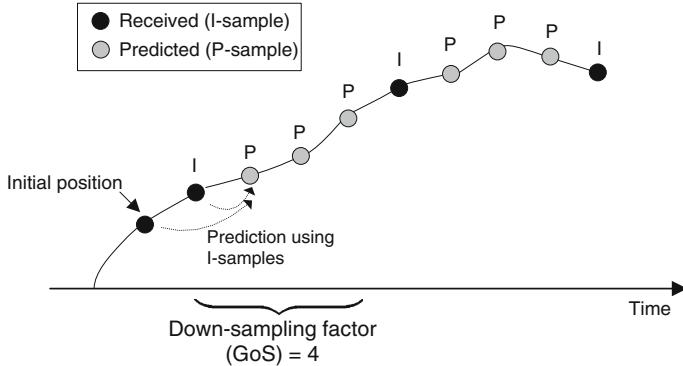


Fig. 5.5 1-DoF position data reconstruction process based on the prediction mode. In this example, with the downsampling factor of 4, a GoS consists of one *I*-sample and three *P*-samples

transmitted first, a downsampling factor is chosen as 4 so that a GoS consists of an *I*-sample and three *P*-samples. Since the downsampling factor is 4, the sampling rate is reduced to 250 Hz, given 1 kHz local sampling rate. As shown in the figure, the *P*-samples are computed using only the two most recently received *I*-samples at the slave manipulator.

The force data reconstruction process can be achieved by a similar process at the master manipulator. While the slave manipulator is in contact with the environment, *I*-samples are received with the initial force sample. The *P*-samples \hat{f}_P within the current GoS at the time instance of master local clock k is given by

$$\hat{f}_P[k+i] = f_I[k] + \frac{f_I[k] - f_I[k-n_2]}{n_2} \cdot \frac{i}{D_f} \quad (5.5)$$

for $i = 1, 2, \dots, D_f - 1$. $f_I[k]$ is the *I*-sample within the current GoS and $f_I[k-n_2]$ is the *I*-sample in the previous GoS or the initial sample of the contact force region. If $f_I[k-n_2]$ is the *I*-sample in the previous GoS, then $n_2 = 1$. If $f_I[k-n_2]$ is the initial force sample, then $n_2 = D_f$, which is the positive integer greater than unity.

A 1-DoF example of force data reconstruction process based on the prediction mode is shown in Fig. 5.6. In this example, after the initial force sample in contact force region is transmitted first, a downsampling factor is chosen as 3 so that a GoS consists of one *I*-sample and two *P*-samples. The sampling rate is reduced to 333 Hz, given 1 kHz local sampling rate.

5.3.2.2 Bidirectional Prediction Mode

In the bidirectional prediction mode, *B*-samples are computed using the previous and next *I*-samples. In the case of position data, the *I*-samples, which are chosen by the downsampling factor, are transmitted with a fixed rate. At the slave manipulator,

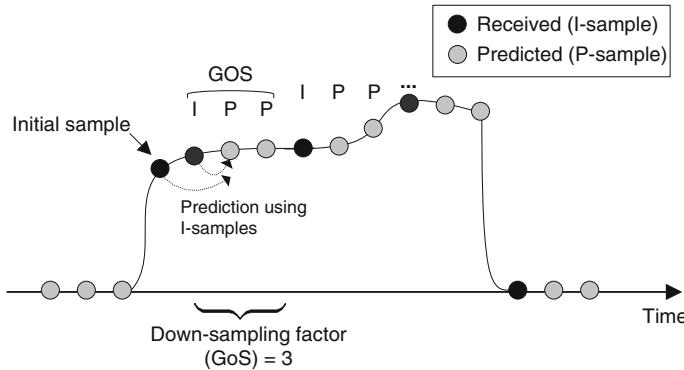


Fig. 5.6 1-DoF force data reconstruction process based on the prediction mode. In this example, with the downsampling factor of 3, a GoS consists of one *I*-sample and two *P*-samples in a contact force region

the *B*-samples \hat{x}_B within the current GoS at the time instance k_s can be computed using the following equations:

$$\hat{x}_B[k_s + i] = x_I[k_s] + \frac{x_I[k_s + D_f] - x_I[k_s]}{D_f} i \quad (5.6)$$

for $i = 1, 2, \dots, D_f - 1$. $x_I[k_s]$ is the *I*-sample within the current GoS and $x_I[k_s + D_f]$ is the *I*-sample in the next GoS.

A 1-DoF example of position data reconstruction process based on the bidirectional prediction mode is shown in Fig. 5.7. In this example, a downsampling factor is chosen as 5, and hence, a GoS consists of one *I*-sample and four *B*-samples. In this case, the sampling rate is reduced to 200 Hz, given 1 kHz local sampling rate. As shown in the figure, the *B*-samples are computed by using the previous and next *I*-samples at the slave manipulator.

In the case of force data, the reconstruction process can be performed using a similar process at the master manipulator. In a contact force region, the *B*-samples \hat{f}_B within the current GoS can be computed as

$$\hat{f}_B[k + i] = f_I[k] + \frac{f_I[k + D_f] - f_I[k]}{D_f} i \quad (5.7)$$

for $i = 1, 2, \dots, D_f - 1$. $f_I[k]$ is the *I*-sample within the current GoS and $f_I[k + D_f]$ is the *I*-sample in the next GoS. A 1-DoF example of force data reconstruction process based on the bidirectional prediction mode is also shown in Fig. 5.8. In this example, a downsampling factor is chosen as 4 so that a GoS consists of one *I*-sample and three *B*-samples. The sampling rate is reduced to 250 Hz, given 1 kHz local sampling rate.

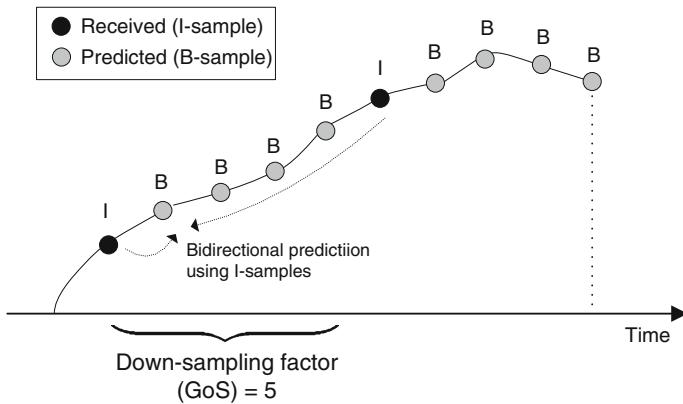


Fig. 5.7 1-DoF position data reconstruction process based on the bidirectional prediction mode. In this example, with the downsampling factor of 5, a GoS consists of one *I*-sample, and four *B*-samples

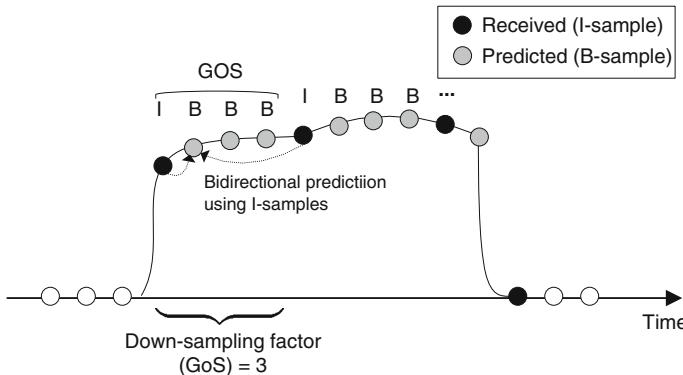


Fig. 5.8 1-DoF force data reconstruction process based on the bidirectional prediction mode. In this example, with the downsampling factor of 4, a GoS consists of one *I*-sample and three *B*-samples

5.3.2.3 Stability of Bidirectional Prediction Mode

In the prediction mode, *P*-samples are obtained by using only the previous *I*-samples, and therefore, the reconstruction process can be performed without introducing any processing delay. However, in the bidirectional prediction mode, *B*-samples are obtained by using the previous and next *I*-samples. Therefore, the receiving manipulators should allow a processing delay until the next *I*-sample is received. For instance, when a downsampling factor is chosen as 4, the processing delay at the receiving manipulator is 4 ms, given 1 kHz local sampling rate.

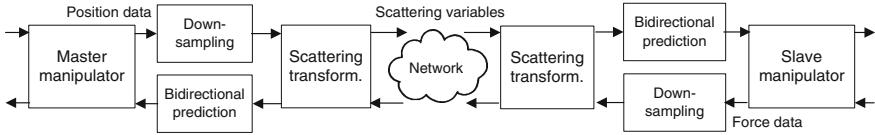


Fig. 5.9 Haptic data compression method based on the scattering transformation

Since such processing delay introduces a constant delay between the master and the slave manipulators, it may cause instability of overall teleoperation systems as presented in Chap. 3 and Appendix C.2. Hence, in the case of bidirectional prediction mode, the passivity-based control methods, such as the scattering transformation [6], wave variables [7], and passivity observer/passivity controller [8], can be additionally used. The passivity-based scattering transformation, which is combined with the proposed haptic data compression method based on the bidirectional mode, is shown in Fig. 5.9. As shown in this figure, downsampled haptic data are transformed to scattering variables from the sending manipulators. At the receiving manipulators, the scattering variables are transformed to haptic data for the reconstruction processes using the bidirectional mode.

5.3.3 Contact Force Detection

In the case of force data, contact force data are generated only if the slave manipulator is in contact with a remote environment or a virtual environment. If the slave manipulator is not in contact with the environment, noncontact force data may be detected or it is not necessary to generate any force data depending on how the slave manipulator is designed. In either case, when the proposed fixed rate downsampling method is used, it needs to be performed only when contact force data are generated in order to compress force data efficiently and to reduce prediction errors between the uncompressed and the reconstructed force data. In the proposed haptic data compression method based on the fixed rate downsampling, we additionally use a contact force detection method, and hence, force data are downsampled in a fixed rate only when the slave manipulator is in contact with the environment. When the slave manipulator is not in contact with the environment, the fixed rate downsampling is not necessarily performed.

The contact force detector can also be useful for the adaptive downsampling method. When the deadband method is used, a noncontact force sample received at the master manipulator may not be zero depending on a choice of deadband threshold. Such example is shown in Fig. 5.10. As shown in this figure, the reconstructed force data at the noncontact force region have a constant value, which may be perceived by the human operator. This problem often occurs when the difference between the last transmitted sample at the contact force region and a zero sample at the noncontact force region does not exceed the deadband threshold.

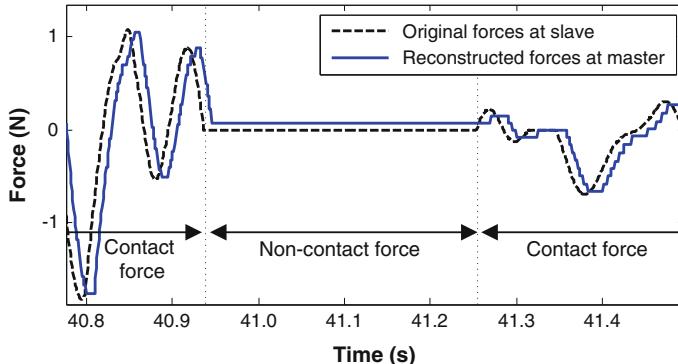


Fig. 5.10 Force data generated from the slave manipulator and reconstructed force data at the master manipulator. Force data are compressed by the deadband method with 7.5 % deadband threshold

In the performance aspect, this problem may not affect the compression ratio. However, the errors between the original and the reconstructed force data may increase, which degrade a quality measurement.

5.4 Experimental Study

In order to demonstrate the proposed haptic data compression method, we conducted extensive experimental studies. The experimental setup using a haptic interface in a virtual environment is presented in Sect. 5.4.1. In the first part of the experiment, presented in Sect. 5.4.2, we provide the objective evaluation results of the proposed haptic data compression method and the deadband method. In the second part of the experiment, we provide the psychophysical evaluation results of the proposed and deadband methods in Sect. 5.4.3. Based on the objective and psychophysical results in this section, the usefulness of the objective evaluation measurement for haptic data is discussed in Sect. 5.5.

5.4.1 Experimental Setup

The experimental setup to demonstrate the proposed haptic data compression method is shown in Fig. 5.11. A SensAble PHANTOM Omni haptic device was used as a master manipulator, which was operated by the PC #1. As a slave manipulator, a three-dimensional (3D) virtual manipulator was implemented in the PC #2, and a local area network (LAN) cable was directly connected between these two Windows-based PCs. We implemented that the master manipulator provided

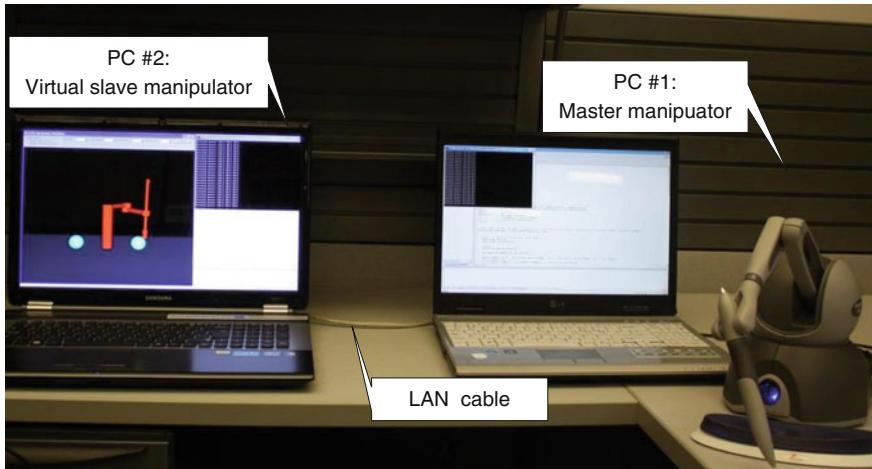


Fig. 5.11 Experimental setup: A PHANToM Omni haptic device was used as a master manipulator, and a virtual slave manipulator was implemented. 3-DoF position and force data were bilaterally transmitted through the local area network (LAN) cable

3-DoF positioning inputs to the virtual slave manipulator through the LAN cable. While the slave manipulator interacted with the virtual environment, 3-DoF force data were computed and fed back into the master manipulator. In order to transmit position and force data in real time, the standard user datagram protocol (UDP) send and receive protocols were implemented in the two PCs. The sampling rate of position and force data was 1 kHz. 3-DoF position and force samples at every sampling period were packetized and transmitted through the UDP protocols. The hardware and software specifications of the experimental setup are shown in Table 5.1. In addition, the details of implementing the virtual slave manipulator, including the kinematic solution, 3D graphic programming, and haptic programming, are described in Appendix G.

Table 5.1 Hardware and software specification for the experimental setup

Hardware	PC #1: Dual core CPU 2.4 GHz, 2.0 GB RAM PC #2: Quad core CPU 2.0 GHz, 3.4 GB RAM
Operating system	MS Windows XP Service Pack 3
Software	MS Visual C++ ver. 6.0
UDP socket library	Winsock ver. 2.0
Graphic library	OpenGL ver. 2.1
Haptics library	OpenHaptics ver. 2.0

5.4.2 Objective Evaluation

5.4.2.1 Performance Measure

For the objective evaluation of the proposed haptic data compression method and the deadband method, we used the operational rate-distortion measure. In this performance measure, the horizontal axis on a graph typically denotes bit rates after the compression process. In the vertical axis, distortion measurements, such as mean squared errors (MSEs) or peak signal-to-noise ratios (PSNRs) are shown by comparing the reconstructed data with the uncompressed data. In the case of force data, the MSE can be obtained by averaging distortions over a number of data sequence such that

$$\text{MSE} = \frac{1}{L} \sum_{k=1}^L (f[k] - \tilde{f}[k])^2 \quad (5.8)$$

where $f[k]$ is an uncompressed force sample from the slave manipulator, and $\tilde{f}[k]$ is a reconstructed force sample at the master manipulator. The differences are averaged over L number of samples. Then, the PSNR, which is described in decibel (dB), can be computed by normalizing the MSE to produce dimensionless quantity such that [9]

$$\text{PSNR} = 10 \log_{10} \left(\frac{f_{\max}^2}{D} \right), \quad (5.9)$$

where f_{\max} is the maximum possible force value that the slave manipulator can exert. Since the PSNR takes an inverse of the MSE, it is called a quality measurement rather than a distortion measurement.

5.4.2.2 Position and Force Data Sets for Objective Evaluation

In order to obtain the operational rate-distortion measures, we collected 3-DoF position and force data sets for a 100-s interval, while a human operator conducted a teleoperation task in the virtual environment. Then, the proposed and deadband methods were processed using the collected data sets. The 1-DoF sample plots from the collected data sets are shown in Fig. 5.12. After the compression and reconstruction processes were performed by the proposed and the deadband methods, PSNRs and bit rates were measured by averaging over the 100-s interval. In order to compute the PSNRs, the data ranges of 3-DoF position and force data were measured as shown in Table 5.2. The bit rates of uncompressed haptic data are also shown in this table.

