

IEEE Standard for Haptic Codecs for the Tactile Internet

IEEE Communications Society

Developed by the
Standards Development Board

IEEE Std 1918.1.1™-2024



STANDARDS

IEEE Standard for Haptic Codecs for the Tactile Internet

Developed by the

Standards Development Board
of the
IEEE Communications Society

Approved 15 February 2024

IEEE SA Standards Board

Abstract: Haptic codecs for the data reduction of kinesthetic and tactile signals are described in this standard. Three types of codecs are presented in this document, including the no-delay kinesthetic codec (Part I), the delay-robust kinesthetic codec (Part II), and the tactile codec (Part III). Additionally, this document introduces handshaking and metadata exchange mechanisms for the exchange of the capabilities of haptic devices (Part IV). This standard also provides additional information (in informative annexes) for codec implementation and evaluation, such as the reference software and hardware setups for kinesthetic and tactile codecs, and cross-validation results.

Keywords: control scheme, haptic codecs, IEEE 1918.1.1™, kinesthetic codec, perceptual deadband approach, stability, tactile codec

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2024 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 14 June 2024. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

W3C® is a trademark held by, and registered in numerous countries by the World Wide Web Consortium.

PDF: ISBN 979-8-8557-0694-9 STD26918
Print: ISBN 979-8-8557-0695-6 STDPD26918

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus balloting process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and

¹ Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

² Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.

adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

³ Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴ Available at: <https://standards.ieee.org/standard/index.html>.

⁵ Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this IEEE standard was completed, the Haptic Codec Task Group of the Tactile Internet Working Group had the following membership:

Eckehard Steinbach, *Chair*

Mohamad Eid, *Vice Chair*

Qian Liu, *Secretary*

Stefan Aust
Ren Chen
Vineet Gokhale
Basak Gülecü
Rania Hassen
Oliver Holland

Thomas Hulin
Shu-Chen Li
Xun Liu
Evelyn Muschter
Yeshwant Muthusamy

Andreas Noll
Michael Panzirsch
Venkatesha Prasad
Harsimran Singh
Xiao Xu
Zhou Yu

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Carol Ansley
Butch Anton
Stefan Aust
Paul Cardinal
Matthias Fritsche
Marco Hernandez
Werner Hoelzl
Pranav Jha
Piotr Karocki
Stuart Kerry
Yongbum Kim
Hyeon Ho Lee

Daozhuang Lin
Qian Liu
Sajeev Manickoth
Rajesh Murthy
Mark Nixon
Sharief Oteafy
Bansi Patel
Arumugam Paventhan
Venkatesha Prasad
A.S.M. Sharifuzzaman Sagar
Cole Scott

Eckehard Steinbach
Bo Sun
James Van De Ligt
John Vergis
Stephen Webb
Hung-Yu Wei
Scott Willy
Andreas Wolf
Chi Xu
Xiao Xu
Yu Yuan
Oren Yuen

When the IEEE SA Standards Board approved this standard on 15 February 2024, it had the following membership:

David J. Law, *Chair*

Vacant Position, *Vice Chair*

Gary Hoffman, *Past Chair*

Alpesh Shah, *Secretary*

Sara R. Biyabani
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Guido R. Hiertz
Ronald W. Hotchkiss
Hao Hu

Yousef Kimiagar
Joseph L. Koepfinger*
Howard Li
Xiaohui Liu
John Haiying Lu
Kevin W. Lu
Hiroshi Mano

Paul Nikolich
Robby Robson
Mark Siira
Lei Wang
F. Keith Waters
Sha Wei
Philip B. Winston
Don Wright

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1918.1.1-2024, IEEE Standard for Haptic Codecs for the Tactile Internet.

As haptic technologies are becoming increasingly available and diversified, supporting both kinesthetic and tactile interaction in a wide range of applications (e.g., tele-operation, gaming and entertainment, automation, and robotics), the need has arisen for the definition of a standard for the compression of haptic data. This is the main target of the Haptic Codec Task Group of the Tactile Internet Working Group.

For haptic interaction across a communication network, typically two types of information need to be exchanged, namely kinesthetic and tactile information. This standard defines haptic codecs for the Tactile Internet (TI), which address TI application scenarios where the human is in the loop (i.e., teleoperation or remote touch applications) as well as scenarios that rely on machine remote control. More specifically, the standard defines (perceptual) data reduction algorithms and schemes for both closed-loop (kinesthetic information exchange) and open-loop (tactile information exchange) communications.

In particular, Part I of this standard describes the no-delay kinesthetic codec (KCnD), which is based on the perceptual deadband approach derived from Weber's law of just noticeable differences (JND).

Part II of this standard describes the delay-robust kinesthetic codec (KCwD), which includes a stabilizing control approach for time-delayed teleoperation.

Part III of this standard describes the tactile codec (TC), which is based on sparse linear prediction (SLP) coding via modeling the human sensitivity characteristics for a given vibrotactile signal.

Part VI of this standard describes mechanisms and protocols for the exchange of the capabilities (e.g., workspace, the number of degrees of freedom, amplitude range) of the haptic devices.

This document also includes informative annexes to provide necessary information for codec implementation and result evaluation, such as the reference setup for KCs, cross-validation results for KCnDs, the reference setup for TC, and the multiplexing scheme for simultaneous transmission of video, audio, and haptic signals.

Contents

1. Overview.....	11
1.1 Scope.....	12
1.2 Purpose.....	13
1.3 Word usage.....	13
2. Normative references.....	13
3. Definitions, acronyms, and abbreviations	13
3.1 Definitions	13
3.2 Acronyms and abbreviations.....	15
3.3 Variables used in this document.....	16
4. Part I: No-delay kinesthetic codec (KCnD).....	17
4.1 Bitstream syntax.....	18
4.2 Kinesthetic decoder	19
5. Part II: Delay-robust kinesthetic codec (KCwD).....	20
5.1 KCwD decoder	22
5.2 KCwD encoder	27
6. Part III: Tactile codec.....	28
6.1 PVC-SLP decoder	28
6.2 Tactile codec dictionaries	33
7. Part IV: Plug-and-play (PnP) haptic interaction for Tactile Internet.....	34
7.1 Haptic handshake	35
7.2 Haptic metadata	35
7.3 Handshake and data packets for the standard codecs	37
Annex A (normative) DCT coefficient dictionaries (D_{DCT}).....	42
Annex B (informative) Reference setup for kinesthetic codecs.....	98
B.1 Reference setup for KCnD.....	98
B.2 Reference setup for KCwD	99
Annex C (informative) Reference encoder, experimental results, and cross-validation results for KCnD	103
C.1 Reference encoder	103
C.2 Experimental results	104
C.3 Cross-validation results.....	105
Annex D (informative) Experimental and cross-validation results for KCwD.....	109
D.1 Implementation	109
D.2 Cross-validation results	109
Annex E (informative) Reference setup for tactile codec	111
E.1 Reference setup for acquisition and presentation of vibrotactile signals.....	111
E.2 Reference setup for tactile codec	111

Annex F (informative) Reference encoder, experimental, and cross-validation results for the tactile codec	113
F.1 Reference encoder.....	113
F.2 Experimental results.....	116
F.3 Cross-validation implementation	116
F.4 Cross-validation results	117
Annex G (informative) Implementation of the PnP haptic interaction for Tactile Internet	121
G.1 Implementation on WebRTC	121
G.2 Tele-writing demonstration.....	123
Annex H (informative) Bibliography.....	124

IEEE Standard for Haptic Codecs for the Tactile Internet

1. Overview

The key objective of the Tactile Internet (TI) is to allow remote physical interaction with convincing haptic experiences and enable fully immersive multi-sensory remote exploration of real or virtual environments where users can see, hear, and in particular, feel remote objects. In this context, the signal transmitted in addition to the audio/visual signals is the haptic information, which refers to both kinesthetic perception (forces, torques, position, velocity, etc., sensed by the muscles, joints, and tendons of the body) and tactile perception (surface texture, friction, etc., sensed by different types of mechanoreceptors in the skin). Together the users are able to perceive the shape and position of objects, the texture, stiffness, friction, temperature, and vibrations, as illustrated in Figure 1.

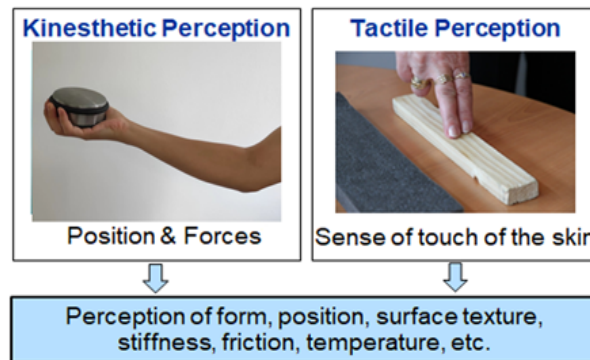


Figure 1—Haptic information consisting of two submodalities, i.e., kinesthetic and tactile perception

Depending on different TI application scenarios, the kinesthetic perception or the tactile perception or a combination of both form the input to the haptic codecs. In haptic technologies, the kinesthetic and tactile perception are often considered independently when different sensing and actuation principles are applied. Therefore, two typical use cases are presented in the following that solely rely on the kinesthetic or tactile information exchange.

The bilateral haptic teleoperation system is a typical use case with kinesthetic information exchanged bidirectionally over the TI. As shown in Figure 2, three main components are involved: the human system interface (HSI on the leader side), the teleoperator (TOP on the follower side), and the communication network connecting them. The HSI acts both as the data input (position and velocity information) and output device (video, audio, and kinesthetic feedback).

The local control loops at the leader and follower are coupled via the communication network into a global closed control loop. As a result, system stability and force feedback quality are strongly affected by communication delay. Additionally, system stability requires a high sampling rate of 1 kHz or even higher

for the kinesthetic signals. Hence, teleoperation systems generally require 1000 or more haptic data packets per second to be transmitted bidirectionally between the leader and the follower device in addition to the audio and video streams, which results in a heavy burden to the communication network. Therefore, kinesthetic codecs (KCs) should be employed in both leader and follower sides to tackle this issue by performing kinesthetic data reduction.