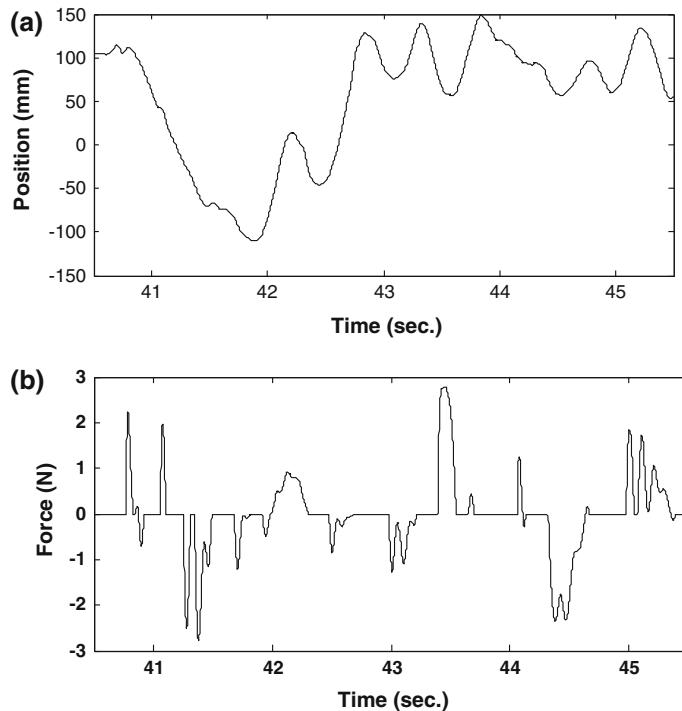


Fig. 5.12 1-DoF (x -axis) haptic **a** position and **b** force data captured from the experimental setup over a 5-s interval

Table 5.2 Data ranges and bit rates of 3-DoF position and force data measured from the experimental setup

	Position data	Force data
Data ranges	X-axis: (-210, +210 mm) Y-axis: (-105, +145 mm) Z-axis: (-100, +120 mm)	(-3.3, +3.3 N)
Bit rates	239.0 kbps	270.0 kbps

5.4.2.3 Objective Evaluation Results

The operational rate-distortion measures of the proposed position data compression methods, including (1) the downsampling method, (2) the prediction mode, and (3) the bidirectional prediction mode, are shown in Fig. 5.13. In the case of downsampling method, which was evaluated for the 2–5 downsampling factors, the reconstructed data were obtained without using any prediction scheme, and hence, the PSNRs were significantly lower than the prediction and bidirectional prediction modes. When the downsampling factors were 2–10, the PSNRs of the bidirectional prediction mode were better than those of the prediction mode by 4–10 dB. This was because the prediction strategy using the previous and next I -samples provided

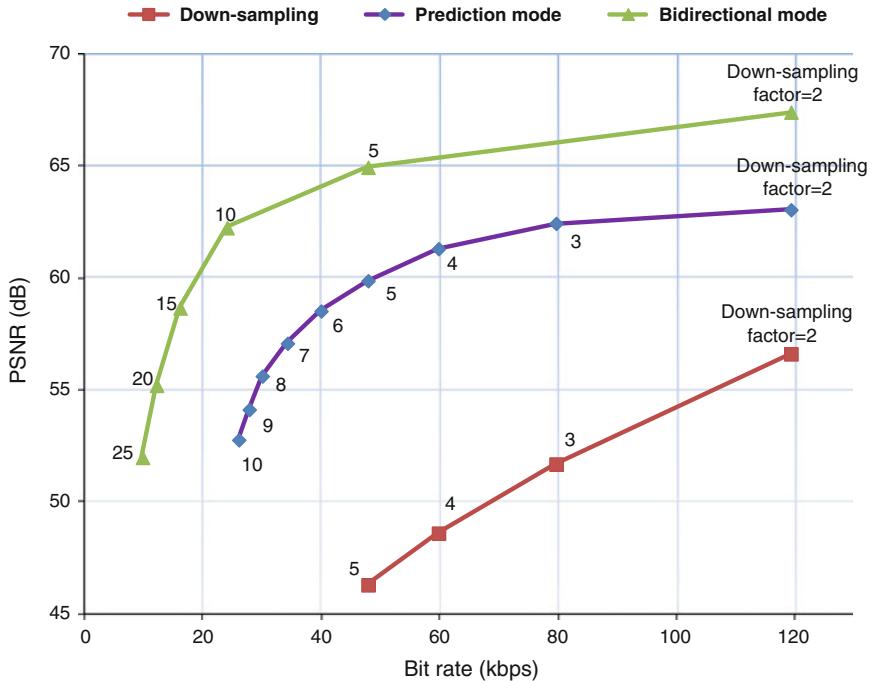


Fig. 5.13 Operational rate-distortion measures of the position data compression methods based on fixed rate downsampling

better quality than the prediction using only the previous I -samples. The bit rates at each downsampling factor were about the same for all the three methods, since all the compression methods were based on fixed rate downsampling.

The operational rate-distortion measures of the proposed force data compression methods, including (1) the downsampling method, (2) the prediction mode, and (3) the bidirectional prediction mode, are shown in Fig. 5.14. Similar to the position data compression, the downsampling method, which did not use any prediction scheme, provided significantly lower PSNRs compared to the prediction and bidirectional prediction modes. The PSNRs of the bidirectional prediction mode were better than those of the prediction mode by 4–10 dB. In addition, the bit rates at each downsampling factor were about the same for all the three methods.

In the case of force data, the deadband methods, which are the adaptive downsamples based on the human perception [1, 2], were also evaluated by the objective performance measure. In this experimental study, we implemented (1) the deadband method and (2) the prediction-based deadband method. The operational rate-distortion measures of these deadband methods are shown in Fig. 5.15. The

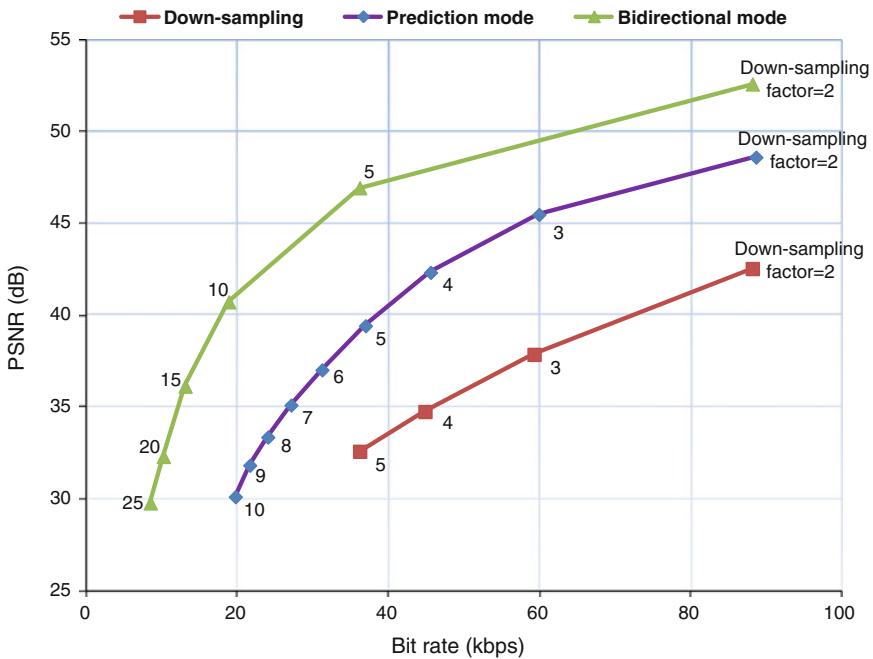


Fig. 5.14 Operational rate-distortion measures of the force data compression methods based on fixed rate downsampling

PSNRs and bit rates were obtained when the deadband thresholds varied between 2 and 15 %. According to the results in Fig. 5.15, the use of the prediction scheme provided the improved compression rates by about 10–30 kbps. However, the PSNRs were not significantly improved by using the prediction scheme since the deadband thresholds basically represented the errors between the current force value and the future force value.

Since these deadband methods are based on adaptive downsampling, the amount of data reduction highly depends on the variations in force data. In other words, if generated force values are implicated in many constant intervals throughout the experiment, the bit rates can be significantly reduced. Similarly, different PSNRs may be obtained depending on different teleoperation tasks. Due to this reason, a direct comparison between the adaptive sampling and the fixed rate sampling is not shown in this section. Nevertheless, given the collected force data set, we could observe that the bidirectional prediction mode performed better than any other methods according to the results from Figs. 5.14 and 5.15. Then, the prediction-based deadband method outperformed the prediction mode of the fixed rate downsampling, and the prediction mode outperformed the deadband method.

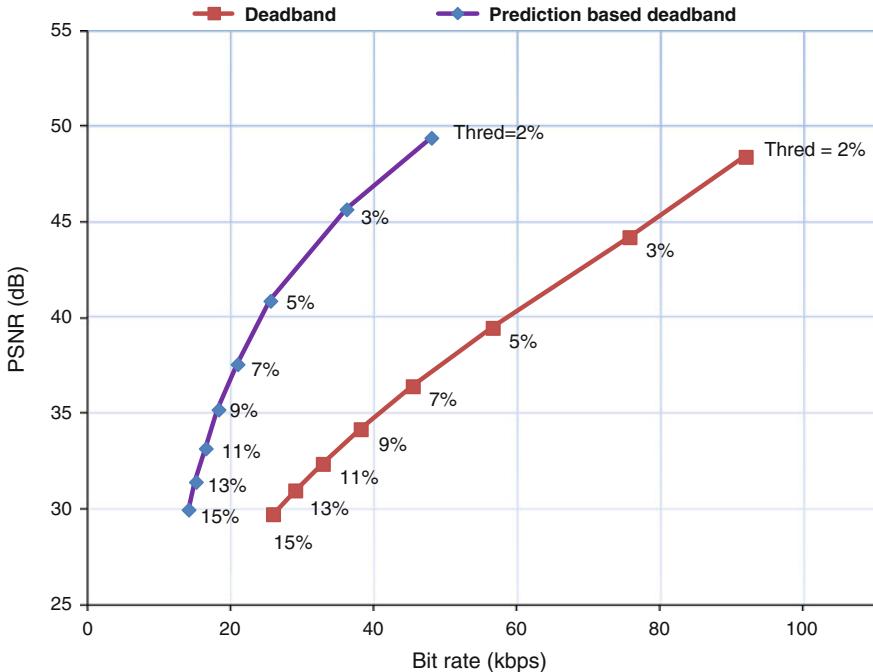


Fig. 5.15 Operational rate-distortion measures of the adaptive downsampling methods (deadband and prediction-based deadband) in the case of force data

5.4.3 Psychophysical Evaluation

5.4.3.1 Virtual Workspace for Pairwise Comparison

Based on the 3D virtual workspace of the experimental setup in Sect. 5.4.1, we performed the psychophysical evaluation for the proposed haptic data compression method and the deadband method. As a slave manipulator in the PC #2, we designed a 3D selective compliant assembly robot arm model, which consists of three rotational joints and one sliding joint in the virtual workspace. Two spherical objects were designed and implemented on the virtual plane, and hence, a human operator was able to make contact and slide on the surface of the objects by manipulating the end effector of the slave manipulator. The virtual workspace for the psychophysical evaluation is shown in Fig. 5.16. As shown in the figure, one of the spherical objects was represented by the compressed haptic data, whereas the other one was the uncompressed haptic data. Hence, using the two spherical objects, we conducted a pairwise comparison that determines any perceptual difference between the compressed haptic data and the uncompressed haptic data within a single virtual workspace [4].

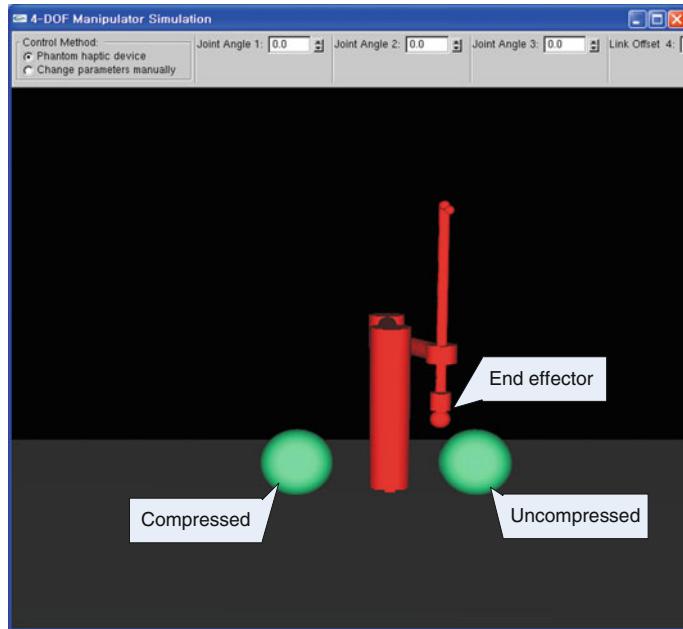


Fig. 5.16 A pairwise comparison for psychophysical evaluation of haptic compression methods

5.4.3.2 Participants

There were 12 participants who conducted this psychophysical evaluation. Two were female and 10 were male. All were between the ages of 20 and 34, and all were currently undergraduate or graduate students at Simon Fraser University (SFU) in the School of Engineering Science. All were right handed, and they had no more than trivial experiences with haptic interfaces. The participants did not have any neurological condition or physical disability, which may affect the evaluation results.

5.4.3.3 Psychophysical Evaluation Procedure

While compressed and uncompressed haptic data were represented in the two spherical objects and transmitted through the LAN cable, the participants were asked to conduct a teleoperation task in the virtual workspace. The virtual teleoperation task includes making contact with the spherical objects and sliding on the surface of the objects as described in Fig. 5.17. Before the evaluation, a practice session was provided so that the participants could conduct the evaluation proficiently in the given amount of time.

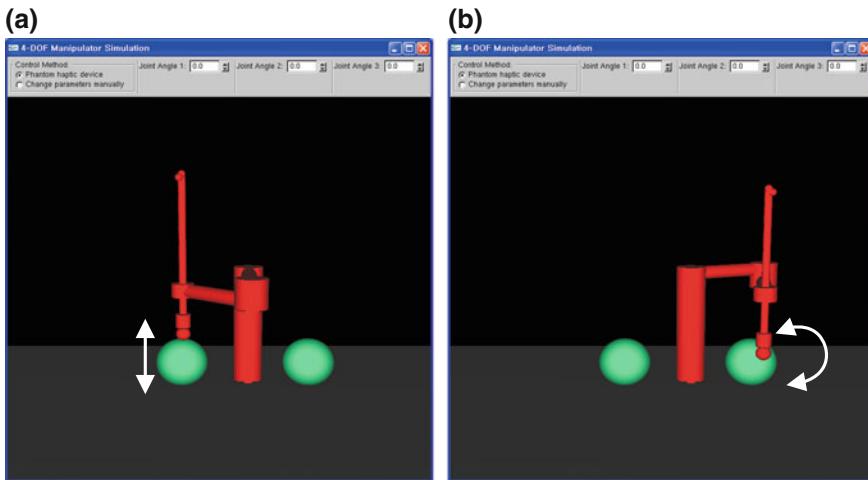


Fig. 5.17 Psychophysical tasks in the virtual workspace: **a** making contact with the spherical objects and **b** sliding on the surface of the objects to feel continuous 3-DoF force data

For each haptic data compression method, the participants conducted a pairwise comparison, while the parameters such as the downsampling factor and deadband threshold were randomly changed. Two identical spherical objects were also randomly provided, but such information about the parameters was not given to the participants. For each evaluation, the participants were allowed to conduct the pairwise comparison for 10 s. After 10 s, they had to determine whether the haptic sensation of two spherical objects is different or not. Based on the answers from the participants, “yes” was considered that a compression method caused sufficient artifacts or degraded perceptual transparency compared to the uncompressed haptic data.

The time limitation was set for each evaluation because the participants might consider other factors to make a decision if too much time was given for a single evaluation. For instance, while the participants made contact with the spherical objects, they might feel a small vibration from the haptic device, which might affect the evaluation result. They might also be unnecessarily suspicious such that they felt small difference between the two spherical objects even though two identical objects were given. Another reason why we set the time limitation was because we planned an evaluation to be completed within a reasonable amount of time before the participants with fatigue might result in degradation of performance.