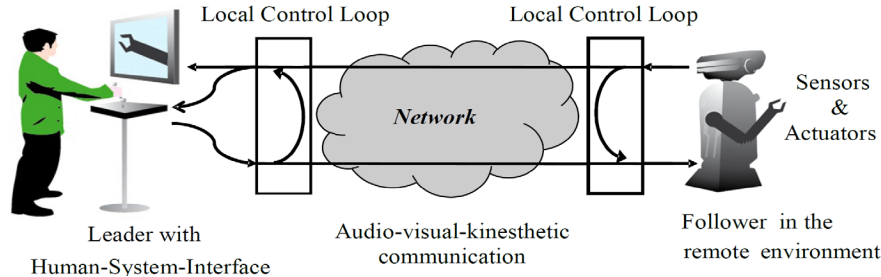
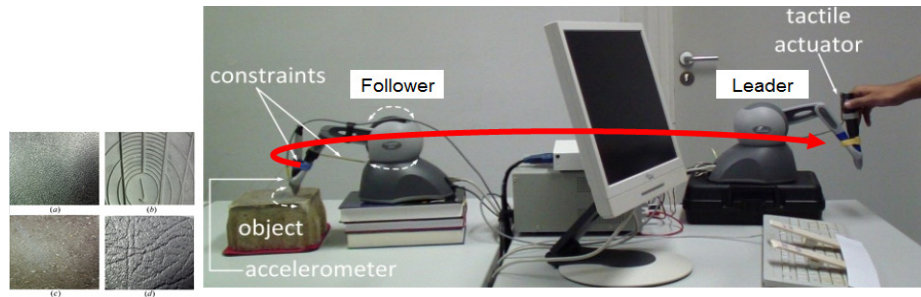


Figure 2—Example of a bilateral haptic teleoperation system with kinesthetic interaction

Different from the close-loop nature of kinesthetic data exchange, the communication of tactile information is typically open loop, as shown in Figure 3, where the haptic device in the follower side is teleoperated by the leader to explore the surface of object.⁶

In this context, the fine vibrations, also known as *vibrotactile signals* are solely transmitted from the follower to the leader.

The tactile data size of a single point of interaction is much smaller than that of video and is comparable to that of speech signals. However, a large number of tactile sensors is expected to be deployed in future haptic communication systems, thereby it is essential to develop tactile codecs to help ensure high-fidelity tactile interaction.



Copyright © 2014 IEEE. All rights reserved. Used with permission from Strese et al. [B24].

Figure 3—Example of the open-loop communication of tactile information in teleaction

1.1 Scope

This standard defines haptic codecs for the TI. These codecs address TI application scenarios where the human is in the loop (i.e. teleoperation or remote touch applications) as well as scenarios that rely on machine remote control. The standard defines (perceptual) data reduction algorithms and schemes for both closed-loop (kinesthetic information exchange) and open-loop (tactile information exchange) communication. These codecs are designed such that they can be combined with stabilizing control and local communication architectures for time-delayed teleoperation. Further, the standard also specifies mechanisms

⁶ The numbers in brackets correspond to those of the bibliography in Annex H.

and protocols for the exchange of the capabilities (e.g., workspace, the number of degrees of freedom, amplitude range, temporal and spatial resolution) of the haptic devices.

1.2 Purpose

This standard provides haptic codecs for the TI enabling the interoperability of different haptic (kinesthetic and tactile) input and output devices. Such interoperability is necessary to achieve necessary market scale in the realization of TI technologies, devices, and applications.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{7, 8}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

There are no normative references in this standard.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.⁹

algorithmic encoder/decoder delay: Processing delay introduced by a kinesthetic encoder/decoder.

closed-loop kinesthetic interaction: While interacting with a real or virtual environment, kinesthetic signals are transmitted from leader to follower side, and the resultant kinesthetic feedback signals are transmitted from follower to leader side.

codec complexity: Measure of runtime execution complexity of a codec.

⁷ The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁸ The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

⁹ *IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

degrees-of-freedom: The ability of the kinesthetic haptic interface to read or display kinesthetic signals in a single independent direction in 3D space.

dynamic interaction: Single and/or multi-point interaction with a remote or virtual object that can undergo motion.

haptic device/interface: This provides its position and movement signals as the output to controlling a remote agent, and receives force, torque, or vibrotactile feedback for displaying.

kinesthetic codec (KC): Adapts the kinesthetic signals to the network conditions including limited transmission capacity, network delay/jitter, packet loss, etc.

kinesthetic signals: Signals that are necessary for the perception of position/orientation and external forces/torques applied. Hence, position, velocity, angular velocity, force, and torque all fall into the category of kinesthetic signals.

lossless encoding: Original signals and reconstructed signals are identical, i.e., there is no reconstruction error.

lossy encoding: The reconstructed signal that is allowed to deviate from the original signal.

multi-point interaction: The kinesthetic haptic interface supports more than one point of contact for the interaction with a real or virtual environment. Multi-point interaction, for instance, is relevant for grasping an object.

objective quality evaluation: Codec performance is evaluated using a mathematical evaluation metric.

offline encoding: The codec encodes recorded kinesthetic signal traces. The encoding speed does not have to be in real time.

online encoding: The codec is capable of encoding kinesthetic signals fast enough to support a closed-loop kinesthetic interaction.

open-loop tactile interaction: Interacting with real or virtual environment, the vibrotactile signals are transmitted in one direction (from follower side to leader side)

perceptual tactile codec: An algorithm that integrates models of human tactile perception into the design of coding procedure

position drift: Position mismatch between the leader and the follower. It can lead to task failure or dangerous action on the follower side. Position drifts are quantified by applications according to applied coordinate systems. Losing a packet that contains position commands, compressing the velocity signals, or revising the position/velocity commands in the passivity controller will lead to position drift.

single-point interaction: Kinesthetic haptic interface provides a single point of contact for the interaction with a real or virtual environment.

stabilizing control: While interacting with a real or virtual environment, there exists a bidirectional exchange of kinesthetic signals between the leader and follower side. This closes a control loop between leader and follower, which can get destabilized in the presence of communication delay. Stabilizing control architectures confirms the stability of the control loop.

static interaction: Single and/or multi-point interaction with a stationary remote or virtual object.

subjective quality evaluation: Codec performance is evaluated using a panel of human test subjects.

system passivity: A system is passive if it cannot produce energy on its own. A system is stable if it is passive.

system stability: A control system is stable if its trajectories are bounded by a function of the size of the input for all sufficiently large times.

tactile frame: Portion of the vibrotactile signal consisting of 200 samples

vibrotactile sampling frequency, f_s : The rate used to digitize the acquired vibration signal using analog sensors.

vibrotactile signal: High-frequency vibrations induced when exploring textured surfaces either using bare-fingers or tool-mediated interaction. It pertains to the roughness submodality of human tactile perception

3.2 Acronyms and abbreviations

ASF	acceleration sensitivity function
DCT	discrete cosine transform
DDCT	Huffman Dictionary of Discrete Cosine Transform Coefficients
DRC	Huffman Dictionary of Reflection Coefficients
KC	kinesthetic codec
KCnD	kinesthetic codec with no delay
KCwD	kinesthetic codec with delay
LPC	linear predictive coding
PnP	plug and play
PVC-SLP	perceptual vibrotactile codec based on sparse linear prediction
Q / DeQ	quantization / de-quantization
RC	reflection coefficients
SLP	sparse linear prediction
TC	tactile codec
TI	Tactile Internet
ZRL	zero-run-length

3.3 Variables used in this document

Kinesthetic codec—KC

Variable	Explanation
x_m	Leader position
v_m	Leader velocity
x_s^r	Most recently received position at the follower side
v_s^r	Most recently received velocity at the follower side
$\hat{v}_s$	Reconstructed velocity at the follower side
f_s	Follower force to be transmitted to the leader
$\hat{f}_m$	Reconstructed force at the leader side
f_s^c	Follower control force after the passivity controller. It is also the force applied to the follower device
f_m^r	Leader received/desired force before the passivity controller (PC). It is the reconstructed follower control force f_s with delay T_2
f_m	Leader force after the passivity controller. Force applied to the leader device
T_1	Forward communication delay
T_2	Backward communication delay
α	Adaptive damping to dissipate excessive energy on the leader side
β	Adaptive damping to dissipate excessive energy on the follower side
P_i	Power at port i
E_i	Energy at port i
T_s	Sampling time
W^{PC1}	Energy to be dissipated at the passivity controller 1
$E^{R2L,des}$	Desired maximum energy output for the leader. Transmitted from right to the left.
W_{diss}^{PC1}	Energy dissipated at the passivity controller 1
p	Deadband parameter

Tactile Codec—TC

Variable	Explanation
f_s	Sampling frequency
δ_{RC}	RCs quantization scale
γ_Q	Quantized RCs vector
γ	RCs vector
$\hat{\gamma}$	Dequantized RCs vector
δ	DCT coefficients quantization scale
$e(k)$	k^{th} quantized DCT coefficient
$\hat{e}(k)$	k^{th} dequantized DCT coefficient
$A_{ASF}(k)$	ASF value at k^{th} DCT coefficient frequency
r	Residual signal
$\tilde{r}$	Reconstructed residual signal
x	Vibrotactile signal
$\tilde{x}$	Reconstructed vibrotactile signal

4. Part I: No-delay kinesthetic codec (KCnD)

This clause presents a kinesthetic codec (KC) for the TI without considering the communication delay between the leader and the follower system. The defined KC is able to reduce the packet rate of kinesthetic information while maintaining a high quality of experience (e.g., indistinguishable local and remote operations).

Kinesthetic data involves one or more streams of information relevant for the kinesthetic interaction with a remote real or virtual environment involving the bidirectional exchange of one or more kinesthetic information streams. Taking bilateral teleoperation with haptic feedback as an example, as illustrated in Figure 2, kinesthetic information in this context relates to information relevant for the human kinesthetic perceptual mechanisms, which allow for the perception of the position and orientation of our body parts and joints and external forces and torques applied to them. Hence, position, velocity, angular velocity, force, and torque all fall into the category of kinesthetic information. The corresponding data streams can be of different dimensionality, typically ranging from one dimension to six dimensions for a single interaction point. The exact values of these kinesthetic data streams depend on the interaction pattern, the object properties (stiffness, friction, etc.) and need to be captured for real-world interaction using appropriate sensors or rendered in the case of kinesthetic interaction in a virtual environment.

In haptic teleoperation systems, a high sampling rate (1000 Hz to 4000 Hz) is necessary to ensure the tracking performance and stability of the local control loops at the leader and the follower. Currently available teleoperation systems can easily have more than 20 degrees of freedom (DoFs), e.g., for a bimanual teleoperation system with additional finger position sensing and force feedback. To minimize the end-to-end delay, typically at every sampling instant a data packet consisting of all current sample values is generated and sent out. The resulting high packet rate is hard to maintain in communication networks. For efficient transmission, data reduction techniques have to be used to lower the amount of data and in our case

the rate at which packets have to be transmitted. Lower packet rates put less demand on the network connection and therefore reduce the probability of congestion along the transmission path.

In this clause, the KC for the no-delay scenario (KCnD) is defined. This codec does not contain a control scheme to cope with the stability of the teleoperation in the presence of communication delay. In fact, there are a number of teleoperation applications that do not require geographically distributed leader and follower systems, e.g., telesurgery on site, completely local interaction with metaverse (VR) environments. In these cases, communication delay is negligible and the KCnD setup provides a simple solution that does not require to apply additional control schemes and transmit extra control parameters. Moreover, for future extension, developers can also run their own control architecture on top of the KCnD. For this case, defining a separate KC for the no-delay case is beneficial.

The KCnD defined in this standard is based on the perceptual deadband (PD) approach, which is inspired by Weber's law of just noticeable differences (JND) for kinesthetic information. For the PD approach, only if the relative difference between two stimuli exceeds a certain perceptual threshold (the perceptual deadband or PD for short), the signal will be transmitted.

In this part of the standard, only the kinesthetic decoder is specified. Encoder designers are left free to design their own encoding algorithms as long as their output can be decoded by the decoder defined in this standard. A non-normative reference encoder is provided in Annex C for illustration purposes.

4.1 Bitstream syntax

There are two types of haptic bitstreams, including the forward bitstream (i.e., the position/velocity information transmitted from the leader to the follower) and the feedback bitstream (i.e., the force/torque feedback transmitted from the follower to the leader), indicated by the "TYPE" bit in the message header.

Header: The first bit indicates the type of the current bitstream (T), e.g., 0 for the forward bitstream, 1 for the feedback bitstream. The following 12 bits are the field indicator (F), which specifies if the signals in the corresponding dimension (x, y, or z) are transmitted. The next 3 bits are reserved for future extension (X). The last 32 bits indicate the sequence number of the current message. Table 1 and Table 2 show the bitstream structure of the KCnD header and the field indicator for the forward and feedback channel.

Table 1—Bitstream header structure for KCnD

1 bit	12 bit	3 bit	32 bit
T	F	X	S

T	Type	0 for the forward bitstream, 1 for the feedback bitstream
F	Field indicator	See the following syntax table
X		Extension bits for future uses
S	Sequence number	In the encoder, the sequence number is increased by 1 for each sampling step. The packet transmitted contains the most recent sequence number, telling the decoder how many samples have been dropped in between versus the current timestamp. The link between the timestamp and the sequence number is handled by the handshaking mechanism.

Table 2—Field indicator syntax for KCnD

Field indicator: 12 bits for forward bitstream	3 for position, 3 for orientation, 3 for velocity, 3 for angular velocity
Field indicator: 6 bits for feedback bitstream 6 0s to fill up the “F” sector	3 for force, 3 for torque, e.g., “111000000000” indicates that the feedback bitstream contains 3-D force signals, but no torque signals.

Payload: the payload contains a series of 32-bit float numbers indicating the corresponding position/orientation, velocity/angular velocity, and force/torque values. For the forward channel, the leader sends its position, orientation, velocity, and angular velocity signals to the follower. For feedback channel, the follower sends the force and torque signals back to the leader. See Figure 4.

forward bitstream

32 bit												
pos-x	pos-y	pos-z	ori-x	ori-y	ori-z	vel-x	vel-y	vel-z	ang-x	ang-y	ang-z	

feedback bitstream

32 bit					
force-x	force-y	force-z	torque-x	torque-y	torque-z

Figure 4—Example of payload of forward and feedback bitstreams

4.2 Kinesthetic decoder

Figure 5 presents the KCnD codec scheme in a two-port teleoperation system. In the forward channel the leader transmits its velocity signals to the follower, while in the feedback channel the follower sends force signals back to the leader. The KCnD standardizes the bistream at each decoder side.

Figure 6 illustrates an exemplar decoding process for one dimension. Note that the encoder is not standardized for KCnD. However, an exemplar implementation of encoder is also provided in Annex C as an informative reference.

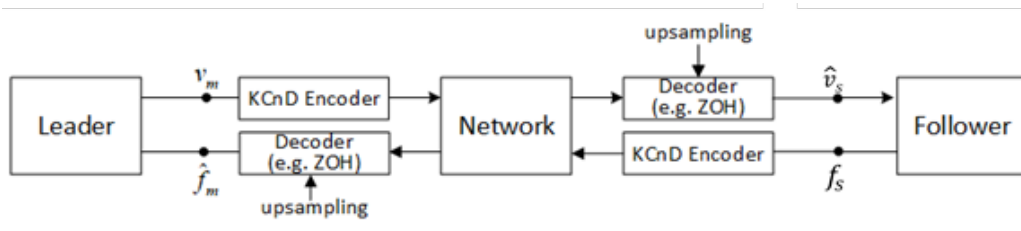


Figure 5—KCnD codec scheme in two-port bilateral haptic teleoperation system

The variables shown in Figure 5 are as follows:

- v_m is the leader velocity
- $\hat{v}_s$ is the reconstructed velocity at the follower side after the DB decoder
- f_s is the follower force to be transmitted to the leader
- $\hat{f}_m$ is the reconstructed force at the leader side after the DB decoder

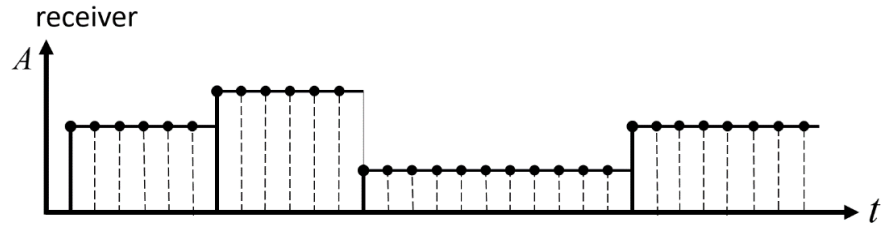


Figure 6—Illustration of the decoding process (e.g., upsampling with ZOH strategy) for one dimension

The same decoding process is repeated for all dimensions individually. The samples with solid lines correspond to receive packets. During the decoding process for one dimension, the decoder reconstructs the signal back to its original sampling rate (typically in the range from 1000 HZ to 4000Hz). The actual reconstruction process is not part of the nominative standard. In Figure 6, zero-order hold (ZOH) strategy is used for illustration purposes. That is, the reconstructed signal is held constant at its previously received sample value until the next significant sample arrives.

Note that only the format of decoder bitstreams are standardized for the KCnD. The upsampling schemes e.g., ZOH, are not standardized. Therefore, the proposed ZOH approach is considered as one example reconstruction mode in this standard. More sophisticated mechanisms may also be used but will not be defined in this standard.

As an example, the feedback bitstream of the force signals as shown in Figure 6 is illustrated by following the standardized bitstream rules in Table 1 and Figure 4.

Assume that X in Figure 7 corresponds to the force signal in the y dimension in the feedback direction, and the sequence number starts with 1. All other dimensions are 0. The resulting 4 packets received by the decoder are as follows.

header				payload
T	F	X	S-32bits	Payload-32bits
1	0	1	0 0 0 0 0 0 0 0 0 0...0 0 0 0 1	X(1) in binary
1	0	1	0 0 0 0 0 0 0 0 0 0...0 0 1 1 1	X(7) in binary
1	0	1	0 0 0 0 0 0 0 0 0 0...0 1 1 0 1	X(13) in binary
1	0	1	0 0 0 0 0 0 0 0 0 0...1 1 0 0 0	X(24) in binary

Figure 7—Example bitstream/packet stream for teleoperation

This KCnD solution has been implemented in the reference setup (see Annex B) for KCs provided by the Haptic Codec Task Group. The implemented core functions of the codec can be downloaded from the link given in B.2.1. Experimental and cross-validation results are shown in Annex B.