5.4.3.4 Psychophysical Evaluation Results

We first evaluated the position data compression methods, including (1) the downsampling method, (2) the prediction mode, and (3) the bidirectional prediction

mode. Given the sampling rate of 1 kHz, position data were downsampled by a fixed rate from the master manipulator. The reconstruction process was performed in real time at the virtual slave manipulator by using the (1) up-sampling, (2) prediction, and (3) bidirectional prediction methods. The psychophysical evaluation results of the three position data compression methods with different downsampling factors are shown in Table 5.3.

As shown in Table 5.3, in the case of downsampling method, all 12 participants did not detect any artifacts compared to the uncompressed position data when the downsampling factor was 2, which corresponded to the sampling rate of 500 Hz. However, all the participants felt artifacts when the downsampling factor was larger than 4 (250 Hz). In the case of prediction mode, the participants started to feel the difference from the downsampling factor of 6 (167 Hz), and all the participants felt artifacts with the downsampling factor larger than 8 (125 Hz). Finally, in the case of bidirectional prediction mode, all the participants did not feel the difference up to the downsampling factor of 15 (67 Hz), and all they detected artifacts with the downsampling factor larger than 19 (53 Hz). In this bidirectional mode, as all the participant did not feel any difference up to the downsampling factor of 15, they did not perceive any transparency or stability impairment up to this downsampling factor.

Table 5.3 Psychophysical evaluation results of the position data compression methods based on fixed rate downsampling

Downsampling factor (sample rate)	Downsampling (%)	Prediction mode (%)	Bidirectional mode (%)
2 (500 Hz)	100	100	100
3	83	100	100
4	25	100	100
5 (200 Hz)	0	100	100
6	0	75	100
7	0	33	100
8	0	8	100
9	0	0	100
10 (100 Hz)	0	0	100
11	0	0	100
12	0	0	100
13	0	0	100
14	0	0	100
15	0	0	100
16	0	0	83
17	0	0	75
18	0	0	50
19	0	0	33
20 (50 Hz)	0	0	0

Number of participants who did not detect artifacts was represented in percentage

For the second psychophysical experiment, we evaluated the force data compression methods, including (1) the downsampling method, (2) the prediction mode, (3) the bidirectional prediction mode, (4) the deadband method, and (5) the prediction-based deadband method. For each method, the participants were asked whether they felt any difference between the compressed and the uncompressed force data by conducting the pairwise comparison. The psychophysical evaluation results for the first three methods based on fixed rate downsampling are shown in Table 5.4. In the case of downsampling method, 92 % of participants did not detect any artifacts at the downsampling factor of 2 (500 Hz), and all the participants felt artifacts with the downsampling factor larger than 4 (250 Hz). In the case of prediction mode, the participants started to feel the difference from the downsampling factor of 6 (167 Hz), and all the participants felt artifacts with the downsampling factor larger than 8 (125 Hz). In the case of bidirectional prediction mode, all the participants did not feel the difference up to the downsampling factor of 15 (67 Hz), and all they detected artifacts with the downsampling factor larger than 19 (53 Hz). In this bidirectional mode, as all the participants did not feel any difference up to the downsampling factor of 15, they did not perceive any transparency or stability impairment up to this downsampling factor.

Table 5.4 Psychophysical evaluation results of the force data compression methods based on fixed rate downsampling

Downsampling factor (sample rate)	Downsampling (%)	Prediction mode (%)	Bidirectional mode (%)
2 (500 Hz)	92	100	100
3	33	100	100
4	0	100	100
5 (200 Hz)	0	100	100
6	0	100	100
7	0	75	100
8	0	8	100
9	0	0	100
10 (100 Hz)	0	0	100
11	0	0	100
12	0	0	100
13	0	0	100
14	0	0	100
15	0	0	100
16	0	0	92
17	0	0	67
18	0	0	33
19	0	0	17
20 (50 Hz)	0	0	0

Number of participants who did not detect artifacts was represented in percentage

Table 5.5 Psychophysical evaluation results of the force data compression methods based on adaptive downsampling

Deadband threshold (%)	Deadband (%)	Prediction-based deadband (%)
2	100	100
3	58	100
4	17	100
5	0	100
6	0	100
7	0	100
8	0	100
9	0	75
10	0	58
11	0	42
12	0	25
13	0	0
14	0	0

Number of participants who did not detect artifacts was represented in percentage

In the case of force data compression, we also implemented the deadband methods in the experimental setup and performed the psychophysical evaluation in real-time. The evaluation results of (1) the deadband method and (2) the prediction-based deadband method are shown in Table 5.5. In the case of deadband method, which did not use any prediction scheme, some participants started to feel the difference even from the deadband threshold of 3 %. All the participants detected artifacts with the deadband threshold larger than 4 %. In the case of prediction-based deadband method, all the participants did not feel any difference up to the deadband threshold of 8 %, and all they felt artifacts with the deadband threshold larger than 12 %.

5.5 Discussion: Psychophysical Evaluation Results Represented in PSNR

In Sect. 5.4.2, we used the PSNR as a quality measurement of haptic data compression methods. Because haptic teleoperation systems or haptic interfaces are applications based on the sense of touch, the psychophysical evaluation, which was presented in Sect. 5.4.3, is a preferred evaluation method in general. Nevertheless, the PSNR measurement is a widely used objective quality measurement for other perceptual applications, such as video and audio signals because comparisons with other different types of compression methods can be efficiently shown in a single graph with ease of computations. In this section, based on the objective and psychophysical results obtained in Sect. 5.4, we present how the PSNR measures can be interpreted as perceptual quality and the usefulness of the PSNRs for evaluating

haptic data. We used the psychophysical evaluation results in Sect. 5.4.3 and observed at what PSNR levels the participants perceived the difference or degradation compared to the uncompressed haptic data.

In the case of position data compression methods, the psychophysical thresholds for the downsampling method were the downsampling factors of 3–4, according to Table 5.3. Note that this range was determined from the downsampling factor, where anyone of participants felt artifacts to the downsampling factor where all the participants felt artifacts. Then, according to Fig. 5.13, this downsampling factor range corresponded to the PSNRs of 48.6–51.7 dB. Similarly, the psychophysical thresholds of the prediction mode and the bidirectional prediction mode in Table 5.3 were the downsampling factors of 6–8 and 16–19, respectively. Then, from Fig. 5.13, the corresponding PSNRs were 55.6–58.5 and 55.9–58.0 dB, respectively. These psychophysical thresholds for the position data compression methods are shown in Table 5.6. As shown in this table, we could observe that the psychophysical thresholds of the both prediction and bidirectional prediction modes were in a similar range, whereas the downsampling method had a different range. Since the downsampling method did not have any prediction scheme, recovered position data at the virtual slave manipulator had jumpiness between two consecutive position samples. Therefore, in this downsampling method, the participants felt artifacts even though the downsampling factors were relatively small.

In the case of force data compression methods based on fixed rate downsampling, the psychophysical thresholds of the downsampling method, the prediction mode, and the bidirectional prediction mode were the downsampling factors of 2–3, 7–8, and 16–19, respectively, according to Table 5.4. From Fig. 5.14, the corresponding PSNRs were 37.8–42.5, 33.1–35.1, and 33.0–35.4 dB, respectively. In the case of adaptive downsampling methods, the psychophysical thresholds of the deadband method and the prediction-based deadband methods were the deadband thresholds of 3–4 and 9–12 %, respectively, according to Table 5.5. From Fig. 5.15, the corresponding PSNRs were 41.5–44.2 dB and 32.3–35.2 dB, respectively. These psychophysical thresholds for force data compression methods represented in PSNRs are shown in Table 5.7.

As shown in Table 5.7, we could observe that the force compression methods without using any prediction scheme (i.e. the downsampling method and the

Table 5.6 Psychophysical thresholds of the position data compression methods represented in PSNRs

Psychophysical thresholds	Downsampling	Prediction mode	Bidirectional mode
PSNR (dB)	48.6–51.7 dB	55.6–58.5 dB	55.9–58.0 dB

Table 5.7 Psychophysical thresholds of the force data compression methods represented in PSNRs

Psychophysical thresholds	Downsampling	Prediction mode	Bidirectional mode	Deadband	Prediction deadband
PSNR (dB)	37.8–42.5	33.3–35.1	33.0–35.4	41.5–44.2	32.3–35.2

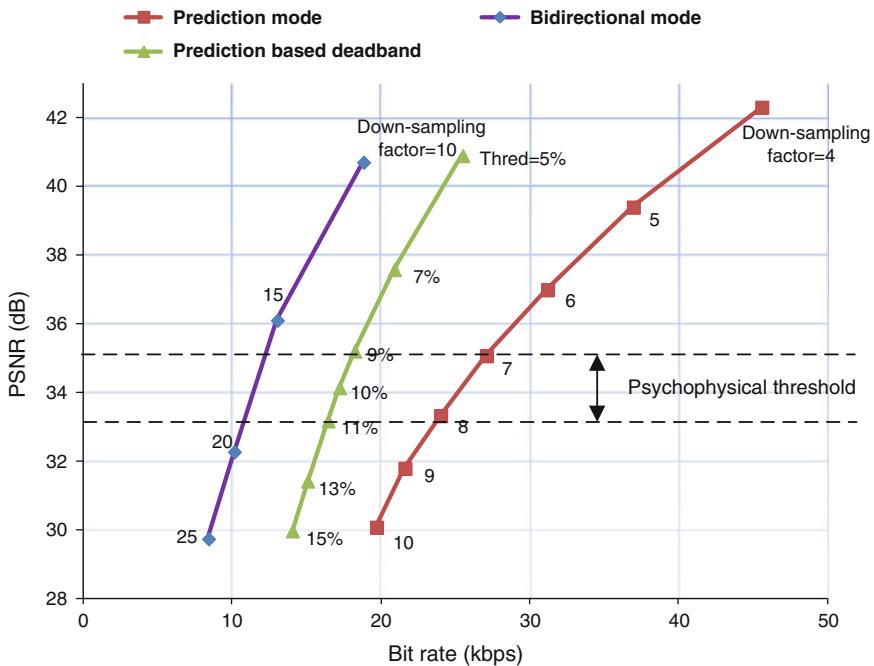


Fig. 5.18 Psychophysical thresholds for force data compression methods based on prediction schemes. All three methods had a similar PSNR range of psychophysical thresholds

deadband method) had different PSNR ranges of the psychophysical thresholds compared to the prediction-based methods (i.e., the prediction mode, the bidirectional prediction mode, and the prediction-based deadband method). This is because the compression methods without the prediction scheme had jumpiness between two consecutive force samples even in relatively smaller downsampling factors or deadband thresholds. However, for the three prediction-based methods, the participants felt artifacts at the certain range of PSNRs as shown in Fig. 5.18. Therefore, when haptic data compression methods were based on the prediction methods, the PSNR was useful to determine the psychophysical thresholds, and it could be used as a quality measurement in order to evaluate the haptic data compression methods.

References

1. P.G. Otanez, J.R. Moyne, D.M. Tilbury, Using deadbands to reduce communication in networked control systems. In *Proceedings of American Control Conference*, Anchorage, AK, pp. 3015–3020 (2002)
2. P. Hinterseer, E. Steinbach, S. Hirche, M. Buss, A novel, psychophysically motivated transmission approach for haptic data streams in telepresence and teleaction systems. In

- Proceedings of International Conference of Acoustics, Speech, and Signal Processing*, Philadelphia, PA, vol. 2, pp. 1097–1100, Mar 2005
- 3. M.H. Zadeh, D.W.L. Wang, E. Kubica, Perception-based lossy haptic compression considerations for velocity-based interactions. *Multimedia Syst.* **13**, 275–282 (2007)
 - 4. S. Payandeh, J. Dill, J. Zhang, A study of level-of-detail in haptic rendering. *ACM Trans. Appl. Percept.* **2**(1), 15–34 (2005)
 - 5. P. Hinterseer, E. Steinbach, Model-based data compression for 3D virtual haptic teleinteraction. In *Proceedings of IEEE International Conference on Consumer Electronics*, Las Vegas, NV, pp. 23–24, Jan 2006
 - 6. R.J. Anderson, M.W. Spong, Bilateral control of teleoperators with time delay. *IEEE Trans. Autom. Control* **34**(5), 494–501 (1989)
 - 7. G. Niemeyer, J.E. Slotine, Stable adaptive teleoperation. *IEEE J. Oceanic Eng.* **16**(1), 152–162 (1991)
 - 8. B. Hannaford, J.-H. Ryu, Time-domain passivity control of haptic interfaces. *IEEE Trans. Robot. Autom.* **18**, 1–10 (2002)
 - 9. P.C. Cosman, R.M. Gray, R.A. Olshen, Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. *Proc. IEEE* **82**(6), 919–932 (1994)

Chapter 6

Haptic Data Digitization and Forward Error Correction

Nomenclatures

- b Number of bits
- C_l Codeword length
- r Data part of a codeword
- s Parity part of a codeword

When haptic data are bilaterally transmitted through the network, erroneous haptic data can be detected at the receiving manipulators due to unreliable network conditions. Packet loss is considered as one of the main network problems causing erroneous haptic data. Packet corruption and out-of-ordered packets, which are caused by delay jitter and noisy link between the master and slave manipulators, are also detected as errors. Other than the network problems, errors can be caused by natural sources at the remote environment, such as thermal or radiation environments, while a teleoperation task is performed [1].

In order to recover such erroneous data, there are two types of approaches: The first type of approach is the estimation or prediction method as we presented in Chaps. 2–4. This type of approach is basically performed at the receiver without using any redundant information from the sending manipulators. The second type of approach is the error correction method that corrects the missing or erroneous part of the data by using redundant information added from the sender. In this chapter, we present the second type of approach, which is the error correction method for haptic data. After haptic data are fully digitized, which are represented in bit sequences or digital symbols, we perform forward error correction (FEC) in order to recover erroneous haptic data caused by unreliable network conditions. The remaining of this chapter is organized as follows. In Sect. 6.1, we present an overview of error correction methods for haptic data. In Sect. 6.2, we review the haptic data digitization method, which is based on the sampling and quantization processes. The proposed FEC method using digitized haptic data is presented in Sect. 6.3. In Sect. 6.4, we present the methodology to recover the packet loss by using the proposed FEC method. An experimental study using a haptic interface in

a virtual environment through the local network is provided in Sect. 6.5. The experimental study includes the psychophysical evaluation of haptic data digitization and the error correction performance of the proposed FEC method.

6.1 Forward Error Correction for Haptic Data

When erroneous data are detected at a receiver, the current network protocol technology provides transport control protocol (TCP) that performs automatic repeat query (ARQ) method to correct the errors. In this ARQ method, the receiver sends back an acknowledgment (ACK) to the sender for indication of correct packet reception. If a transmitting packet is lost or corrupted, the receiver sends back an unacknowledgment (NACK) for requesting the retransmission.¹ Since such ARQ method in TCP causes relatively large variations in network delay, error correction method is generally not recommended for haptic teleoperation systems. In the case of user datagram protocol (UDP), on the other hand, since it does not provide any error correction capability, there are more chances of packet loss, out-of-ordered packets, or any other unreliable network conditions, which degrade the performance and stability of overall teleoperation systems.

In digital transmission systems, such as telecommunication systems and digital multimedia systems, FEC codes are widely used to detect and correct erroneous data caused by any unreliable network condition or natural source. Unlike the ARQ error correction method, FEC codes do not use the backward link that requests the retransmission, but it requires higher bandwidth in the forward link in order to send redundant information for error correction capability. A haptic teleoperation system based on an FEC method and UDP is shown in Fig. 6.1. In order to perform the FEC method, haptic data need to be converted to digital forms, such as bit sequences or digital symbols. Given the digitized haptic data, an FEC encoding process that adds redundant information is performed before transmitting through the network. Then, using the redundant information, an FEC decoding process detects a limited number of errors that may occur during the network transmission and corrects the errors without performing the retransmission. In this chapter, we first review the haptic data digitization in order to represent haptic data into digitized symbol information. Given the digitized symbols of haptic data, the proposed FEC method is applied to compensate for unreliable network condition, such as the noisy link and packet loss. We conduct the psychophysical evaluation of the digitized haptic data. In this evaluation, we determine the required number of bits, in which a human operator may not feel any perceptual difference compared to nondigitized haptic data. Then, given the number of bits, we implement the proposed FEC method and demonstrate its error correction performance.

¹The details of TCP including the retransmission mechanism are also presented in Sect. 4.2.