5. Part II: Delay-robust kinesthetic codec (KCwD)

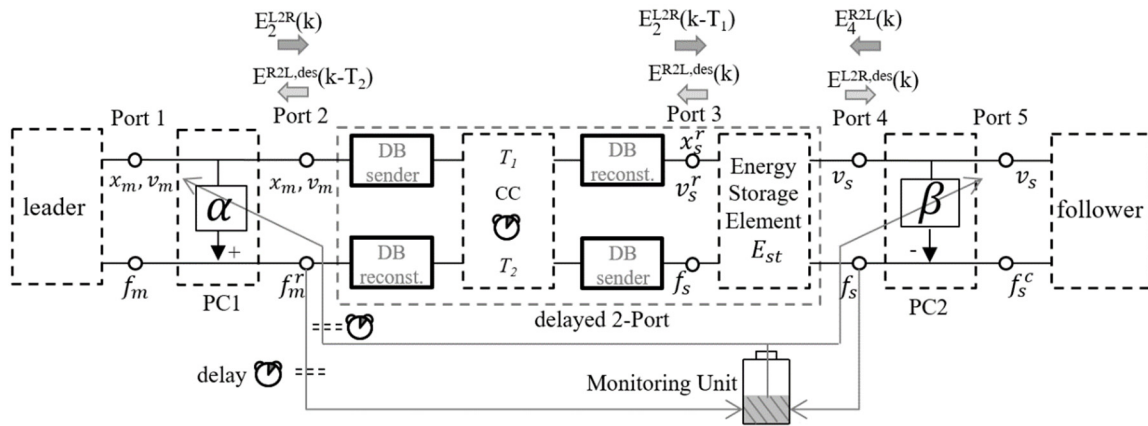
This clause presents a KC for the TI, which mitigates communication impairments, such as communication delay, delay jitter, and packet loss between the leader and the follower system. Although these communication impairments are from different network challenges, all of them can lead to unstable interaction such as oscillation on the leader/follower or unpredictable vibrations in the force signals.

The defined KC is able to reduce the packet rate of kinesthetic information while maintaining a high quality of experience and system stability. For the KCwD, what is standardized are the combination of the haptic packet rate reduction encoder/decoder and the passivity controller. The KCwD combines the state-of-the-art

time-domain passivity approach with energy reflection (TDPA-ER) (Panzirsch [B20]) and the perceptual deadband coding (DB) scheme (Hunterseer [B9]). All necessary details for implementation are explained in this clause. Readers can also refer to Xu [B27] for more details about the experimental setup and evaluation results.

An overview of the presented KCwD for a time-delayed teleoperation is illustrated in Figure 8. The follower's motion controller is considered as a storage element. It provides the computed control force and stores energies received from the leader and follower (see 5.2.3 for more details). The computed force f_s will be sent to the follower to control its motion and sent to the leader side as the feedback force. The leader (the haptic interaction device, HSI) is referred to as the main device (subscript m), while the follower (the remote robotic system, TOP) is referred to as the secondary device (subscript s) in the following. The adaptive dampings α and β are computed according to the energy flow on the corresponding side. They are not coupled. The energy flow shows the transmission of energy information at different ports in the teleoperation system. See 5.1.2 and 5.1.6.2 for more information.

The standardized modules are the DB senders and receivers for the DB scheme, and the PC1, PC2, and monitoring unit for the passivity controller. Generally, the TDPA helps ensure the passivity of a two-port network including a communication channel (CC) that can introduce communication impairments. It dissipates excessive output energy with the help of damping at the leader and follower side. The state-of-the-art TDPA-ER (Panzirsch [B20]) is adopted as the control scheme, which also ensures that there is no position drift between the leader and follower devices.



Copyright © 2020 IEEE. All rights reserved. Used with permission from Xu et al. [B27].

Figure 8—System overview: Combination of TDPA-ER and DB approach
[Note that the sampling (k) of the position, velocity, and force signals are hidden for a better layout]

The variables in Figure 8 are as follows:

- v_m is the leader velocity
- v_s^r is the follower received/desired velocity after the DB decoder
- v_s is the follower velocity
- f_s is the computed follower control force before the passivity controller (PC2). It is transmitted back to the leader.
- f_s^c is the follower control force after the passivity controller. Force applied to the follower device.
- f_m^r is the leader received/desired force before the passivity controller (PC1). It is the reconstructed follower control force f_s with delay T_2 .
- f_m is the leader force after the passivity controller. Force applied to the leader device.
- T_1 and T_2 are the forward and backward communication delays
- α and β are the adaptive dampings to dissipate excessive energy (see 5.1.4.2 and 5.1.6.3)
- E_i^{L2R} : energy flow direction from left to right at port i
- E_i^{R2L} : energy flow direction from right to left at port i

Note that according to the signal flows defined in Figure 8, L2R means positive energy value at port i (energy flows from the leader to the follower) and R2L means negative energy value [energy flows from the follower to the leader. See Equation (3) for more details].

5.1 KCwD decoder

An overview of the KCwD (combination of the TDPA-ER and the DB approach) is illustrated in Figure 8. The KCwD standardizes both the encoder and the decoder side, because computation of the used energy values cannot be defined independently for the encoder and the decoder.

The DB modules follow the concept introduced in Part I for KCnD. In order to guarantee system passivity, the energy flow information needs to be exchanged along with the position, velocity, and for signals. See Figure 8. For the TDPA-ER approach, it considers the follower controller as an energy storage element (ESE). The input energy from both leader and follower sides is stored in the ESE. At each sampling step, the ESE assigns the stored energy to the leader and follower as the upper bound of the output energy. Considering the delayed two-port network in which Port 1 is the left port and Port 5 is the right port, the passivity of the delayed two-port network (containing CC, ESE, PC1, PC2, DB sender, and DB reconstruction) is ensured if the input energy at Port 1 and Port 5 in Figure 8 is larger than or equal to the output energy at the corresponding ports [see Equation (1)]:

$$E_1^{L2R}(k) + E_5^{R2L}(k) \geq E_1^{R2L}(k) + E_5^{L2R}(k) \quad (1)$$

The presented codec in the following is designed for position-force teleoperation architectures (position/velocity signals in the forward channel and computed force signals in the backward channel). Other architectures, e.g., position-position teleoperation is not supported by the presented KCwD.

5.1.1 Bitstream syntax

Similar to the KCnD case, the exchange of haptic data also includes two directions of information exchange: the forward and the backward bitstreams. Additionally, exchange of energy information is required in the payload data. Hence, the header of the KCnD is extended accordingly.

For the header part, the first bit indicates the type of the current bitstream: 0 for the forward bitstream, 1 for the feedback bitstream. The following 18 bits are the field indicator for the availability of the corresponding haptic/energy information in each dimension in the payload. Five bits are left for future extension, the 32-bit number at the end represents the sequence number of the sent packet. Table 3 and Table 4 show the bitstream structure of the KCwD header and the field indicator for the forward and feedback channel.

Table 3—Bitstream header structure for KCwD

1 bit	18 bit	5 bit	32 bit
T	F	X	S

T	Type	0 for the forward bitstream, 1 for the feedback bitstream
F	Field indicator	See the syntax table below
X		Extension bits for future use
S	Sequence number	In the encoder, the sequence number is increased by 1 for each sampling step. The packet transmitted contains the most recent sequence number, telling the decoder how many samples have been dropped in between versus the current timestamp. The link between the timestamp and the sequence number is handled by the handshaking mechanism.

Table 4—Field indicator syntax for KCwD

Field indicator: 18 bits for forward bitstream	3 for position, 3 for orientation, 3 for velocity, 3 for angular velocity 3 for translational energy, 3 for rotational energy
Field indicator: 12 bits for backward bitstream 6 0s to fill up the “F” sector	3 for force, 3 for torque, 3 for translational energy, 3 for rotational energy

In the following, an example of the bitstream for KCwD is described. Figure 9(a) and (b) illustrate the bitstream structure. For the same example bitstream/packet stream shown in Figure 7, the header and payload information are illustrated in Figure 9(c).

E_x , E_y , and E_z are the translational kinetic energy in the x, y, and z direction. E_{ox} , E_{oy} , and E_{oz} are the rotational kinetic energy in the x, y, and z directions. See 5.1.2 for the details of calculation.

32 bit																	
pos-x	pos-y	pos-z	ori-x	ori-y	ori-z	vel-x	vel-y	vel-z	ang-x	ang-y	ang-z	Ex	Ey	Ez	Eox	Eoy	Eoz

(a) Forward bistream

32 bit																	
Fx	Fy	Fz	Tx	Ty	Tz	Ex	Ey	Ez	Eox	Eoy	Eoz						

(b) Feedback bistream

header				payload	
T	F	X	S (32 bit)	32 bits	32 bits
1	0 1 0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 ... 0 0 0 0 1	X(1): force in Y	X(1): energy in Y
1	0 1 0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 ... 0 0 1 1 1	X(7): force in Y	X(7): energy in Y
1	0 1 0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 ... 0 1 1 0 1	X(13): force in Y	X(13): energy in Y
1	0 1 0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 ... 1 1 0 0 0	X(24): force in Y	X(24): energy in Y

(c) Example bitstream/packet stream for the KCwD

Figure 9—Structure and example of payload for forward and feedback bitstreams for the KCwD

5.1.2 Calculation of energy values

This subclause explains how to calculate the power and energy variables for haptic codec and system control. Note that all variables/values in the following calculation, unless otherwise stated, are in the form of float numbers with 32-bit precision.

Energy calculation at each port is defined as follows:

The power P_i at the port (i) is calculated with the velocity v_i and the force f_i for translational kinetic energy or with the angular velocity ω_i and the torque τ_i for rotational kinetic energy [see Equation (2)]:

$$\text{Translational: } P_i(k) = v_i(k)f_i(k) \quad \text{rotational: } P_i(k) = \omega_i(k)\tau_i(k) \quad (2)$$

According to the sign, the flow direction of the power at each port is determined by two parts [see Equation (3)]:

$$P_i^{L2R}(k) = \begin{cases} 0 & \text{if } P_i(k) < 0 \\ P_i(k) & \text{else} \end{cases}$$

$$P_i^{R2L}(k) = \begin{cases} 0 & \text{if } P_i(k) > 0 \\ -P_i(k) & \text{else} \end{cases} \quad (3)$$

The power P_i^{L2R} flows in left to right direction from leader to follower and the power P_i^{R2L} in the opposite direction. The respected energy is calculated via integration over time with the sampling time T_s [see Equation (4) and Equation (5)]:

$$E_i^{L2R}(k) = T_s \sum_{j=0}^k P_i^{L2R}(j) \quad (4)$$

$$E_i^{R2L}(k) = T_s \sum_{j=0}^k P_i^{R2L}(j) \quad (5)$$

5.1.3 Leader encoder—PC1 energy calculation

The DB sender is considered as the leader encoder. It decides if the current velocity and energy data needs to be transmitted to the follower side. See 5.2.3 for more information.

At every sampling instant k , the leader's velocity v_m and the calculated input energy at Port 2 [input energy from the leader, see Equation (4) where $i=2$] are transferred to the DB module at the leader (the DB sender in Figure 8) for encoding and transmission over the network.

5.1.4 Leader decoder

The leader decoder includes the DB reconstruction module and the PC1 controller. It receives and reconstructs the signals from the follower side, and controls the force that is applied to the leader device for guaranteeing passivity.

5.1.4.1 DB reconstruction

The leader decoder includes the DB reconstruction module, and the PC1 controller. It receives and reconstructs the signals from the follower side, and controls the force that is applied to the leader device for guaranteeing passivity.

If no packet is received at the sampling time, the desired force and energy values are set to the former values [see Equation (6)]:

$$f_m^r(k) = f_m^r(k^*), \quad E^{R2L,des}(k - T_2) = E^{R2L,des}(k^*) \quad (6)$$

where $f_m^r(k^*)$ is the most recently received force at the leader side. $E^{R2L,des}(k^*)$ is the most recently received energy at the leader side at the time instant k^* while the current time instant is k .

5.1.4.2 PC1 controller

At every sample time, the reconstructed force $f_m^r(k)$ is revised according to $E^{R2L,des}(k - T_2)$ to help ensure system stability. PC1 has to assure that not more energy than $E^{R2L,des}(k - T_2)$ leaves at Port 2 in the R2L direction (sent to the leader device). Excessive energy W^{PC1} must be dissipated [see Equation (7)]:

$$W^{PC1}(k) = E^{R2L,des}(k - T_2) - E_2^{R2L}(k) - W_{diss}^{PC1}(k - 1) \quad (7)$$

If $|v_m(k)| \neq 0$ and $W^{PC1}(k) < 0$, then $\alpha(k) = \frac{-W^{PC1}(k)}{T_s |v_m(k)|^2}$, and $f_m(k) = f_m^r(k) + \alpha(k)v_m(k)$

else $\alpha(k) = 0$, $f_m(k) = f_m^r(k)$, and $W^{PC1}(k) = 0$

where

$\alpha(k)$ is the adaptive damping at the leader side for passivity control

$E_2^{R2L}(k) = \sum_{j=0}^k P_2^{R2L}(j)T_s$ is the unchanged energy output at the port 2

$W_{diss}^{PC1}(k) = \sum_{j=0}^k W^{PC1}(j)$ is the sum of the earlier dissipated energy at PC1. Energy $W^{PC1}(k)$ is dissipated by PC1 according to Figure 8 if $W^{PC1}(k)$ is negative.

5.1.5 Follower encoder—PC2 controller

At every sampling step, follower's computed control force f_s and the energy value to be transmitted to the leader $E^{R2L,des}$ are calculated and sent to the follower's DB module (DB sender) for transmission. The way to compute $E^{R2L,des}$ is described in the 5.1.6. The way to compute the follower's control force f_s is exemplified in 5.2.3.

5.1.6 Follower decoder

The follower decoder includes the DB reconstruction module, the monitoring unit, and the PC2 controller. It receives and reconstructs the signals from the leader side, monitors system passivity at the follower side, and controls the force that is applied to the robot and transmitted back to the leader.

5.1.6.1 DB reconstruction

DB reconstruction refers to the module “DB reconst” in Figure 8. It decodes/reconstructs the received signals for the follow-up computation. If no packet is received at each sampling time, the desired velocity and energy values are set to the former values [see Equation (8), Equation (9), and Equation (10)]:

$$v_s^r(k) = v_s^r(k^*) \quad (8)$$

$$x_s^r(k) = x_s^r(k-1) + v_s^r(k)T_s \quad (9)$$

$$E_2^{L2R}(k - T_1) = E_2^{L2R}(k^*) \quad (10)$$

where

- $x_s^r(k)$ is the calculated desired position at the follower side according to the reconstructed velocity signal
- $v_s^r(k)$ is the most recently received velocity at the follower side
- $E_2^{L2R}(k^*)$ is the most recently received energy that was sent by the leader

Note that position drift between the leader and follower should be corrected once the leader's position $x_m(k)$ is received at the follower side. This is to directly apply the received position information $x_s^r(k) = x_m(k - T_1)$ for the subsequent computation. Otherwise, simply $x_s^r(k) = x_s^r(k-1) + v_s^r(k)T_s$ is applied.

5.1.6.2 Monitoring unit

The monitoring unit is adopted for computing the energy flow and deciding the amount of energy to be transferred to the leader and follower devices. As explained in Figure 8, the follower's motion controller is considered as the storage element. Calculations of the energy storage E_{st} and the assigned energy to the leader ($E^{R2L,des}$) and follower ($E^{L2R,des}$) are done in the monitoring unit in Figure 8.

At each sampling step, the energy increments since the last sample time at the Port 2 and Port 4 are considered as the inputs to the delayed two-port network and are transmitted in the ESE [see Equation (11)].

$$E_{st}(k) = E_{st}(k-1) + E_2^{L2R}(k - T_1) - E_2^{L2R}(k - T_1 - 1) + P_4^{R2L}(k)T_s \quad (11)$$

Next, the upper bound of the output energy from the ESE is calculated and transmitted to the leader and the follower. If the output power at Port 3 and Port 4 ($P_{out}(k) = P_3^{R2L}(k) + P_4^{L2R}(k)$) is less than the stored energy in the ESE, the same amount of energy is assigned accordingly. Otherwise, all energy in the ESE is assigned to the leader and follower according to the ratio of $P_3^{R2L}(k)$ and $P_4^{L2R}(k)$ in $P_{out}(k)$ [see Equation (12), Equation (13), and Equation (14)]:

$$p^{R2L,des}(k) = \begin{cases} E_{st}(k)P_3^{R2L}(k)/(T_s P_{out}(k)) & \text{if } E_{st}(k) < P_{out}(k)T_s \\ P_3^{R2L}(k) & \text{else} \end{cases}$$

$$p^{L2R,des}(k) = \begin{cases} E_{st}(k)P_4^{L2R}(k)/(T_s P_{out}(k)) & \text{if } E_{st}(k) < P_{out}(k)T_s \\ P_4^{L2R}(k) & \text{else} \end{cases} \quad (12)$$

$$E^{R2L,des}(k) = E^{R2L,des}(k-1) + p^{R2L,des}(k)T_s \quad (13)$$

$$E^{L2R,des}(k) = E^{L2R,des}(k-1) + p^{L2R,des}(k)T_s \quad (14)$$

where $E^{R2L,des}(k)$ is sent to the follower's encoder for transmission. $E^{L2R,des}(k)$ is applied for the follower's passivity control PC2.

After this, the ESE needs to update its energy storage by subtracting the assigned energy from $E_{st}(k)$ [see Equation (15)]:

$$E_{st}(k) = E_{st}(k) - (p^{R2L,des}(k) + p^{L2R,des}(k))T_s \quad (15)$$

5.1.6.3 PC2 controller

Similarly to PC1 [as shown in Equation (16)],

$$W^{PC2}(k) = E^{L2R,des}(k) - E_4^{L2R}(k) - W_{diss}^{PC2}(k-1) \quad (16)$$

$$\text{If } |v_s(k)| \neq 0 \text{ and } W^{PC2}(k) < 0, \beta(k) = \frac{-W^{PC2}(k)}{T_s |v_s(k)|^2}, f_s^c(k) = f_s(k) - \beta(k)v_s(k)$$

$$\text{else } \beta(k) = 0, f_s^c(k) = f_s(k), W^{PC2}(k) = 0$$

Please note that to help prevent singularities, in the divisions of the upper equations, lower limits of the absolute velocity and power shall be set to $v_s > 0$. These limits depend on the operating system.

5.2 KCwD encoder

The following design criteria are suggested for the KCwD.

5.2.1 Leader DB sender

The leader velocity $v_m(k)$, position $x_m(k)$, and energy $E_2^{L2R}(k)$ are transmitted, if, as shown in Equation (17):

$$|v_m(k) - v_m(k^*)| > p|v_m(k^*)| \quad (17)$$

where

p is the perceptual deadband parameter (DBP)
 $v_m(k^*)$ is the most recently transmitted leader velocity

The transmission of position is to correct the position drift introduced by the DB approach.

5.2.2 Follower DB Sender

The follower control force $f_s(k)$ and $E_2^{R2L,des}(k)$ are transmitted, if, as shown in Equation (18):

$$|f_s(k) - f_s(k^*)| > p|f_s(k^*)| \quad (18)$$

where

$f_s(k^*)$ is the most recently transmitted follower force

5.2.3 Energy storage element

The storage element is a motion controller to produce the follower's computed force. A controller with a position gain k_p is adopted in this codec. The follower's computed control force is shown in Equation (19):

$$f_s = k_p \cdot (x_s^r - x_s) \quad (19)$$

6. Part III: Tactile codec

This clause specifies coded representation of vibrotactile information and the decoding process for this coded representation. A short-form designation of the tactile coding algorithm specified in the body of the present document is *perceptual vibrotactile codec based on sparse linear prediction* (PVC-SLP). An overview of PVC-SLP as a reference tactile codec (TC) is provided in the following subclauses. The decoding flow is diagramed in Figure 10. More detailed information on the processing blocks will be found in the following subclauses. An example encoder architecture is provided in Annex E, which might help in understanding the decoding process described in this clause. The coded representations specified herein are suitable for use in vibrotactile information transmission and storage applications.