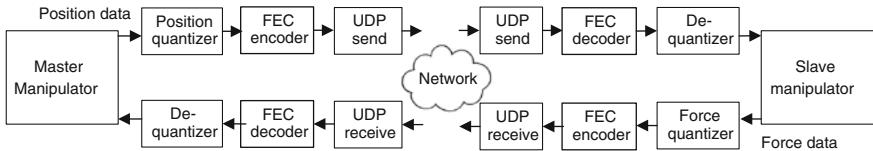


Fig. 6.1 A haptic teleoperation system using a forward error correction (FEC) method, which is based on quantization and user datagram protocol (UDP) protocols

6.2 Haptic Data Digitization

In the current haptics technology, haptic data, including position, velocity, and force data, are mostly available in floating point format in order to preserve the perceptual quality of haptic data and to maintain stability of an overall system. Nevertheless, digitization of haptic data has been studied in the area of haptics. Diolaiti et al. [2] conducted an experimental study when stiff virtual objects were represented in digitized haptic data. They also addressed stability of a haptic interface when haptic data were fully digitized. In general, the digitization of haptic data can be achieved by a quantization process in addition to a sampling process. When haptic data are digitized, they can be not only robust against unreliable network conditions but also can be processed by a variety of digital signal-processing methods, such as error correction, for improved reliability of teleoperation systems [3].

In signal processing, the quantization is the process of mapping a large set of continuous signals to a smaller set of discrete signals or bit sequences [4]. A quantizer is a signal-processing algorithm or a device that performs the quantization. Given the sampling rate of local clocks (e.g. 1 kHz), the quantization process for position and force data is shown in Fig. 6.2a. When a number of bits b is determined, the range of haptic data can be divided into 2^b intervals. Then, any position or force sample within the interval is mapped to a quantized value, which is the middle value of that interval. Each quantized value has a corresponding index, which is a binary sequence predefined by an index book. The index book consists of binary sequences and corresponding position or force intervals, given the number of bits. A simple example of force data quantization using the index book and corresponding indices is shown in Fig. 6.2b. When haptic data are transmitted from the sending manipulators as shown in Fig. 6.2a, the haptic data through the network are sampled and quantized into binary sequences. At the receiving manipulators, dequantizers are used to recover original haptic data using the index book.

In this quantization process for haptic data, the quantization error between the quantized and the floating point haptic data is inevitably introduced after received data are recovered by the dequantizers. Hence, when haptic data are digitized, it is important to retain signal fidelity that ensures the perceptual transparency of teleoperation systems while eliminating unnecessary part of haptic data. Psychophysically, it is known that a relatively small difference of haptic data may

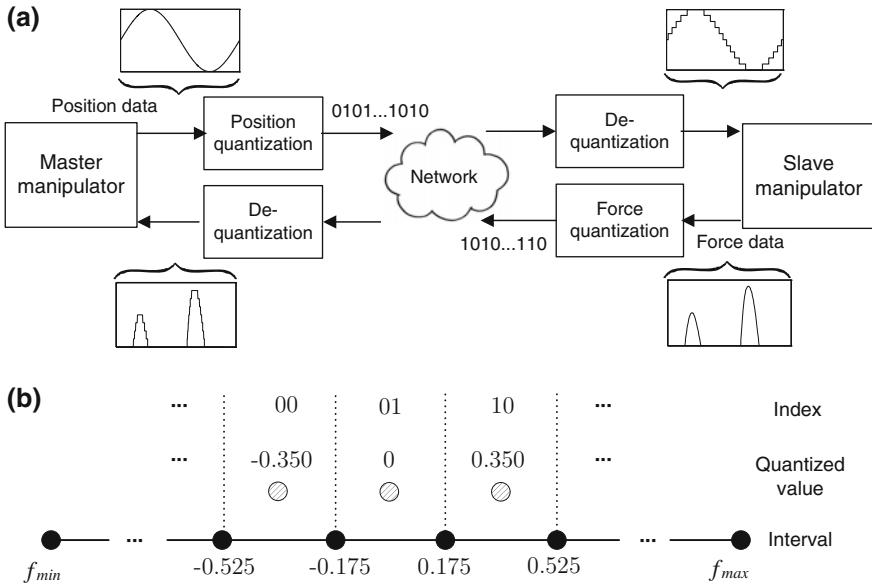


Fig. 6.2 **a** Haptic position and force data quantizations. **b** An example of force data quantization using an index book and corresponding indices

not be noticed by human perception [5]. Therefore, the number of bits, which is defined by the index book design, needs to be chosen such that the differences between the quantized values and the corresponding floating point haptic data are unnoticeable by human perception through a psychophysical evaluation. Note that such quantization process can also be used as a rate-distortion optimized quantization that performs a lossy compression by managing the amount of quantization errors and the number of bits [4]. In this chapter, however, we focus on the quantization process simply for digitizing haptic data in order to be processed by the proposed FEC method.

6.3 Forward Error Correction for Haptic Data

In coding theory and information theory, FEC codes² can be categorized into two groups: *Block codes* use fixed size of bits or symbols for encoding and decoding processes. These types of codes include the Reed–Solomon (RS), Golay, and Hamming codes. *Convolutional codes* use arbitrary length of neighboring information for the error correction. In many digital applications, a convolutional code is mostly used as a concatenated form with the block codes in order to achieve the

²Types of FEC codes are presented in Appendix H.

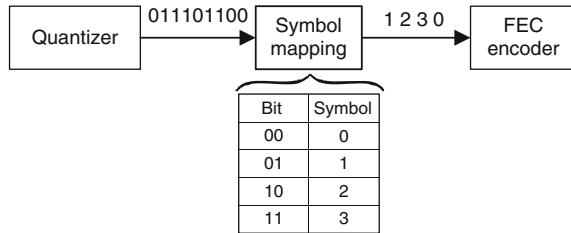


Fig. 6.3 Bit-to-symbol mapping for FEC encoding process. In this example, a symbol is converted by using 2 bits

performance improvement. In this chapter, we use an RS code, which is one of the widely used block codes for many recent digital applications, such as multimedia systems, compact disk, and hard disk [3].

6.3.1 *Symbol Mapping*

An RS code processes digital symbol data instead of bit sequences. Hence, bit sequences, which are generated from the presented quantization method in Sect. 6.2, need to be converted to symbol data. A bit-to-symbol mapping scheme before performing the FEC encoding process is shown in Fig. 6.3. In this example, four symbols are used to be mapped with two-bit sequences. The number of bits per symbol can be defined by considering data rate over limited network bandwidth and robustness against unreliable network conditions. If a higher number of bits are used to map a symbol, higher data rate can be transmitted through the network. Otherwise, if a lower number of bits are used to map a symbol, which generates a smaller set of symbols, the probability of errors can also be lower compared to using the higher number of bits. After symbols are transmitted to the receiving manipulators, the received symbols, which are processed by a FEC decoder for error correction, need to be converted back to bit sequences to be dequantized.

6.3.2 *Reed–Solomon Codes*

An RS code is a linear systematic block code based on the finite field theory. In coding theory, a systematic code is an error correcting code, in which the encoded output codeword contains the input symbols. The output codeword to be processed by an RS code consists of two parts as shown in Fig. 6.4: The data part is the first r number of symbols to be protected against any errors. The parity part is the following $2s$ number of redundant information. In general, such RS code is represented as an RS (C_l, r) code, where C_l is the codeword length and r is the data

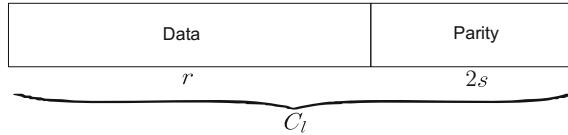


Fig. 6.4 A Reed–Solomon (RS) codeword consists of the data part with length r and the parity part with length $2s = C_l - r$

part length in the codeword. For a symbol that has a length of b bits, the length of C_l is given by $C_l = 2^b - 1$. The error correction capability of an RS code depends on the parity part length, which is given by $2s = C_l - r$, and this implies that a decoder of RS (C_l, r) code can correct up to s number of erroneous symbols in the codeword with length C_l .

When the proposed RS code is used as an FEC method for haptic teleoperation systems, a codeword can be obtained from multiple-degree of freedom (m-DoF) haptic data within a sampling period. For example, if 1-DoF position sample is represented by a 9-bit sequence and a symbol is mapped with 3 bits, 3-DoF position data at a sampling period require 9 symbols in the data part. In this case, we can choose the RS(15, 9) code for every sampling period, and this codeword can be used to construct a single packet transmitting from a UDP send protocol in real time. Then, at a receiving manipulator, an RS decoder uses the parity part with length 6, and it can correct up to 3 symbols if any errors occur during the network transmission.

6.4 Forward Error Correction for Packet Loss Recovery

When haptic data are transmitted through the layered architecture of the Internet protocol suite, some of unreliable network conditions, such as noisy link between the master and slave manipulators, may not be appropriate to be processed by the proposed FEC method. For example, when UDP is used as a transport protocol, it is able to provide the checksum field for error detection as presented in Fig. D.1 of Appendix D. If a packet corruption occurs due to the noisy link and it is detected in the transport layer, one can design that the corrupted packet can be dropped without forwarding to the application layer. Therefore, in addition to such packet drop in the transport layer, the packet loss problem, which is caused by the network congestion over limited bandwidth, is considered as one of the main challenges in UDP-based haptic teleoperation systems over the network.

In order to compensate for the packet loss problem of haptic teleoperation systems, we present the methodologies using the proposed FEC method. The first methodology for 1-DoF haptic data is illustrated in Fig. 6.5. In this methodology, every sample of haptic data is packetized from a sending manipulator, and it is represented by three symbols. In the FEC encode stage of Fig. 6.5, a parity packet is

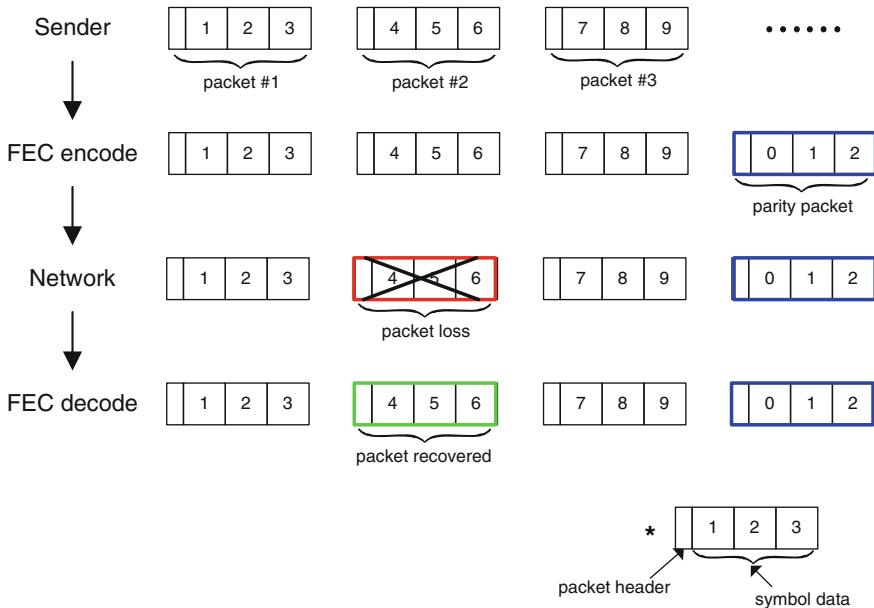


Fig. 6.5 Haptic data loss recovery using the FEC encode and decode schemes

additionally assigned after three consecutive data packets are transmitted through a UDP send protocol. Since UDP does not provide any retransmission or error correction scheme, any packet may be lost during the network transmission as shown in the network stage of Fig. 6.5. When a receiving manipulator receives the packets by a UDP receive protocol, the FEC decode process is performed in order to detect the three symbols of the lost packet and to recover them using the parity packet. In this FEC decode stage, the length of the codeword is determined by three consecutive data packets and one parity packet. Therefore, the receiving manipulator is able to detect which packet is missing and also to correct the lost packet depending on the length of the parity packet. Note that in the case of packet drop, which is executed in the transport layer, it can also be detected as an error in the FEC decode stage, and hence, it is corrected depending on the length of the parity packet.

Since the methodology of haptic data loss recovery described in Fig. 6.5 requires a multiple number of packets in order to perform the FEC decode process, this process inevitably introduces a constant group delay. In the case of Fig. 6.5, a parity packet is assigned after every three data packets, and hence, 3-ms group delay needs to be introduced, given 1 kHz sampling rate. When 3-DoF haptic data are transmitted through the network, which is more practical case to perform a haptic teleoperation task, another methodology to recovery haptic data loss is illustrated in Fig. 6.6. In this methodology, every sample of 3-DoF haptic data is packetized from a sending manipulator, and each packet consists of three symbols. In the FEC encode stage of Fig. 6.6, a parity packet is additionally assigned after three data

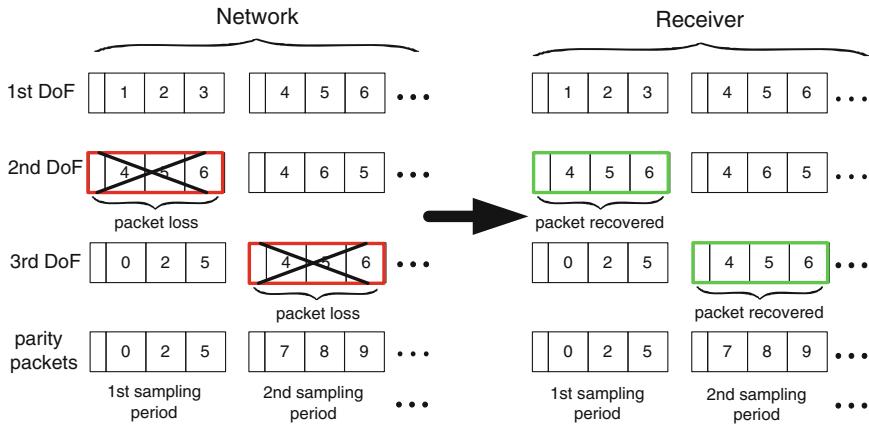


Fig. 6.6 A FEC method for haptic data loss recovery when 3-degree of freedom (DoF) haptic data are transmitting through the network

packets within a sampling period, and they are transmitted through the network using a UDP send protocol. When any packet at every sampling period is assumed to be lost in the network stage, the FEC decoder uses the parity packets to recover the lost packets in the FEC decode stage. In this example, since the FEC decode process is performed at every sampling period, it does not introduce a group delay.

6.5 Experimental Study

We performed experimental studies in order to demonstrate the haptic data digitization (quantization) and the proposed FEC method. We used the experimental setup based on the haptic interface in virtual environment presented in Chap. 5. In the first part of the experiment, presented in Sect. 6.5.1, we provide the psychophysical evaluation results of the haptic data digitization. In the second part of the experiment, presented in Sect. 6.5.2, we demonstrate the error correction performance of the proposed FEC method, given the psychophysical evaluation results.

6.5.1 Psychophysical Evaluation for Haptic Data Digitization

The objective of this psychophysical evaluation is to determine the number of bits, in which the participants do not feel the difference between the quantized haptic data and the floating point haptic data. The numbers of bits for position and force data, which were determined in this experimental study, were used to implement the proposed FEC method.

6.5.1.1 Psychophysical Evaluation Procedure

In order to conduct the psychophysical evaluation, 12 students (10 male and 2 female) between ages of 20 and 34 participated. The details about the participants are presented in Sect. 5.4.3.2. We implemented uniform quantizers for position and force data with different numbers of bits. Data ranges of 3-DoF positions were (-210 , 210 mm), (-105 , 145 mm), and (-100 , 120 mm), and 3-DoF force data had (-3.3 , 3.3 N). At the slave side, we used the implemented virtual slave manipulator with two spherical objects. The details of implementing the virtual slave manipulator are also presented in Appendix G. Based on the virtual workspace, we conducted the pairwise comparison, which determines the perceptual difference between the quantized haptic data and the floating point haptic data. The virtual workspace for psychophysical evaluation is shown in the left figure of Fig. 6.7. As shown in this figure, one of the spherical objects was represented by the quantized haptic data, whereas the other one was the floating point haptic data.