6.1 PVC-SLP decoder

At the receiver side, the decoder (Figure 10) unpacks the received bitstream and extracts the DCT bitstream, RC bitstream, the used quantization scale, and the dynamic range of the original input signal. RC and DCT Huffman dictionaries are loaded. RC indexes are then decoded, dequantized, and finally transformed back to SLP filter coefficients for each frame, which will be explained in the following subclause in more detail. Subsequently, the zero-run-length (ZRL) decoding is applied to the residual DCT coefficients, and they are dequantized with the perceptual dequantizer, then inverse DCT transformed to restore a signal residual component. Eventually, the vibrotactile signal is synthesized by filtering the restored residual using the SLP coefficients in the SLP synthesis filter, and the signal is restored back to its dynamic range.

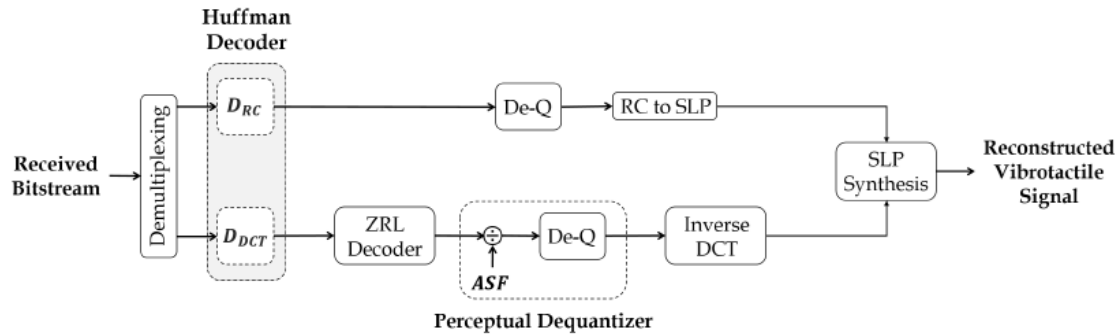


Figure 10—Decoder structure of reference TC or PVC-SLP

6.1.1 Coded frames specifications

A coded vibrotactile bitstream is made up of a sequence of coded frames. Each frame contains a header and a payload. The header at the beginning of each frame contains information needed to decode the payload.

The number of frames for a given vibrotactile signal varies based on the sampling rates since the tactile codec applies a fixed-frame length format. The frame is of 200 samples with commonly used 16 bits per sample. The number of frames per second N can be computed from the sampling frequency f_s as in Equation (20):

$$N = \frac{f_s}{200} \quad (20)$$

The haptic handshake metadata shall include an entry of the required sampling rate used to acquire the tactile information. Table 5 summarizes the sampling rates compatible with this tactile codec and the corresponding frame length (in milliseconds) as well as the number of frames as an illustrative example. Higher network layers shall be responsible for frame synchronization to appropriately decode the received vibrotactile signals.

Table 5—PVC-SLP compatible sampling rates and corresponding frame length

Fs (Samples/second)	Frame length (milliseconds)	Number of frames/second
2800	71.4	14
4000	50	20
6000	33.3	30
8000	25	40
10 000	20	50
44 000	4.5	220
48 000	4.16	240

6.1.2 Bitstream syntax

The order of arrival of information within this bitstream is as follows: header, encoded DCT coefficients of the residual, and encoded RCs. The header occupies 32 bits, which breaks down to 6 bits for the quantization level, and 26 bits for the encoded vibrotactile frame dynamic range. The frame min-max values are stored using 13 bits each such that the resulting precision is comparable to IEEE Std 754™ for single precision floating-point numbers. Table 6 summarizes the syntax elements of the bitstream. An example of encoded vibrotactile signals is shown in Table 7. See Figure 11.



Figure 11—Detailed description of bitstream elements

Table 6—Breakdown of bitstreams' elements

Header	32bits	DL	6bits	DCT coefficients quantization value
		MM	26bits	13-bits floating-point representation of the maximum and absolute minimum of a vibrotactile signal (3bits for the integer part and 10bits for the decimal)
Payload	Variable length	RC	Encoded RC coefficients bitstream	
		Residual DCT	Encoded DCT coefficients of residual bitstream with EOB indicator at the end of the stream	

Table 7—Example bitstream syntax elements of sample vibrotactile signal

FS Syntax	Binanzation of syntax elements	Corresponding decimal values
DL	011000	0.01
MM	000000100011, 000000101101	0.03437, -0.04476
RCs	001011xxxxxx.....xxxxxxxxxx11111	
Residual DCTs	00001xxxxxx.....xxxxxxxxxx001100	

6.1.3 Huffman decoder

After unpacking the received bitstream, the next step at the decoder is to load the Huffman dictionaries for RCs and residual DCT coefficients. There is only one Huffman dictionary for RCs, D_{RC} , which is used for Huffman-based entropy decoding of the RC bitstream. The value to codeword mapping of D_{RC} is given in 6.2.2. The Huffman dictionary for DCT coefficients, D_{DCT} , is loaded corresponding to the given quantization scale δ , which is extracted from the bitstream header DL in Table 6. The mapping between the value of quantization scale δ and the DL bitstream is given in 6.2.1. Similarly, the DCT bitstream is decoded using the dictionary D_{DCT} . The D_{DCT} dictionary tables for each quantization scale δ are given in 6.2.3.

6.1.4 RCs dequantization

After decoding the RC bitstream to RCs, dequantization with the constant quantization scale of $\delta_{RC} = 0.03$ is performed using Equation (21):

$$\hat{\gamma} = DeQ(\gamma_Q) = \gamma_Q * \delta_{RC} \quad (21)$$

where

γ_Q is a vector of length 20 containing the quantized RCs

$\hat{\gamma}$ is the dequantized vector of γ_Q

The dequantized RCs are transformed back to SLP coefficients.

6.1.5 RC to SLP transformation

The input is the dequantized RCs vector $\hat{\gamma} = [\hat{\gamma}_1, \hat{\gamma}_2, \dots, \hat{\gamma}_N]$, with $N = 20$. The zeroth order SLP vector term is 1. Starting with this term, recursion is used to calculate the higher order SLP vectors until the entire matrix is calculated. This SLP matrix [see Equation (23)] consists of SLP vectors of order 0 through N found by using the Levinson recursion. Equation (22) shows the recursion steps to obtain SLP matrix:

for $p = 0, 1, \dots, N - 1$

$$\tilde{a}_{p+1,m} = \tilde{a}_{p,m} + \hat{\gamma}_{p+1} \tilde{a}_{p,p+1-m}, \quad 1 \leq m \leq p$$

$$\tilde{a}_{p+1,p+1} = \hat{\gamma}_{p+1} \quad (22)$$

$$SLP_{matrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & \tilde{a}_{11} & 0 & 0 & \dots & 0 \\ 1 & \tilde{a}_{21} & \tilde{a}_{22} & 0 & \dots & 0 \\ 1 & \tilde{a}_{31} & \tilde{a}_{32} & \tilde{a}_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & \tilde{a}_{N1} & \tilde{a}_{N2} & \tilde{a}_{N3} & \dots & \tilde{a}_{NN} \end{bmatrix} \quad (23)$$

In the SLP matrix [see Equation (22)], the last row represents the decoded SLP coefficients.

6.1.6 Zero-run-length (ZRL) decoding of residual DCT bitstream

After Huffman decoding of the DCT bitstream, ZRL decoding is applied to obtain the quantized DCT coefficients of the residual. In the ZRL encoded residual DCT block, each value that comes after the value of 0 indicates the number of repetitions of 0. In order to apply ZRL decoding, each non-zero value is written to the decoded values as they are, and each zero is first written and then repeated by the number that comes after the zero in ZRL encoded residual DCT block. End of a DCT block is indicated by the value EOB. An example of a ZRL encoded DCT coefficient block and the output of the corresponding ZRL decoding is depicted in Figure 12.



Figure 12—Before ZRL decoding (left), after ZRL decoding (right)

6.1.7 Perceptual dequantization of DCT coefficients of the residual

The dequantization of the one frame 200 sample DCT coefficients of the residual is performed using the acceleration sensitivity function (ASF), and the quantization scale δ (DL), as depicted in Figure 10. The corresponding dequantization formula is given in Equation (24):

$$\hat{e}(k) = DeQ(e(k)) = e(k) * \delta / A_{ASF}(k) \quad (24)$$

where

$e(k)$ is the k^{th} quantized DCT coefficient

$\hat{e}(k)$ is the k^{th} dequantized DCT coefficient in one signal frame of 200 samples

$A_{ASF}(k)$ is the value of the ASF at the k^{th} DCT coefficient frequency: $A_{ASF}(k) = A_{ASF}(w)|_{w=2\pi k/M}$

for $M = 200$.

6.1.8 Type-II inverse DCT of DCT coefficients

Given the perceptually dequantized DCT coefficients $\hat{e}(k)$, the type-II inverse DCT is applied and the reconstructed residual signal $\tilde{r}(n)$ is obtained with Equation (25):

$$\tilde{r}(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} \hat{e}(k) \frac{1}{\sqrt{1+\delta_{k0}}} \cos\left(\frac{\pi k}{2M}(2n-1)\right) \quad (25)$$

for $M = 200$ and $n = 0, 1, \dots, M-1$

where δ_{k0} is the Kronecker delta: $\delta_{k0} = 1$ if $k = 0$, $\delta_{k0} = 0$ if $k \neq 0$.

6.1.9 Perceptual Tactile Sensitivity Model—Acceleration sensitivity function

The ASF (Hassen, Steinback [B7], Hassen et al. [B8]) given in Table 8 is inspired by the four-channel mechanoreceptors mediating the sensory perception of tactile sensations.

Table 8—ASF values for one vibrotactile frame

1	1 26	0.5678 51	0.2924 75	0.0998101	0.0282126	0.0091151	0.0044176	0.0044
2	0.9912 27	0.5584 52	0.2816 76	0.0951102	0.0269127	0.0087152	0.0044177	0.0044
3	0.9661 28	0.5487 53	0.2710 77	0.0905103	0.0257128	0.0083153	0.0044178	0.0044
4	0.9280 29	0.5389 54	0.2605 78	0.0862104	0.0245129	0.0080154	0.0044179	0.0044
5	0.8815 30	0.5288 55	0.2503 79	0.0821105	0.0234130	0.0077155	0.0044180	0.0044
6	0.8319 31	0.5185 56	0.2403 80	0.0782106	0.0223131	0.0073156	0.0044181	0.0044
7	0.7841 32	0.5080 57	0.2305 81	0.0744107	0.0213132	0.0070157	0.0044182	0.0044
8	0.7416 33	0.4973 58	0.2210 82	0.0709108	0.0203133	0.0068158	0.0044183	0.0044
9	0.7074 34	0.4863 59	0.2117 83	0.0675109	0.0194134	0.0065159	0.0044184	0.0044
10	0.6818 35	0.4752 60	0.2026 84	0.0643110	0.0185135	0.0062160	0.0044185	0.0044
11	0.6638 36	0.4640 61	0.1938 85	0.0612111	0.0177136	0.0060161	0.0044186	0.0044
12	0.6521 37	0.4526 62	0.1853 86	0.0583112	0.0169137	0.0057162	0.0044187	0.0044
13	0.6450 38	0.4411 63	0.1771 87	0.0555113	0.0161138	0.0055163	0.0044188	0.0044
14	0.6408 39	0.4295 64	0.1691 88	0.0529114	0.0154139	0.0053164	0.0044189	0.0044
15	0.6380 40	0.4179 65	0.1615 89	0.0504115	0.0147140	0.0051165	0.0044190	0.0044
16	0.6352 41	0.4063 66	0.1541 90	0.0480116	0.0141141	0.0049166	0.0044191	0.0044
17	0.6316 42	0.3946 67	0.1469 91	0.0457117	0.0135142	0.0047167	0.0044192	0.0044
18	0.6273 43	0.3830 68	0.1401 92	0.0435118	0.0129143	0.0045168	0.0044193	0.0044
19	0.6222 44	0.3714 69	0.1336 93	0.0415119	0.0123144	0.0044169	0.0044194	0.0044
20	0.6164 45	0.3599 70	0.1273 94	0.0395120	0.0118145	0.0044170	0.0044195	0.0044
21	0.6097 46	0.3484 71	0.1213 95	0.0376121	0.0113146	0.0044171	0.0044196	0.0044
22	0.6024 47	0.3370 72	0.1155 96	0.0359122	0.0108147	0.0044172	0.0044197	0.0044
23	0.5945 48	0.3256 73	0.1100 97	0.0342123	0.0103148	0.0044173	0.0044198	0.0044
24	0.5860 49	0.3144 74	0.1048 98	0.0326124	0.0099149	0.0044174	0.0044199	0.0044
25	0.5771 50	0.3033 75	0.0998 99	0.0311124	0.0095150	0.0044175	0.0044200	0.0044
			100	0.0296125				

6.1.10 SLP synthesis and vibrotactile signal reconstruction

After transforming the quantized coefficients back to SLP representation and perceptually dequantizing the residual and transforming it back to the time domain, the vibrotactile signal at the decoder side can ultimately be reconstructed using Equation (26):

$$\tilde{x}(n) = \tilde{r}(n) + \sum_{i=1}^N \tilde{a}_i \tilde{x}(n-i), \quad (26)$$

where

$\tilde{x}(n)$ is the reconstructed vibrotactile signal
 $\tilde{r}(n)$ is the reconstructed residual signal
 $\tilde{a}_i$ is the reconstructed i^{th} SLP coefficient

6.1.11 Post-processing: Signal min-max restoration

The last step to fully reconstruct the signal after SLP synthesis is to restore the signal original dynamic range by using the min-max values extracted from the bitstream header. Signal min-max restoration is applied using Equation (27) and Equation (28):

$$\tilde{x}(\tilde{x} > 0) = \tilde{x}(\tilde{x} > 0) * Max \quad (27)$$

$$\tilde{x}(\tilde{x} < 0) = \tilde{x}(\tilde{x} < 0) * Min \quad (28)$$

where

Max refers to the signal max obtained from the bitstream header

Min refers to the signal min obtained from the bitstream header

6.2 Tactile codec dictionaries

In this subclause, PVC-SLP dictionaries are listed. In Table 9 quantization scale (DL) dictionary and in Table 10 RCs dictionary are given. DCT coefficient dictionaries for different quantization scales are provided in Annex A.

6.2.1 Quantization scale (DL) dictionary

Table 9—Quantization scale (DL) dictionary

Value	Codeword	Value	Codeword
0.001	"10000"	0.1	"00110"
0.002	"01000"	0.15	"10110"
0.004	"11000"	0.2	"01110"
0.006	"00100"	0.3	"11110"
0.008	"10100"	0.4	"00001"
0.01	"01100"	0.5	"10001"
0.02	"11100"	0.6	"01001"
0.03	"00010"	0.7	"11001"
0.04	"10010"	0.8	"00101"
0.05	"01010"	0.9	"10101"
0.75	"11010"	1.0	"01101"

6.2.3 RC dictionary (D_{RC})

Table 10—RCs dictionary

Value	Codeword	Value	Codeword	Value	Codeword
−33	"0110000"	−10	"01100100"	13	"001000111"
−32	"0110011"	−9	"00100001"	14	"001000110"
−31	"10001001"	−8	"1001011"	15	"001011100"
−30	"10011111"	−7	"0010110"	16	"100111000"
−29	"011001010"	−6	"100100"	17	"0010000011"
−28	"100011110"	−5	"001001"	18	"1000101010"
−27	"100011100"	−4	"01101"	19	"0010111011"
−26	"100011011"	−3	"00010"	20	"1001110100"
−25	"100011010"	−2	"0011"	21	"00101110101"
−24	"100111011"	−1	"101"	22	"001011101001"
−23	"100011101"	0	"11"	23	"10001010110"
−22	"0010000000"	1	"010"	24	"10011101010"
−21	"100011111"	2	"0000"	25	"10011101011"
−20	"0010000010"	3	"0111"	26	"100010101110"
−19	"0010000001"	4	"00011"	27	"100010101111"
−18	"100111001"	5	"10000"	28	"0010111010000"
−17	"011001011"	6	"001010"	29	"001011101000100"
−16	"100010000"	7	"100110"	30	"001011101000101"
−15	"100010100"	8	"0110001"	31	"0010111010001100"
−14	"100010001"	9	"1001010"	32	"001011101000111"
−13	"001000101"	10	"00101111"	33	"0010111010001101"
−12	"10011110"	11	"10001011"		
−11	"10001100"	12	"001000100"		

7. Part IV: Plug-and-play (PnP) haptic interaction for Tactile Internet

This clause presents a PnP system for enabling seamless haptic interaction between heterogeneous haptic interfaces. The system comprises two key features: (1) a haptic metadata (HM) format to describe the characteristics of the haptic interface and requirements of the TI session, and (2) a haptic handshake protocol to automatically exchange HM between leader and follower for initiating TI communication.

7.1 Haptic handshake

Prior to the TI interaction, it is important that the TI session attributes, such as application requirements and functional capabilities of leader and follower, are exchanged between each other and a consensus on the same be reached. This is achieved via a “haptic handshake” protocol characterized by a three-way communication between the leader and the follower as shown in Figure 13. The leader/follower side initiating the TI session (leader) advertises all of its attributes to the other side (follower) through the “request” message. For example, the leader could be capable of supporting a multitude of haptic codecs and a certain maximum haptic sampling rate. In response, the follower chooses a feasible option out of the advertised specifications that it supports. For example, it may be compatible only with a subset of the advertised codecs or may support a lower sampling rate than the leader. These are communicated to the leader through “response” message. Upon reception of the response message, the leader transmits an “ACK” message acknowledging the arrival of the response message as well as indicating that the consensus on the attributes to be used is reached.

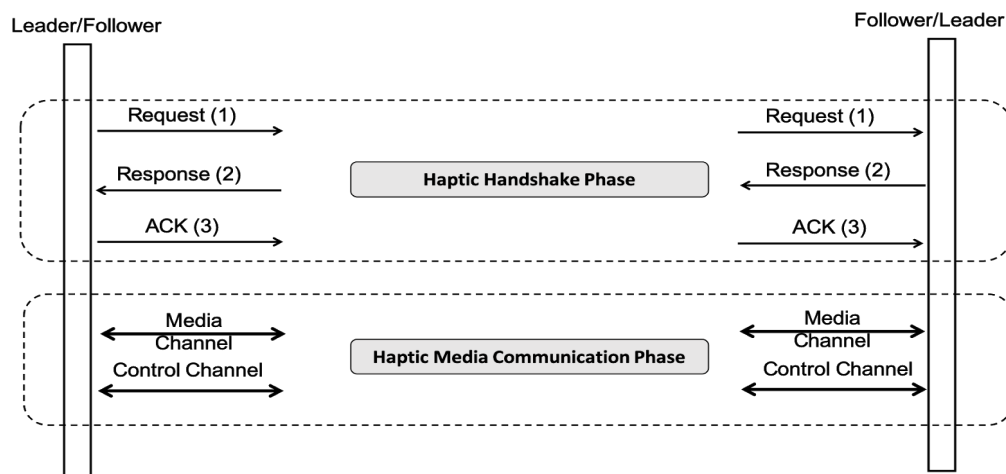


Figure 13—Schematic of the proposed Haptic handshake protocol
(It involves a simple 3-way handshake consisting of request, response, and acknowledgment.
The figure also shows the haptic media communication phase
when media data and control channels are established.

Following the haptic handshake phase, live haptic data as generated by the haptic codecs along with control messages are exchanged in real time between the leader and the follower in the “haptic media communication phase” (Figure 13). Owing to the scope of HCTG, the focus is restricted on haptic messages only in both handshake and media communication phases.

During the haptic media communication phase, if the session attributes change and need to be renegotiated then the control channel shall be utilized. The renegotiation messages exchanged could be similar to the 3-way handshake described in the haptic handshake phase. Additionally, the control channel could also support exchange of messages for monitoring network performance and adapting the application parameters in real time.