Although the quantized and floating point haptic data were transmitted through the local area network cable, the participants were asked to conduct a teleoperation task in the virtual workspace. The virtual teleoperation task includes making contact with the spherical objects and sliding on the surface of the objects as described in the right figures of Fig. 6.7. After a practice session, the participants conducted the pairwise comparison while the number of bits for quantization was randomly changed. Two identical spherical objects were also randomly provided, but such

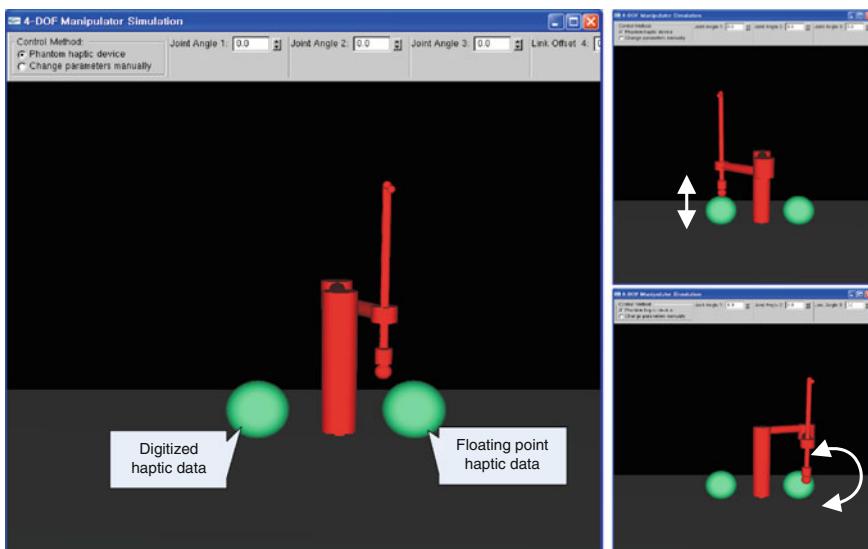


Fig. 6.7 Psychophysical evaluation of haptic data quantization: a virtual workspace for a pairwise comparison of floating point and quantized haptic data (Left), psychophysical tasks such as making contact with the spherical objects (Top-right), and sliding on the surface of the spherical objects for generating continuous 3-DoF haptic data (Bottom-right)

information about the parameters was not given to the participants. For each evaluation, the participants were allowed to conduct the pairwise comparison for 10 s. After 10 s, they had to determine whether the haptic sensation of two spherical objects is different or not. Based on the answers from the participants, “yes” was considered that the haptic data quantization caused sufficient artifacts or degraded perceptual transparency compared to the floating point haptic data.

6.5.1.2 Psychophysical Evaluation Results

The psychophysical evaluation results of (1) position data quantization and (2) force data quantization are shown in Table 6.1. We measured the perceptual thresholds when different numbers of bits were applied. As shown in the table, in the case of position data quantization, all 12 participants felt the difference between the floating point and the quantized haptic data when the number of bits was less than or equal to 7. Some participants did not feel artifacts from 8 bits, and all the participants did not notice any difference with the number of bits larger than 10. In the case of force data quantization, some participants did not feel artifacts from 8 bits, and all the participants did not notice any difference with the number of bits larger than 9. During this experimental study, it was reported that the artifacts were hardly detected when the participants simply made contact with the spherical objects as shown in the top-right figure of Fig. 6.7. On the other hand, the artifacts and perception degradations were mostly detected when they were sliding on the surface of the spherical objects, which generated continuous 3-DoF haptic data as shown in the bottom-right figure of Fig. 6.7. According to the psychophysical evaluation results, we determined the required bits as 11 and 10 bits for quantizing position and force data, respectively. Therefore, these bit resolutions were used to implement the proposed FEC method and to evaluate the error correction performance in Sect. 6.5.2.

Table 6.1 Psychophysical evaluation results of the position and force data quantizations

Number of bits (bits)	Position data (%)	Force data (%)
6	0	0
7	0	0
8	8.3	8.3
9	33.3	58.3
10	83.3	100
11	100	100
12	100	100

Number of participants who did not detect artifacts was represented in percentage

6.5.2 Performance Evaluation for Forward Error Correction

In this experimental study, we demonstrate the error correction capability of the proposed FEC method in haptic teleoperation systems. Given 11 and 10 bits for position and force data quantizations, respectively, the proposed FEC method was implemented, and the error correction performance was demonstrated under (1) the additive noise and (2) the packet loss behavior.

6.5.2.1 Performance Under Additive Noise

In this study, the proposed FEC method was evaluated under the additive noise environment, which was modeled as additive white Gaussian noise. Such additive noise is commonly used to understand the underlying performance of the proposed method before considering more realistic network behaviors. Hence, we used the additive noise to generate erroneous haptic data and to evaluate the error correction performance of the proposed FEC method. In order to correct any errors caused by such additive noise, we implemented the RS(15, 7) code for position data in the uplink transmission and the RS(7, 5) code for force data in the downlink transmission. Before the FEC encoding, a bit-to-symbol mapping scheme was used with 2 bits per symbol rate. Since we used 10 bits for the force data quantization, the data part of codeword was simply $r = 5$ as presented in Fig. 6.4. On the other hand, for the position data quantization, we had to add 3 zero bits so that the data part of codeword was $r = 7$ in order to express 11 bits of position data.

Figure 6.8 shows that floating point haptic data were corrupted by the additive noise. The amount of additive noise was determined by signal-to-noise ratio (SNR) denoted in dB, and a higher SNR represents a lower amount of additive noise. In this study, we added 30 dB SNR for position data and 25 dB SNR for force data in order to generate erroneous haptic data. When these haptic data are quantized and transmitted, the receiving manipulators find digital symbols instead of floating point data. Hence, the quantized haptic data may be more robust against the additive noise with relatively higher SNRs. However, in relatively lower SNRs, the receiving manipulators can easily make incorrect decisions, which result in erroneous symbols. Position and force data that were quantized and recovered by such incorrect decisions are shown in Fig. 6.9. As presented in floating point haptic data in Fig. 6.8, the quantized position and force data were also corrupted by the additive noise with 30 and 25 dB SNRs, respectively.

The erroneous haptic data, which were generated from the additive noise, were corrected by the implemented FEC codes. The error correction results of position and force data are shown in Fig. 6.10. As shown in the figures, the proposed FEC codes corrected all the errors, given the levels of additive noise. In this study, we also evaluated the implemented FEC codes when position and force data were corrupted under different levels of additive noise. We measured the number of bit errors between the decoded haptic data at the receiving manipulators and the

Fig. 6.8 Floating point haptic data corrupted by additive noise: **a** position data with 30 dB SNR, and **b** force data with 25 dB SNR

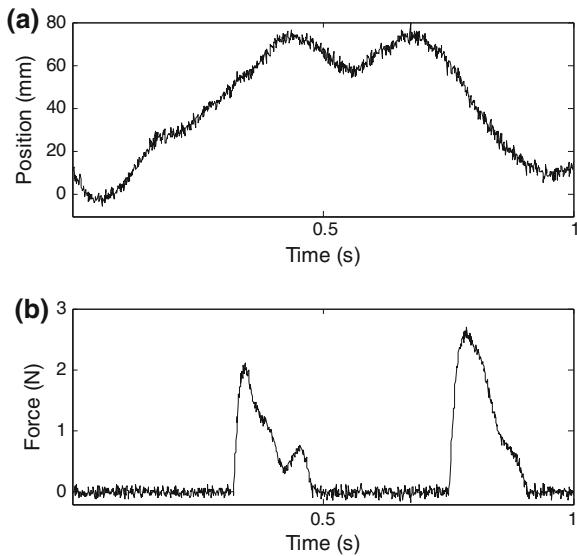
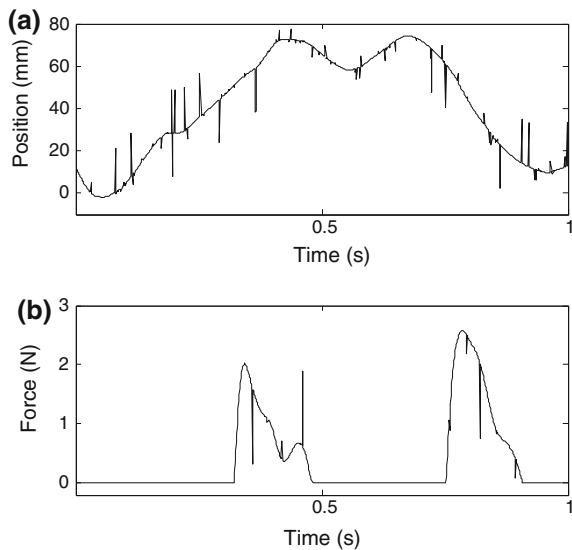


Fig. 6.9 Quantized haptic data recovered at the receiving manipulators without using the FEC method: **a** quantized position data corrupted by additive noise with 30 dB signal-to-noise ratio (SNR), and **b** quantized force data corrupted by additive noise with 25 dB SNR



original haptic data at the sending manipulators for a 10-s operation. The bit error rates (BER) of the FEC codes with different SNRs are shown in Fig. 6.11. As shown in this figure, when the RS(15, 7) code was used for position data, the errors were found up to 29 dB SNR. In the case of force data, the errors were found up to 24 dB SNR when the RS(7, 5) code was used. Note that such BER performance depended on the choices of FEC codes, including the length of parity part, bit-to-symbol mapping schemes, and types of haptic data. In this study, we only

Fig. 6.10 Error-corrected haptic data using the proposed FEC codes: **a** position data were corrected by the RS(15, 7) code, and **b** force data were corrected by the RS(7, 5) code

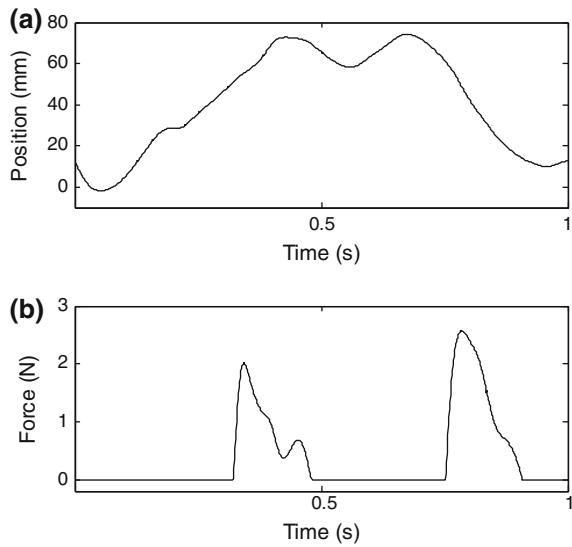


Fig. 6.11 Bit error rates (BER) of position and force data error corrections. The RS(15, 7) code was used for position data, and the RS(7, 5) code was used for force data



chose the shorter length of codeword in order to process within a sampling period. With relatively longer length of codeword, the proposed code can be more robust against the additive noise and can correct more number of errors. In Fig. 6.11, we could also observe that for the force data case, the number of errors was relatively small at lower SNRs. This is because there are noncontact force regions in force data as shown in Fig. 6.10b. Since such noncontact forces are represented as zeros, the decoder at the master manipulator could make decision easily.

6.5.2.2 Performance Under Packet Loss Behavior

In this experiment, the proposed FEC method was evaluated under the packet loss behavior through the network. Packetized haptic data were randomly selected to be lost during the network transmission, and the examples of such packet loss behaviors for 1-DoF position and force data are shown in Fig. 6.12. Note that in these specific examples, the packet loss rates of position and force data were 3 and 5 %, respectively.

In order to recover the packet losses by using the proposed FEC method, we used the methodology presented in Fig. 6.5: One parity packet was additionally assigned after every three consecutive haptic data packets were transmitted for both position and force data cases. The bit-to-symbol mapping scheme was used with 4 bits per symbol rate. For each sample of position and force data, zero bits were added so that each packet of position and force data has a fixed size of 3 symbols. After the bit-to-symbol mapping, we implemented the RS(15, 9) code for both position and force data. Therefore, the data part of codeword was $r = 9$, and the implemented FEC code was able to recover errors up to $s = (15 - 9)/2 = 3$ symbols.

The packet loss recovery results of position and force data by using the implemented FEC code are shown in Fig. 6.13. We assumed that one out of three data packets was randomly lost during the transmission. The parity packets, which need to be transmitted after every three data packets for the FEC decoding process, were assumed to be transmitted without any loss. Hence, the packet loss rates in the both uplink and downlink transmissions were 25 %. Since we used the RS(15, 9) code, which could correct errors up to 3 symbols, one dropped packet consisting of 3 symbols could be fully recovered as shown in the figures.

Fig. 6.12 Packet loss behaviors of **a** 1-DoF position data (3 % loss rate) and **b** 1-DoF force data (5 % loss rate)

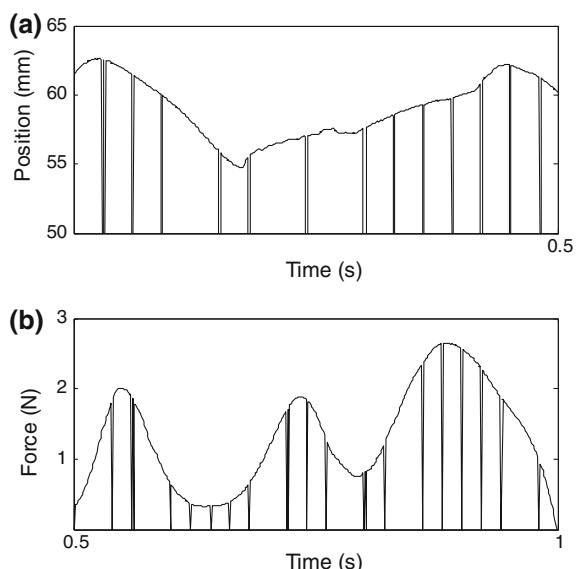
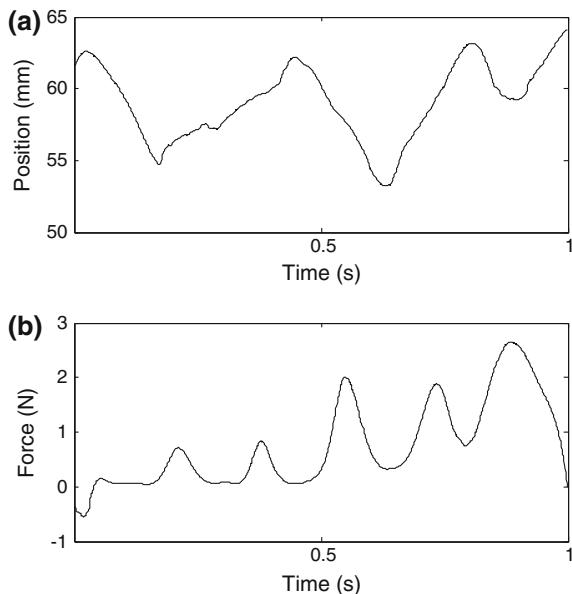


Fig. 6.13 Haptic data loss recovery using the FEC codes: **a** the RS(15, 9) code was used to recover position data, and **b** force data



6.6 Discussion

As presented in Sect. 6.4, the proposed FEC method for packet loss recovery may introduce a constant group delay depending on the design of parity packets or the length of RS codeword. A relatively small group delay with a choice of shorter codeword length may be negligible. However, when a longer codeword length is chosen for improved error correction capability, an overall teleoperation system may be unstable. In order to prevent such instability problem, the passivity-based control methods, including the scattering transformation [6], wave variables [7], and passivity observer/passivity controller [8], can be additionally adopted, which are intensively presented in Chap. 3.

References

1. M. Sitti, H. Hashimoto, Tele-nanorobotics using atomic force microscope, in *Proceedings IEEE/RSJ International Conference Intelligent Robots and Systems*, Victoria, BC (IEEE, New York, 1998), pp. 1739–1746
2. N. Diolaiti, G. Niemeyer, F. Barbagli, J.K. Salisbury, Stability of haptic rendering: discretization, quantization, time delay, and coulomb effects. *IEEE Trans. Robot.* **22**(2), 256–268 (2006)
3. S. Lin, D.J. Costello, *Error Control Coding: Fundamentals and Applications* (Prentice Hall, Englewood Cliffs, NJ, 1983)

4. Y. Wang, J. Ostermann, Y.-Q. Zhang, *Video Processing and Communications* (Prentice Hall, Upper Saddle River, NJ, 2002)
5. L. Bizo, J. Chu, F. Sanabria, P. Killeen, The failure of weber's law in time perception and production. *Behav. Process.* **71**(2), 201–210 (2006)
6. R.J. Anderson, M.W. Spong, Bilateral control of teleoperators with time delay. *IEEE Trans. Autom. Control* **34**(5), 494–501 (1989)
7. G. Niemeyer, J.J.E. Slotine, Stable adaptive teleoperation. *IEEE J. Ocean. Eng.* **16**(1), 152–162 (1991)
8. B. Hannaford, J.-H. Ryu, Time-domain passivity control of haptic interfaces. *IEEE Trans. Robot. Autom.* **18**, 1–10 (2002)

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this book, we explored haptic data processing, including prediction/estimation, compression, and error correction, for teleoperation systems or haptic interfaces over the network.

We first presented the haptic data prediction and estimation method in order to overcome unreliable network conditions, such as time-varying delay, packet loss, out-of-ordered packets, and packet corruption. In this method, based on the proposed structure of transport layer, the information about unreliable network conditions was forwarded to the application layer, and hence, the proposed Bayesian approach was effectively used as either prediction or estimation. We also introduced the stochastic representation of position and force data transmitting through the network, and the stability analysis based on the stochastic formalism. The experimental study using two haptic devices connected by the local area network cable showed that the proposed method in the transport and application layers compensated for unreliable network conditions in real time. It also confirmed that the Bayesian approach based on the information from the transport layer outperformed the Bayesian approach without such information.

For the haptic data compression methods, we mostly focused on downsampling methods, including the fixed rate downsampling and the adaptive downsampling. Based on the fixed rate downsampling, we proposed the prediction methods and contact force detection scheme for further efficiency. In the experimental study, we introduced the objective evaluation measure for haptic data compression methods. By conducting the psychophysical evaluation using the 3-degree of freedom haptic interface in virtual environment, we investigated the usefulness of this objective evaluation measure for haptic applications. According to both the objective and the psychophysical evaluation results, the PSNR metric could be used as a quality measurement for the prediction-based haptic compression methods.

Finally, we presented the forward error correction (FEC) method for improved reliability of haptic teleoperation systems or haptic interfaces. Based on the digitization of haptic data, we used the Reed–Solomon code as a FEC method to correct erroneous haptic data that may occur during the network transmission. In the experimental setup using the haptic interface in virtual environment, we performed the psychophysical evaluation to find the required number of bits for the haptic data digitization. Given the bit resolutions of position and force data, we also evaluated the error correction performance of the proposed FEC method under the additive noise and the packet loss behavior.

7.2 Future Work

- *Integrated methods:* In this book, we demonstrated the performances of the prediction/estimation method, the compression method, and the error correction method in different experimental setups. In order to achieve the performance improvement and further robustness over unreliable network conditions, an integrated structure of the proposed methods can be developed. For example, an integration of the prediction/estimation method and the error correction method will definitely provide further robustness against the unreliable network. The haptic compression method may also provide enhancement when it is integrated with the prediction/estimation method or the error correction method. An example of integrated structure is shown in Fig. 7.1.

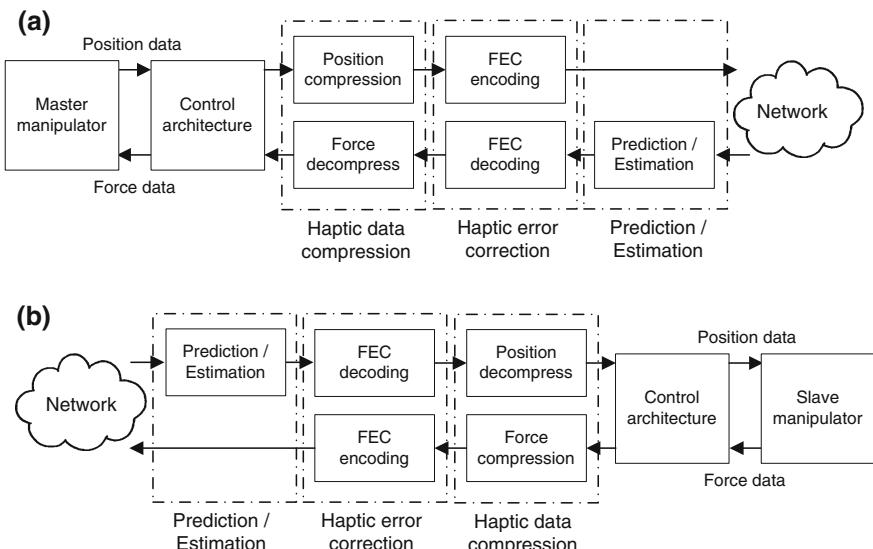


Fig. 7.1 An integrated structure of haptic data compression, error correction, and prediction/estimation methods. **a** Master side. **b** Slave side

- *Hardware platform of the proposed methods:* Since the experimental setup presented in this thesis is a software-based platform, there may be a computational limitation when the proposed methods are integrated. As a future work, the integrated structure can be implemented on the hardware-based platform, such as FPGA, in order to realize parallel processing of the proposed methods with computational complexity.
- *Haptic data compression:* In Chap. 5, we presented the haptic data compression methods mostly based on downsampling. The future research in this topic will include other types of compression methods, such as lossless entropy coding methods or transform coding methods, which may be performed in real time.
- *Metric measurement of haptic data compression methods:* In the part of experimental study of haptic data compression methods, we presented that at what PSNR values the participants perceived the distortion compared to uncompressed haptic data. Besides the PSNR metric in this thesis, the future research in this topic will include what types of metric measurements may be suited to evaluate the perceptual quality of haptic data.
- *Haptic data error correction:* In Chap. 6, we introduced the RS code as a FEC method to compensate for erroneous position and force data. As the RS code is widely used for other multimedia and telecommunication systems, it is also used as a concatenated form with other types of error correcting codes, such as a convolution code. For further reliability improvement in transmitting haptic data, such concatenated code will be considered as a future objective. Furthermore, other types of error correcting codes, which may be performed in real-time, will be also investigated.

Appendix A

Basics of Bayesian Filtering

A.1 Bayesian Statistics

In statistics, the Bayesian theory allows modeling uncertainty in real-world problems by incorporating prior knowledge and observational evidences. Any uncertainty in the Bayesian statistics is considered as a stochastic variable. In order to solve the stochastic problem, let us first define a generic state-space form in discrete time domain:

$$\mathbf{x}[k] = \mathbf{g}(\mathbf{x}[k-1] + \mathbf{l}[k-1]), \quad (\text{A.1})$$

$$\mathbf{y}[k] = \mathbf{h}(\mathbf{x}[k] + \mathbf{r}[k]). \quad (\text{A.2})$$

where (A.1) and (A.2) are called the state and observation equations, respectively. $\mathbf{x}[k]$ and $\mathbf{y}[k]$ are the state and observation vectors, respectively. $\mathbf{l}[k-1]$ and $\mathbf{r}[k]$ are the state and observation noises, respectively. $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$ are the state and observation transition functions, which may be nonlinear. The state (A.1) represents the state transition probability, which is $p(\mathbf{x}[k] | \mathbf{x}[k-1])$, and the observation (A.2) is the observation process $p(\mathbf{y}[k] | \mathbf{x}[k])$, which is related to the observation noise model.

Finding the state $\mathbf{x}[k]$ is an inverse problem. Given available observations $\mathbf{y}[0:k]$ (i.e. $\mathbf{y}[0], \mathbf{y}[1], \mathbf{y}[2], \dots, \mathbf{y}[k-1], \mathbf{y}[k]$), and known transition functions, $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$, $\hat{\mathbf{x}}[k]$ needs to be obtained by the optimal or sub-optimal solutions. In the Bayesian statistics, a complete solution to find the state is computing a *posterior distribution*, which is $p(\mathbf{x}[k] | \mathbf{y}[0:k])$ or $p(\mathbf{x}[k] | \mathbf{y}[k])$. Note that a posterior distribution is a distribution of an unknown quantity, which is a random variable conditioned on the evidence obtained from an observation. In order to compute this distribution recursively, we need to assume the following: (1) The states follow a first-order Markov process, such that $p(\mathbf{x}[k] | \mathbf{x}[0:k-1]) = p(\mathbf{x}[k] | \mathbf{x}[k-1])$, (2) the observations are independent of the given states. According to the Bayes' rule, the posterior distribution can be solved as:

$$\begin{aligned}
p(\mathbf{x}[k] \mid \mathbf{y}[0:k]) &= \frac{p(\mathbf{y}[0:k] \mid \mathbf{x}[k])p(\mathbf{x}[k])}{p(\mathbf{y}[0:k])} \\
&= \frac{p(\mathbf{y}[k], \mathbf{y}[0:k-1] \mid \mathbf{x}[k])p(\mathbf{x}[k])}{p(\mathbf{y}[k], \mathbf{y}[0:k-1])} \\
&= \frac{p(\mathbf{y}[k] \mid \mathbf{y}[0:k-1], \mathbf{x}[k])p(\mathbf{y}[0:k-1] \mid \mathbf{x}[k])p(\mathbf{x}[k])}{p(\mathbf{y}[k] \mid \mathbf{y}[0:k-1])p(\mathbf{y}[0:k-1])} \\
&= \frac{p(\mathbf{y}[k] \mid \mathbf{y}[0:k-1], \mathbf{x}[k])p(\mathbf{x}[k] \mid \mathbf{y}[0:k-1])p(\mathbf{y}[0:k-1])p(\mathbf{x}[k])}{p(\mathbf{y}[k] \mid \mathbf{y}[0:k-1])p(\mathbf{y}[0:k-1])p(\mathbf{x}[k])} \\
&= \frac{p(\mathbf{y}[k] \mid \mathbf{x}[k])p(\mathbf{x}[k] \mid \mathbf{y}[0:k-1])}{p(\mathbf{y}[k], \mathbf{y}[0:k-1])}
\end{aligned} \tag{A.3}$$

In (A.3), the posterior distribution is described by the following three distributions:

- *Prior distribution:* $p(\mathbf{x} \mid \mathbf{y}[0:k-1])$ is called the prior distribution, which is defined by the following integration:

$$p(\mathbf{x} \mid \mathbf{y}[0:k-1]) = \int p(\mathbf{x}[k] \mid \mathbf{x}[k-1])p(\mathbf{x}[k-1] \mid \mathbf{y}[0:k-1]) \, d\mathbf{x}$$

where $p(\mathbf{x}[k] \mid \mathbf{x}[k-1])$ is the transition density of the state.

- *Likelihood distribution:* $p(\mathbf{y}[k] \mid \mathbf{x}[k])$ is called the likelihood distribution, and it determines the observation noise model according to (A.2).
- *Evidence distribution:* $p(\mathbf{y} \mid \mathbf{y}[0:k-1])$ is called the evidence distribution, which is defined by the following equation with an integral:

$$p(\mathbf{y} \mid \mathbf{y}[0:k-1]) = \int p(\mathbf{y}[k] \mid \mathbf{x}[k])p(\mathbf{x}[k] \mid \mathbf{y}[0:k-1]) \, d\mathbf{x}.$$

In the Bayesian statistics, these three distributions are computed or approximated in order to obtain the posterior distribution.

A.2 Bayesian Filtering

A Bayesian filter is a method to apply the Bayesian statistics in order to estimate or predict the state given the available observations. Using a Bayesian filter, an optimal solution is to find the minimum between the actual state and the estimated state. One of the popular criteria to define the optimal solution is the minimum mean-squared error (MMSE), which is given by

$$\mathbb{E}\left(\|\mathbf{x}[k] - \hat{\mathbf{x}}\|^2 \mid \mathbf{y}[0:k]\right) = \int \|\mathbf{x}[k] - \hat{\mathbf{x}}\|^2 p(\mathbf{x}[k] \mid \mathbf{y}[0:k]) d\mathbf{x}. \quad (\text{A.4})$$

Note that there are other criteria that also define the optimal solution, such as the maximum a posteriori (MAP), maximum likelihood, and min-max criteria [49]. Based on the MMSE criterion, a Bayesian filter is said to be optimal when it finds the posterior distribution that minimizes the error. The optimal solution using a Bayesian filter is given by

$$\mathbb{E}(\mathbf{x}[k] \mid \mathbf{y}[0:k]) = \int \mathbf{x}[k] p(\mathbf{x}[k] \mid \mathbf{y}[0:k]) d\mathbf{x}. \quad (\text{A.5})$$

Note that (A.5) is called a conditional mean because the expectation of the state $\mathbf{x}[k]$ is conditioned on the observations $\mathbf{y}[0:k]$. This optimal solution is only conceptual since the integral function is not solvable in practice. Hence, Bayesian filters, including the Kalman filter¹ and the Bayesian particle filter, are used as the sub-optimal or locally optimal solution in practice.

A.3 Bayesian Particle Filter

The Bayesian particle filter is a sub-optimal solution that sequentially finds the state based on the Monte Carlo simulation. Hence, it is also called a sequential Monte Carlo (SMC) method. In this method, the state-space is expressed as a set of samples, called particles. Given a number of particles with some random measures, the samples with higher probability are concentrated to represent the state. Because of the Monte Carlo characteristic of the Bayesian particle filter, a large number of particles provides a higher probability to approach the optimal solution.

To be more specific, let $\{\mathbf{x}^i[0:k], w^i[k]\}_{i=1}^N$ denote the random measures in order to represent the posterior distribution $p(\mathbf{x}[k] \mid \mathbf{y}[0:k])$. Note that $\{\mathbf{x}^i[0:k], i = 1, \dots, N\}$ is the set of random samples with the associated weights $\{w^i[k], i = 1, \dots, N\}$. The associated weights, which are called the importance weights, are normalized such as $\sum_i w^i[k] = 1$. Then, the posterior distribution can be approximated as

$$p(\mathbf{x}[k] \mid \mathbf{y}[0:k]) \approx \frac{1}{N} \sum_{i=1}^N w^i[k] \delta(\mathbf{x}[0:k] - \mathbf{x}^i[0:k]). \quad (\text{A.6})$$

The importance weights can be chosen as the following principle: Suppose $p(\mathbf{x}) \propto \pi(\mathbf{x})$ is a probability distribution, where $\pi(x)$ can be evaluated. If $\{\mathbf{x}^i[0:k]\}_{i=1}^N$ are

¹Note that the standard Kalman filter can be used as the optimal solution by the assumptions such as linearity and normal distribution.

distributed by $q(\mathbf{x})$, which is called the importance distribution, the importance weight $w^i[k]$ can be given by

$$w^i[k] \propto \frac{p(\mathbf{x}^i[0:k] | \mathbf{y}[0:k])}{q(\mathbf{x}^i[0:k] | \mathbf{y}[0:k])}. \quad (\text{A.7})$$

Since the random samples $\{\mathbf{x}^i[0:k]\}_{i=1}^N$ are drawn from the importance distribution $q(\cdot)$, any kind of distribution may be chosen as $q(\cdot)$. When we choose $q(\cdot)$ as a prior distribution such as $p(\mathbf{x}^i[k] | \mathbf{x}^i[k-1])$, the variance of importance weight can be minimized. Furthermore, given the choice of $q(\cdot)$, the implementation can be relatively easy. However, because the choice of $q(\cdot)$ does not take into account any observation, the performance may be limited.

The generic Bayesian particle filter is performed by the following five steps: the initialization, prediction, update, resampling, and iteration steps. These processes are visualized in Fig. A.1. After the initial state $\{\mathbf{x}_m^i[0:k]\}_{i=1}^N$, which may be known or randomly chosen, is determined, (a) the prediction step is performed at the time instance k by randomly distributing N number of samples (particles). (b) In the update step, given the probability distribution, the importance weights $\{w^i[k]\}_{i=1}^N$ are computed for every particle in order to determine the posterior distribution $p(\mathbf{x}[k] | \mathbf{y}[0:k])$. (c) The resampling step is performed to regenerate the computed

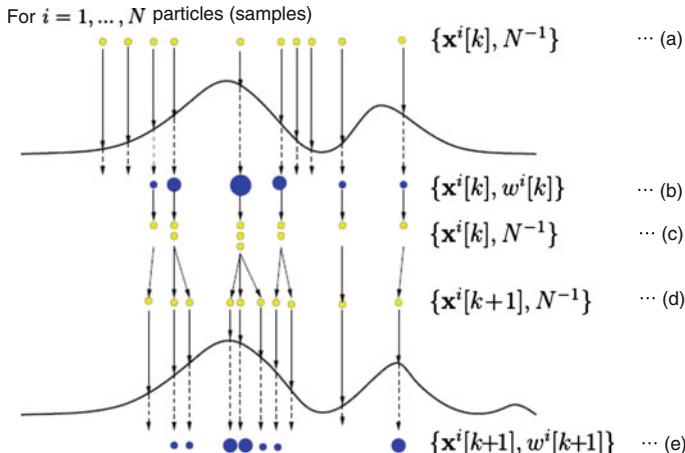


Fig. A.1 Graphical representation of the Bayesian particle filter: *a* prediction of the state at time k with unweighted samples, *b* weighted samples, *c* resampling, *d* prediction of the next state at time $k+1$ with unweighted samples, and *e* weighted samples at time $k+1$ [50]

samples based on the associated importance weights. Given the computed weights from (b), the samples with higher weights are concentrated and the samples with lower weights are discarded. Hence, a new set of the states $\{\mathbf{x}_m^i[k]\}_{i=1}^N$ is obtained. (d) The time instance is evolved to $k + 1$, and then unweighted N number of particles is randomly distributed as the prediction step. (e) The importance weights $\{w^i[k + 1]\}_{i=1}^N$ are computed for every particle to determine the posterior distribution $p(\mathbf{x}[k + 1] | \mathbf{y}[0:k + 1])$ as the update step again.