The two mentioned phases *haptic handshake phase* and *haptic media communication phase* correspond to the operational states *haptic negotiation* and *operation* of a general TI device as defined in Holland [B10].

7.2 Haptic metadata

A detailed description of the metadata exchanged during the haptic handshake phase haptic metadata (HM) is provided here. As shown in Figure 14, HM attributes are classified into two broad categories.

- a) **Media:** This describes the attributes related to source coding and communication of the haptic data. For instance, the type of kinesthetic and tactile codecs, their related parameters, sampling frequencies for haptic signals, and data format for each codec will be specified in this category.
- b) **Interface:** This category describes the various capabilities of the haptic interface (both leader and follower), such as input and output DoFs, resolution, ranges of displayable force and torque, workspace dimensions, vibration frequency, and so on.

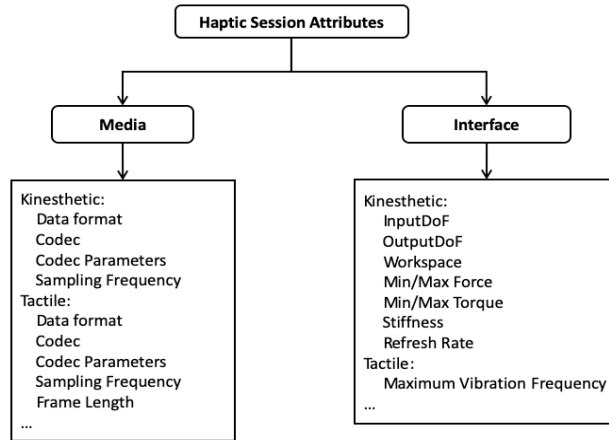


Figure 14—Classification of haptic session attributes as defined in HM

HM messages are defined based on ASN.1 (Abstract Syntax Notation One) for providing a technology-neutral description of the attributes. The ASN.1 schema for haptic handshake and media communication messages are presented as follows. The message header of a handshake message includes “packet type” for indicating the type of message (request, response, or ACK) and “packet sequence number” for facilitating retransmission in event of packet losses. In addition to the preceding fields, the media messages headers include “haptic timestamp” to indicate the time of generation of the haptic sample being carried in the payload. On the other hand, while the payload of HM messages carries attributes, the payload of media messages carries actual haptic information complying with the data format specified in HM phase. Further, examples of these messages are also presented for facilitating easy implementation. The schematic representation of HM packet format for the request, response, and data messages is shown in Figure 15.

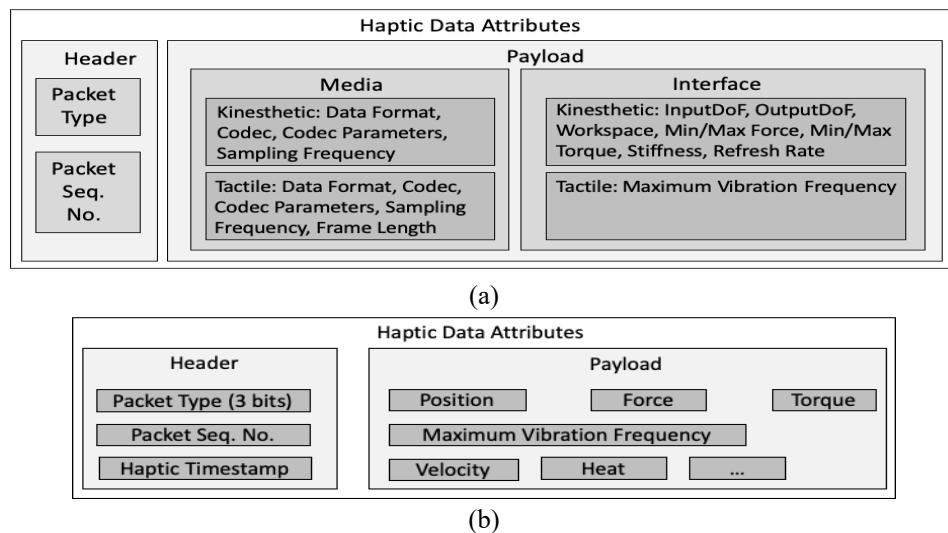


Figure 15—Schematic representation of HM packet format for (a) request/response, (b) data

The schema for the request and response messages is shown in Figure 16. The schema for the ACK message is shown in Figure 17.

```

DATA ::= SEQUENCE{
    header Header,
    payloadAttr HapticAttributes
}

Header ::= SEQUENCE{
    packetType VisibleString ("Request" | "Response"),
    packetSeqNo INTEGER
}

HapticAttributes ::= SEQUENCE {
    media Media,
    interface Interface
}

Media ::= SEQUENCE {
    kinCodec VisibleString ("KCnD" | "KCwD") OPTIONAL,
    tacCodec VisibleString ("PVC-SLP") OPTIONAL,
    kinCodecParam REAL OPTIONAL,
    tacCodecParam REAL OPTIONAL,
    kinSampleFrequency REAL DEFAULT 1000 OPTIONAL,
    tacSampleFrequency REAL DEFAULT 2800 OPTIONAL,
    tacFrameLength REAL,
    ....
}

Interface ::= SEQUENCE {
    kinInputDOF REAL (1..10),
    kinOutputDOF REAL (1..10),
    kinWorkspace SEQUENCE (SIZE(3)) OF REAL,
    kinForceRange SEQUENCE (SIZE(6)) OF REAL OPTIONAL,
    kinTorqueRange SEQUENCE (SIZE(6)) OF REAL OPTIONAL,
    tacMaximumVibrationFrequency REAL,
    ....
}

```

Figure 16—Schema for the request/response message
[The numbers are just examples as introduced before.]

```

DATA ::= SEQUENCE{
    header Header
}

Header ::= SEQUENCE{
    packetType VisibleString ("ACK"),
    packetSeqNo INTEGER
}

```

Figure 17—Schema for the ACK message

7.3 Handshake and data packets for the standard codecs

This subclause presents a realization of the handshake and data communication protocol for the two haptic codec standards, namely the kinesthetic codec with no delay (KCnD) and the tactile codec (TC). A similar example can be created for the case of KCwD.

7.3.1 Example of a KCnD handshake and data communication

The use case shown in Figure 18 is utilized to demonstrate the handshake and media communication for KCnD. In this scenario, two heterogeneous haptic interfaces communicate over a computer network with desirable QoS parameters. Codecs are deployed to encode the outgoing kinesthetic data stream and decode the incoming kinesthetic data stream from the remote node. The communication is established using the haptic handshake protocol (request, response, and ACK messages) described in 7.1, followed by haptic media communication.

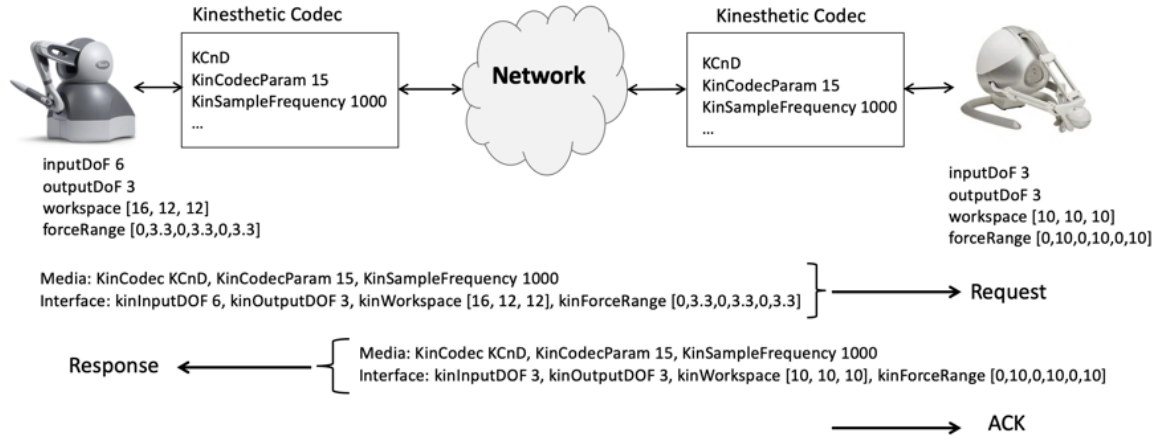


Figure 18—Example of kinesthetic codecs handshake

The leader device has the following specifications: input DoF = 6, output DoF = 3, workspace = [16, 12, 12] cm, and a force range of [0, 3.3] N along the three axes. The follower device is characterized by the following: input DoF = 3, output DoF = 3, workspace = [10, 10, 10] cm, and a force range of [0, 10] N along the three axes. The KCnD codec specifications include a sampling rate of 1000 Hz. The leader sends the request message shown in Figure 19. The follower replies with the response message shown in Figure 20. Finally, the ACK message is sent from the leader back to the follower to confirm the negotiated specifications and start the haptic data communication (Figure 21).

```
DATA ::= {
  header{
    packetType "Request",
    packetSeqNo 1
  }
  payloadHapticAttr{
    media{
      kinCodec "KCnD",
      kinCodecParam 15,
      kinSampleFrequency 1000,
      ...
    }
  }
}

interface{
  kinInputDOF 6,
  kinOutputDOF 3,
  kinWorkspace{
    16,
    12,
    12
  }
  kinForceRange{
    0, 3.3,
    0, 3.3,
    0, 3.3
  }
}
```

Figure 19—Request message from leader to follower

```
DATA ::= {
  header{
    packetType "Response",
    packetSeqNo 1
  }
  payloadHapticAttr{
    media{
      kinCodec "KCnD",
      kinCodecParam 15,
      kinSampleFrequency 1000,
      ...
    }
  }
}

interface{
  kinInputDOF 3,
  kinOutputDOF 3,
  kinWorkspace{
    10,
    10,
    10
  }
  kinForceRange{
    0, 10,
    0, 10,
    0, 10
  }
}
```

Figure 20—Response message sent from follower to leader