Appendix B

A Bayesian Approach for Multivariable Teleoperation Systems

B.1 Uplink Transmission

In order to obtain the estimates of multiple stochastic variables by using the proposed Bayesian approach, we first define a new state vector π_m , which is the following concatenated vector:

$$\pi_m[k_s] \triangleq (\mathbf{x}_m[k_s], n_{m[k]}). \quad (\text{B.1})$$

Then, the optimal solution in the Bayesian formulation is given by

$$\mathbb{E}(\pi_m[k_s] | \mathbf{x}_s[k_{d_0}:k_d]) = \iint \pi_m[k_s] p(\mathbf{x}_m[k_s], n_{m[k]} | \mathbf{x}_s[k_{d_0}:k_d]) d\mathbf{x}_m dn_m \quad (\text{B.2})$$

where $p(\mathbf{x}_m[k_s], n_{m[k]} | \mathbf{x}_s[k_{d_0}:k_d])$ is a joint posterior distribution conditioned on the observations $\mathbf{x}_s[k_{d_0}:k_d]$. According to the Bayesian approach, this joint posterior distribution can be computed recursively by solving the following prediction and update steps:

$$p(\mathbf{x}_m[k_s + 1], n_{m[k + 1]} | \mathbf{x}_s[k_{d_0}:k_d]) = \iint p(\mathbf{x}_m[k_s + 1], n_{m[k + 1]} | \mathbf{x}_m[k_s], n_{m[k]}) \\ \times p(\mathbf{x}_m[k_s], n_{m[k]} | \mathbf{x}_s[k_{d_0}:k_d]) d\mathbf{x}_m dn_m \quad (\text{B.3})$$

$$p(\mathbf{x}_m[k_s], n_{m[k]} | \mathbf{x}_s[k_{d_0}:k_d]) = \frac{p(\mathbf{x}_s[k_d] | \mathbf{x}_m[k_s], n_{m[k]}) p(\mathbf{x}_m[k_s], n_{m[k]} | \mathbf{x}_s[k_{d_0}:k_d - 1])}{p(\mathbf{x}_s[k_d] | \mathbf{x}_s[k_{d_0}:k_d - 1])} \quad (\text{B.4})$$

Using the Bayesian particle filter, the joint posterior distribution (B.4) is empirically approximated by drawing a set of randomly chosen weighted samples such that

$$\hat{p}(\mathbf{x}_m[k_s], n_m[k] \mid \mathbf{x}_s[k_{d_0}:k_d]) \approx \frac{1}{N} \sum_{i=1}^N \tilde{w}_m^i[k_s] \delta(\pi_m^i[k_s] - \pi_m^i[k_s]) \quad (\text{B.5})$$

where the importance weight $w_m^i[k_s]$ is given by

$$w_m^i[k_s] = \frac{p(\mathbf{x}_m^i[k_{s_0}:k_s], n_m^i[0:k] \mid \mathbf{x}_s[k_{d_0}:k_d])}{q(\mathbf{x}_m^i[k_{s_0}:k_s], n_m^i[0:k] \mid \mathbf{x}_s[k_{d_0}:k_d])} \quad (\text{B.6})$$

To avoid computing the whole previous trajectory of states, which may requires high computational complexity, we reformulate (B.6) in a recursive form by factorizing $q(\cdot)$ as

$$\begin{aligned} q(\mathbf{x}_m[k_{s_0}:k_s], n_m[0:k] \mid \mathbf{x}_s[k_{d_0}:k_d]) &= q(\mathbf{x}_m[k_s], n_m[k] \mid \mathbf{x}_m[k_s-1], n_m[k-1], \mathbf{x}_s[k_{d_0}:k_d]) \\ &\quad \cdot q(\mathbf{x}_m[k_{s_0}:k_s-1], n_m[0:k-1] \mid \mathbf{x}_s[k_{d_0}:k_d-1]) \end{aligned} \quad (\text{B.7})$$

After the numerator in (B.6) is written in a factorized form, a new recursive and time-update form of the importance weight is computed such that

$$w_m^i[k_s] = \tilde{w}_m^i[k_s-1] \frac{p(\mathbf{x}_s[k_d] \mid \mathbf{x}_m^i[k_s], n_m^i[k]) p(\mathbf{x}_m^i[k_s], n_m^i[k]) \mid \mathbf{x}_m^i[k_s-1], n_m^i[k-1]}{q(\mathbf{x}_m^i[k_s], n_m^i[k] \mid \mathbf{x}_m^i[k_s-1], n_m^i[k-1], \mathbf{x}_s[k_d])} \quad (\text{B.8})$$

This importance weight is associated with the state $\pi_m^i[k_s]$ at time k_s for i th particle. When we choose the importance distribution $q(\cdot)$ in (B.8) as a prior distribution for practical implementation of (B.5), we have

$$q(\cdot) = p(\mathbf{x}_m^i[k_s], n_m^i[k] \mid \mathbf{x}_m^i[k_s-1], n_m^i[k-1]). \quad (\text{B.9})$$

Then, (B.8) can be simplified as

$$w_m^i[k_s] = w_m^i[k_s-1] p(\mathbf{x}_s[k_d] \mid \mathbf{x}_m^i[k_s], n_m^i[k]). \quad (\text{B.10})$$

Using (B.10), we can only compute the joint likelihood distribution in (B.10) for practical implementation of the proposed Bayesian particle filter. A direct computation of (B.10) requires solving the joint distribution $p(\mathbf{x}_m^i[k_s], n_m^i[k])$. When we assume that the two stochastic variables are linearly combined with a joint normal distribution \mathcal{N} , the joint distribution can be written as

$$p(\mathbf{x}_m^i[k_s], n_m^i[k]) \sim \mathcal{N}(\mu_x, \mu_n, \sigma_x, \sigma_n, \rho_{xn}) \quad (\text{B.11})$$

where μ_x and μ_n are the means of states \mathbf{x}_m and n_m , respectively. σ_x and σ_n are the variances, and ρ_{xn} is the correlation coefficient between the two. Then, the distribution of true position with i number of particles can be obtained using the marginal property that averages over the information about distributions of n_m , such as

$$p(\mathbf{x}_m^i[k_s]) = \sum_{n_m} p(\mathbf{x}_m^i[k_s], n_m^i[k]). \quad (\text{B.12})$$

Alternatively, we can also assume that the processes of time-varying delays and transmitted position data are described by two different distributions. Hence, the joint likelihood distribution in (B.10) becomes

$$p(\mathbf{x}_s[k_d] | \mathbf{x}_m^i[k_d], n_m^i[k]) = p(\mathbf{x}_s[k_d] | \mathbf{x}_m^i[k_d]) p(\mathbf{x}_s[k_d] | n_m^i[k]). \quad (\text{B.13})$$

Practically, (B.13) may be a preferred approach to implement because we assume that time-varying delays are caused by a distance, queuing in router buffers, and congestion due to limited network bandwidth, rather than depend on the process of transmitting position data. However, if we assume that the process of transmitting position data cause network congestion, which may lead to increasing delays or packet losses, (B.11) can be also considerable.

B.2 Downlink Transmission

For the estimates of multiple stochastic variables in the downlink transmission by using the proposed Bayesian approach, we define a new state vector π_s , which is the following concatenated vector:

$$\pi_s[k'] \triangleq (\mathbf{f}_s[k'], n_s[k_s]). \quad (\text{B.14})$$

Similar to the uplink transmission, the optimal solution of (B.14) in the Bayesian formulation is given by

$$\mathbb{E}(\pi_s[k'] | \mathbf{f}_m[k_{ds_0}:k_{ds}]) = \int \int \pi_s[k'] p(\mathbf{f}_s[k'], n_s[k_s] | \mathbf{f}_m[k_{ds_0}:k_{ds}]) d\mathbf{f}_s dn_s. \quad (\text{B.15})$$

This optimal solution (B.15) can be obtained by solving the prediction and update steps, which are given as $p(\mathbf{f}_s[k'+1], n_s[k+1] | \mathbf{f}_m[k_{ds_0}:k_{ds}])$ and $p(\mathbf{f}_s[k'], n_s[k_s] | \mathbf{f}_m[k_{ds_0}:k_{ds}])$, respectively. Based on the Bayesian particle filter, the joint posterior distribution is empirically approximated by drawing a set of randomly chosen weighted samples such that

$$\hat{p}(\mathbf{f}_s[k'], n_s[k_s] | \mathbf{f}_m[k_{ds_0}:k_{ds}]) = \frac{1}{N} \sum_{i=1}^N \tilde{w}_s^i[k'] \delta(\boldsymbol{\pi}_s[k'] - \boldsymbol{\pi}_s^i[k']) \quad (\text{B.16})$$

The importance weight in (B.16) can be computed recursively by selecting the importance distribution $q(\cdot)$ as a prior distribution. Hence, the simplified form of the importance weight for the downlink transmission is given by

$$w_s^i[k'] = \tilde{w}_s^i[k' - 1] p(\mathbf{f}_m[k_{ds}] \mathbf{f}_s^i[k'], n_s^i[k_s]). \quad (\text{B.17})$$

The joint likelihood distribution in (B.17) can be obtained by either directly computing the joint distribution function $p(\mathbf{f}_s^i[k'], n_s^i[k_s])$ or assuming that transmitting force data and time-varying delays are described by different distribution models. After normalizing the importance weight as $\tilde{w}_s^i[k'] = w_s^i[k'] / \sum_{i=1}^N w_s^i[k']$, the resampling step needs to be performed to regenerate a new set of states $\{\mathbf{f}_s^i[k'], n_s[k_s]\}_{i=1}^N$ based on the associated importance weights.

Appendix C

Non-passive Conditions in the Presence of Network Delay

C.1 Non-passivity in the Presence of Constant Delay

Based on the interconnected two-port network structure shown in Fig. 3.2, a constant delay is introduced between the master and slave manipulator. When the constant delays in the uplink transmission and in the downlink transmission are n_m and n_s , respectively, the delayed power variables have the following relationships:

$$\begin{cases} \dot{\mathbf{x}}_s[k + n_m] = \dot{\mathbf{x}}_m[k] \\ \mathbf{f}_m[k] = \mathbf{f}_s[k - n_s] \end{cases}. \quad (\text{C.1})$$

From the total power flow (3.3), we consider only the power flow at the master side, such that

$$\mathbf{P}_h(t) - \mathbf{P}_m[k] = \dot{\mathbf{x}}_h^T(t)\mathbf{f}_h(t) - \dot{\mathbf{x}}_m^T[k]\mathbf{f}_m[k]. \quad (\text{C.2})$$

When the constant delay is introduced, (C.1) is substituted into (C.2) as

$$\mathbf{P}_h(t) - \mathbf{P}_m[k] = \dot{\mathbf{x}}_h^T(t)\mathbf{f}_h(t) - \dot{\mathbf{x}}_m^T[k]\mathbf{f}_s[k - n_s]. \quad (\text{C.3})$$

By a simple manipulation of the leaving power from the master manipulator, we have

$$\begin{aligned} \mathbf{P}_h(t) - \mathbf{P}_m[k] &= \dot{\mathbf{x}}_h^T(t)\mathbf{f}_h(t) - \dot{\mathbf{x}}_m^T[k]\{\mathbf{f}_m[k] + \mathbf{f}_s[k - n_s] - \mathbf{f}_m[k]\} \\ &= \underbrace{\dot{\mathbf{x}}_h^T(t)\mathbf{f}_h(t) - \dot{\mathbf{x}}_m^T[k]\mathbf{f}_m[k]}_A + \dot{\mathbf{x}}_m^T[k]\{\mathbf{f}_m[k] - \mathbf{f}_s[k - n_s]\}. \end{aligned} \quad (\text{C.4})$$

Since we generally assume that the individual master manipulator is a passive element, the part A in (C.4) is passive, which is $A \geq 0$. However, for the rest part of (C.4), some choices of force variables may have negative value, which results in negative dissipated power. A similar equation can be also derived for the power flow at the slave manipulator. Therefore, the constant delay between the master and slave manipulators may cause the non-passive condition of an overall teleoperation system.

C.2 Passivity of Scattering Transformation-Based Teleoperation System in the Presence of Constant Delay

From the generalized scattering transformation (3.5), a specific form, which is also called wave variables [19], can be obtained such that

$$\mathbf{S} = \frac{1}{\sqrt{2b}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}. \quad (\text{C.5})$$

Then, the re-configurations of the velocity and force variables at both master and slave manipulator sides become

$$\begin{aligned} \mathbf{u}_m[k] &= \frac{b \dot{\mathbf{x}}[k] + \mathbf{f}_m[k]}{\sqrt{2b}} & \mathbf{v}_m[k] &= \frac{b \dot{\mathbf{x}}[k] - \mathbf{f}_m[k]}{\sqrt{2b}} \\ \mathbf{u}_s[k] &= \frac{b \dot{\mathbf{x}}[k] + \mathbf{f}_s[k]}{\sqrt{2b}} & \mathbf{v}_s[k] &= \frac{b \dot{\mathbf{x}}[k] - \mathbf{f}_s[k]}{\sqrt{2b}} \end{aligned} \quad (\text{C.6})$$

From the total power flow (3.3), we only consider the power flow through the network, which is given by

$$\mathbf{P}_{\text{net}}[k] = -\dot{\mathbf{x}}_m^T[k]\mathbf{f}_m[k] + \dot{\mathbf{x}}_s^T[k]\mathbf{f}_s[k]. \quad (\text{C.7})$$

Then, (C.7) in terms of the scattering variables can be solved as

$$\mathbf{P}_{\text{net}}[k] = \frac{1}{2} \{ \mathbf{v}_m^2[k] - \mathbf{v}_s^2[k] + \mathbf{u}_s^2[k] - \mathbf{u}_m^2[k] \}. \quad (\text{C.8})$$

When the constant delays n_m is introduced in both uplink and downlink transmissions, the delayed scattering variables have the following relationships:

$$\begin{cases} \mathbf{v}_s[k] = \mathbf{v}_m[k - n_m] \\ \mathbf{u}_m[k] = \mathbf{u}_s[k - n_m] \end{cases} \quad (\text{C.9})$$

Substituting (C.9) into (C.8), Equation (C.8) becomes

$$\mathbf{P}_{\text{net}}[k] = \frac{1}{2} \{ \mathbf{v}_m^2[k] - \mathbf{v}_m^2[k - n_m] + \mathbf{u}_s^2[k] - \mathbf{u}_s^2[k - n_m] \}. \quad (\text{C.10})$$

From the power flow (C.10), the energy stored in the network up to the time instance k_c can be obtained as

$$\mathbf{H}_{\text{net}}[k_c] = \frac{T_s}{2} \sum_{k=k_c-n_m+1}^{k_c} (\mathbf{v}_m^2[k] + \mathbf{u}_s^2[k]) \quad (\text{C.11})$$

where T_s is the sampling rate. Since $\mathbf{H}_{\text{net}}[k_c] \geq 0$ in (C.11), it theoretically ensures a passive condition. Furthermore, it also shows that the stored energy within the network is independent of any magnitude of a constant delay n_m [18, 19].

C.3 Non-passivity of Scattering Transformation-Based Teleoperation System in the Presence of Time-Varying Delay

In the presence of time-varying delay, if we assume that the delay of the downlink transmission is increasing due to the congested network and the master manipulator holds the last received packet (e.g. Fig. 2.3b), then we may have the following power flow through the network:

$$\mathbf{P}_{\text{net}}[k] = \frac{1}{2} \{ \mathbf{v}_m^2[k] - \mathbf{v}_s^2[k] + \mathbf{u}_s^2[k] - \mathbf{u}_m^2[k-1] \}. \quad (\text{C.12})$$

Given the power flow through the network (C.8), the packet $\mathbf{u}_m^2[k-1]$ is the last received packet at the master manipulator since the current packet $\mathbf{u}_m^2[k]$ is not received yet due to the time-varying delay. By a simple manipulation of (C.12), we have

$$\begin{aligned} \mathbf{P}_{\text{net}}[k] &= \frac{1}{2} \{ \mathbf{v}_m^2[k] - \mathbf{v}_s^2[k] + \mathbf{u}_s^2[k] - \mathbf{u}_m^2[k-1] + \mathbf{u}_m^2[k] - \mathbf{u}_m^2[k] \} \\ &= \frac{1}{2} \underbrace{\{ \mathbf{v}_m^2[k] - \mathbf{v}_s^2[k] + \mathbf{u}_s^2[k] - \mathbf{u}_m^2[k] \}}_B + \mathbf{u}_m^2[k] - \mathbf{u}_m^2[k-1]. \end{aligned} \quad (\text{C.13})$$

From (C.8) to (C.11), the part B in (C.13) is passive, which is $B \geq 0$. However, for the rest part of (C.13), some choices of scattering variables may have negative value, which results in negative energy. Therefore, when the network delay increases, passivity of an overall teleoperation system may not be preserved, and this result is equivalent to the result obtained in [29], which was derived by a different formulation.

C.4 A Passive Condition of Scattering Transformation Based Teleoperation System in a Packet Loss

Given the power flow through the network (C.8), when the packet $\mathbf{u}_m^2[k]$ is lost during the network transmission and the unreceived packet at the master manipulator becomes zero (e.g. Fig. 2.3c, then we may have the following power flow through the network:

$$\mathbf{P}_{\text{net}}[k] = \frac{1}{2} \{ \mathbf{v}_{\mathbf{m}}^2[k] - \mathbf{v}_{\mathbf{s}}^2[k] + \mathbf{u}_{\mathbf{s}}^2[k] \}. \quad (\text{C.14})$$

By a simple manipulation of (C.14), we have

$$\begin{aligned} \mathbf{P}_{\text{net}}[k] &= \frac{1}{2} \{ \mathbf{v}_{\mathbf{m}}^2[k] - \mathbf{v}_{\mathbf{s}}^2[k] + \mathbf{u}_{\mathbf{s}}^2[k] + \mathbf{u}_{\mathbf{m}}^2[k] - \mathbf{u}_{\mathbf{m}}^2[k] \} \\ &= \frac{1}{2} \underbrace{\{ \mathbf{v}_{\mathbf{m}}^2[k] - \mathbf{v}_{\mathbf{s}}^2[k] + \mathbf{u}_{\mathbf{s}}^2[k] - \mathbf{u}_{\mathbf{m}}^2[k] + \mathbf{u}_{\mathbf{m}}^2[k] \}}_C. \end{aligned} \quad (\text{C.15})$$

From (C.8) to (C.11), the part C in (B.15) is passive, which is $C \geq 0$. For the rest part of (C.15), we have only positive scattering variable, so that $\mathbf{P}_{\text{net}}[k] \geq 0$. Therefore, in the case of packet loss, passivity may be maintained because a null packet basically does not introduce extra energy to an overall system [26].

Appendix D

Packet Structures of Standard Transport Protocols

D.1 Packet Structure of UDP

See Figs. D.1 and D.2.

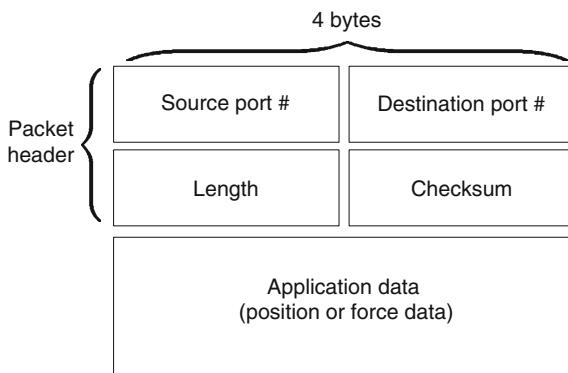


Fig. D.1 Packet structure of standard UDP. *Source port number* This field identifies the sender's port number. *Destination port number* This field identifies the receiver's port number. *Length* This field specifies the length in bytes of the entire packet including header and data. *Checksum* This field is used for error-checking of the header and data

D.2 Packet Structure of TCP

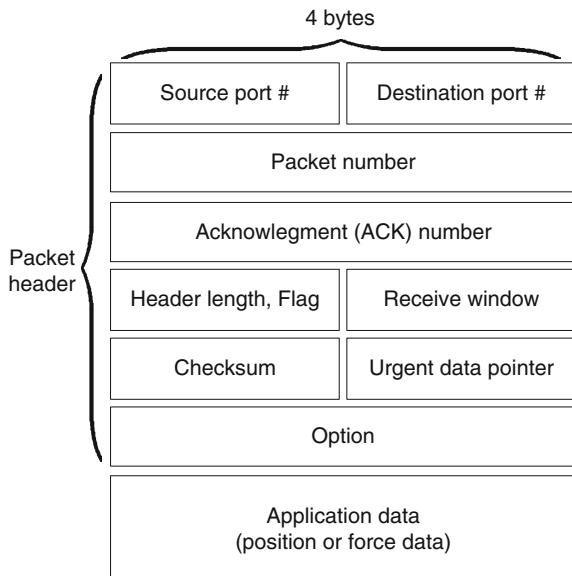


Fig. D.2 Packet structure of standard TCP. *Packet number* From the sender, each packet is assigned by the unique number. *ACK number* The receiver acknowledges the packet number from the sender. *Header Length* The length in bytes of the packet. *Flag* 6-bit flags are used to indicate ACK, connection setup, teardown, urgent. *Receive window* Used for flow control that indicates the number of bytes the receiver accepts. *Urgent date pointer* Indicating the location of urgent data. *Option* The sender and receiver can negotiate the packet size, window size, and time stamp, and etc.

Appendix E

Performance of Low-Pass Filter

The low-pass filters were performed without using the proposed methods in the transport and application layers under the delay profile #2, and the results are shown in Fig. E.1. Since the low-pass filter was the process to compute a moving average only, it was unable to compensate for unreliable network conditions as shown in these figures.

The low-pass filters were performed after the estimation-based Bayesian particle filters, and the results of 1-DoF position and force data are shown in Fig. E.2. As shown in these figures, we could observe that low jitter still remained in the position and force data, but the quality of resulting waveforms was improved compared to Fig. E.1.

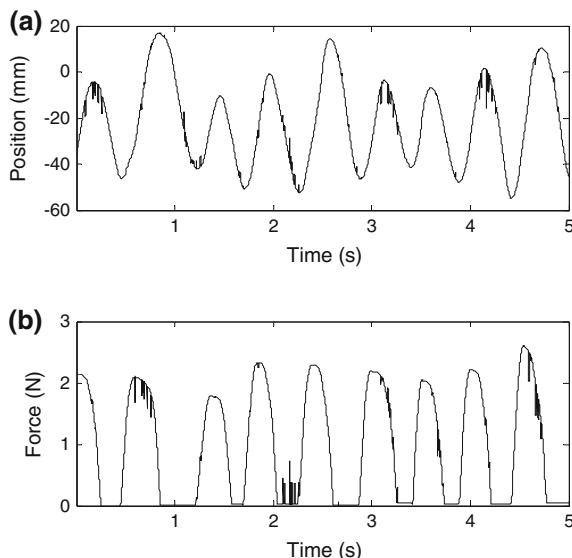


Fig. E.1 Low-pass filtering under the delay profile #2: **a** position data, and **b** force data

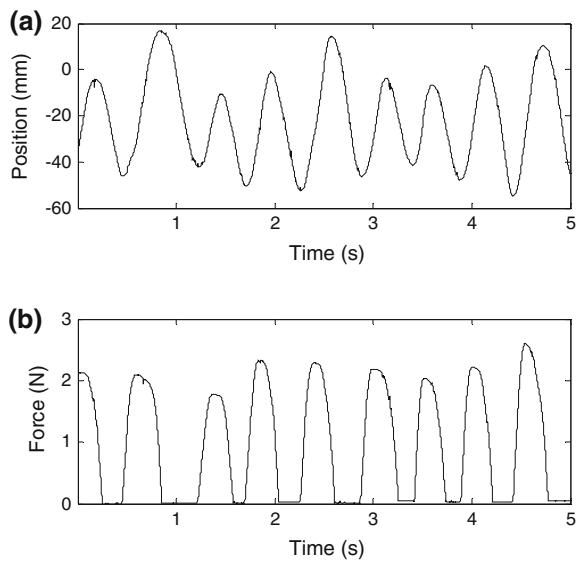


Fig. E.2 Low-pass filtering after the estimation-based Bayesian particle filters under the delay profile #2: **a** position data, and **b** force data

Appendix F

Basics of Data Compression

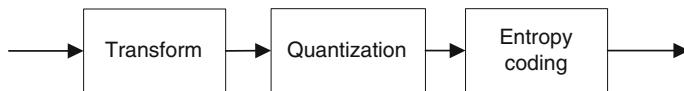
In data compression, such as video, image, and audio compression, there are two types of compression methods: the lossless compression and the lossy compression. In the case of lossless compression, the reconstructed data and the uncompressed source data are identical. Since such compression method does not allow any loss, the compression ratio is significantly low in general. The widely used lossless compression methods are ZIP, Huffman, and arithmetic coding. On the other hand, lossy compression methods discard information that may not be noticeable by human perception, and hence, the reconstructed data and the source data are not identical. Lossy compression methods are widely used for multimedia systems due to high compression ratio. The standard video and image codecs, such as JPEG and MPEG, are mostly based on the lossy compression. The widely used lossless and lossy compression methods in multimedia systems are listed in Tables F.1 and F.2, respectively.

Table F.1 Lossless compression methods

Type of method	Examples	Description
Entropy coding	Huffman, arithmetic codes	<ul style="list-style-type: none">✓ Assign a unique code to each data symbol✓ Frequent symbols have shorter codeword and rare ones have longer codeword✓ Compression is achieved by using the similarity between bit streams

Table F.2 Lossy compression methods

Type of method	Examples	Description
Transform coding	Discrete cosine transform (DCT), Wavelet transform	<ul style="list-style-type: none"> ✓ Most energy of a block can be transformed to a few coefficients ✓ Small coefficients can be discarded without impacting the perceptual quality ✓ Large coefficients can be further compressed with other methods
Predictive coding	Linear, nonlinear	<ul style="list-style-type: none"> ✓ Given the current and previous samples, the next sample can be predicted by a variety of prediction algorithm ✓ High compression ratio can be achieved depending on the performance of the predictor
Quantization	Uniform, non-uniform	<ul style="list-style-type: none"> ✓ The process of representing a large set of data with a smaller set
Down-sampling	Fixed rate, adaptive rate	<ul style="list-style-type: none"> ✓ The process of reducing the sampling rate

**Fig. F.1** A basic compression process in multimedia systems

As an example, a general compression process of multimedia systems is shown in Fig. F.1. In the transform coding, most energy of a block can be transformed to a few coefficients. In the quantization process, smaller coefficients by the transform coding can be discarded. Before transmitting through the network, the entropy coding is performed to assign shorter binary code to more frequent data symbols and longer code to rare data symbols. At the receiver, the inverse process of these methods needs to be performed to reconstruct the data. The predictive coding can be performed at the receiver for further enhancement of compression performance.

Appendix G

Implementation of Virtual Slave Manipulator

For the virtual environment implementation based on the Phantom Omni haptic device, the OpenHaptics library provides Haptic Device API (HDAPI) and the Haptic Library API (HLAPI). The HLAPI provides high-level haptic rendering, which can be mostly designed with OpenGL® graphic library. This library allows the significant reuse of existing OpenGL codes and greatly simplifies synchronization of the haptic and graphics threads. On the other hand, the HDAPI provides low-level access to the haptic device based on control method, and it enables programmers to render force information directly. Because of such low-level access implementation, the virtual slave manipulator in this experimental setup was designed by the HDAPI haptic rendering.

The 3D virtual slave manipulator implemented for the experimental setup is shown in Fig. G.1. We implemented the selective compliant assembly robot arm (SCARA) model, which had three rotational joints and one sliding joint at the end-effector. The dimension of the robotic arm was indicated in the sketch map 2.01. With the haptic device as a master manipulator, the kinematic and force interface of the virtual slave manipulator were designed. While the human operator moved the end-effector of the master manipulator, the end-effector of virtual slave manipulator tracked the positions from its own current position in accordance with the incremental position of the master manipulator. For the rest of joints of the slave manipulator, we derived the inverse kinematic solution. When the end-effector positions $\{x, y, z\}$ and the first two links $\{L_1, L_2\}$ in the figure were obtained by the HDAPI functions, the first two joint angles θ_1, θ_2 were given by

$$\begin{aligned}\theta_1 &= \arctan\left(\frac{x}{z}\right) + \arccos\left(\frac{\sqrt{x^2 + z^2}}{2L_1}\right) \\ \theta_2 &= \arccos\left(\frac{L_1^2 + L_2^2 - x^2 - z^2}{2L_1 L_2}\right) - 180\end{aligned}$$

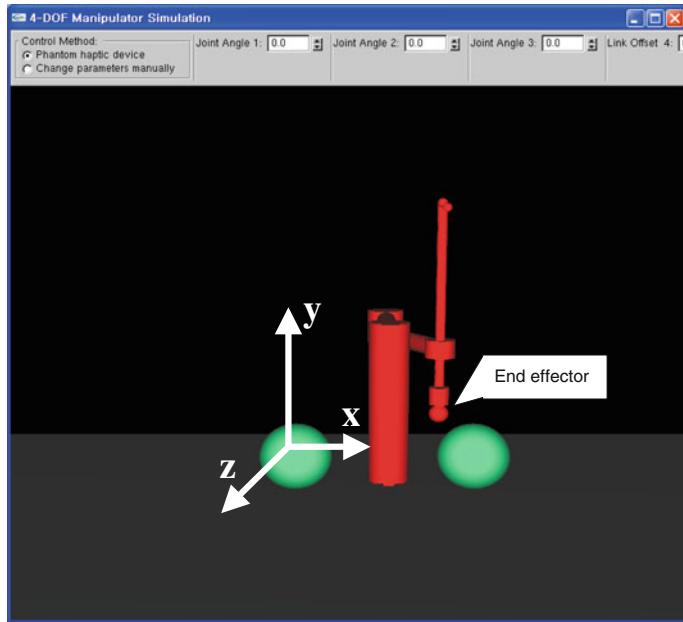


Fig. G.1 The implementation of 3D virtual slave manipulator. The SCARA model consists of three rotational joints and one sliding joint. Given 3-DoF positioning input from the master manipulator, 3-DoF forces are generated and fed into the master manipulator

The last joint θ_3 was directly given by the master manipulator since we defined the third link as $L_3 = 0$. Finally, the last sliding joint d_4 was defined as the direction of y -axis in the OpenGL frame. Hence, the last joint was simply given by $d_4 = y$.

In order to perform a haptic teleoperation task, two spherical objects were implemented. When the end-effector of the virtual slave manipulator was in contact with the spherical objects, the human operator felt the forces between the end-effector and the simplified link representations of the slave manipulator. The end-effector of slave manipulator was defined to have a spherical shape. Hence, by controlling the master manipulator, the human operator could slide on the surface of the spherical objects, and the magnitude and direction of 3-DoF computed reaction force vector were felt at the human operator's hand.

Appendix H

Error Correcting Codes

In coding and information theory, forward error correction (FEC) or channel coding is a technique that detects, correct, and control erroneous data, which are transmitted over unreliable network condition, such as noisy link, packet loss, or even fading channel in wireless transmission. In block codes, which use fixed size of bits or symbols for encode and decode processes, the classical codes include the RS code, Golay, BCH, Multidimensional parity, and Hamming codes. These codes are in general implemented using hard-decision algorithms. The hard-decision means that the decision of input and output data is made by one or zero bit. These classical block codes are listed in Table H.1.

Table H.1 Classical block codes

Type of codes	Applications	Description
RS code	Broadcasting (ATSC, DVB), telecommunication, disk, DVD	<ul style="list-style-type: none">✓ Most widely used in numerous digital applications✓ Generally suitable burst bit errors✓ Used with other FEC codes (e.g. convolution code)
Golay code	High frequency radio systems, ultrasound, WiFi	<ul style="list-style-type: none">✓ Binary Golay code can detect 7-bit and correct 3-bit errors by encoding 12 bits data✓ Ternary Golay code is a linear code over a ternary alphabet
BCH code	Satellite communication, DVD, barcode	<ul style="list-style-type: none">✓ Controlling the number of correctable errors is capable✓ Binary BCH code can correct multiple bit errors✓ Designing a decoder is relatively easy
Hamming code	NAND flash memory	<ul style="list-style-type: none">✓ A linear code that can detect up to two and correct up to one bit errors

Table H.2 High performance FEC codes

Type of codes	Applications	Description
Low density parity check (LDPC)	Broadcasting (DVB-S2), WiMAX, high speed wireless LAN, 3G wireless	<ul style="list-style-type: none"> ✓ Highly efficient linear block code ✓ Closely perform the theoretical maximum by using iterated soft-decision decoding ✓ Due to the complexity, huge parallel computation is necessary for practical implementation
Turbo code	Wireless standards (CDMA 2000), satellite communication	<ul style="list-style-type: none"> ✓ Iterated soft-decoding that combines two or more relatively simple convolutional codes and an interleaver ✓ Work as a block code by approaching the theoretical maximum performance
Raptor code	Broadcasting (DVB-H), IP-based system (IPTV)	<ul style="list-style-type: none"> ✓ Developed for an application layer FEC method ✓ Encode a message into a potentially limitless sequence of encoding symbols ✓ Probability that message can be recovered may increase to 1

In general, these classical block codes are used as concatenated forms with convolutional codes in order to achieve the performance improvement. The convolutional codes are mostly decoded by the Viterbi algorithm. This algorithm provides the optimal decoding with increasing length of codeword, which may require the exponentially increasing complexity. A convolutional code can be implemented using soft-decision algorithms. The soft-decision means that a decoder can process analog type of signal, and hence, much improved error correction performance can be achieved compared to the hard-decision based block codes.

Although above classical block and convolutional codes are widely used in current digital applications, recent advances in hardware technology have enabled more complex codes in order to achieve further performance improvement. The recently developed high performance FEC codes are listed in Table H.2.