

A Candidate Hardware and Software Reference Setup for Kinesthetic Codec Standardization

Amit Bhardwaj*, Burak Cizmeci*, Eckehard Steinbach*, Qian Liu†, Mohamad Eid‡, José Araújo §, Abdulmotaleb El Saddik¶, Ritu Kundu||, Xun Liu||, Oliver Holland||, Mark A. Luden **, Sharief Oteafy †† and Venkatesha Prasad ‡‡

*Technical University of Munich, Germany

Email: {amit.bhardwaj, burak.cizmeci, eckehard.steinbach}@tum.de

†Dalian University of Technology, China. Email: qianliu@dlut.edu.cn

‡New York University Abu Dhabi, UAE. Email: mohamad.eid@nyu.edu

§Ericsson Research, Email: jose.araujo@ericsson.com

¶University of Ottawa, Canada. Email: elsaddik@uottawa.ca

|| King's College London, United Kingdom. Email: {ritu.kundu, xun.2.liu, oliver.holland}@kcl.ac.uk

**The Guitammer Company, USA. Email: mark@guitammer.com

††DePaul University, USA. Email: soteafy@depaul.edu

‡‡Technical University of Delft, Netherlands. Email: r.r.venkateshaprasad@tudelft.nl

Abstract—Recently, the IEEE P1918.1 standardization activity has been initiated for defining a framework for the Tactile Internet. Within this activity, IEEE P1918.1.1 is a task group for the standardization of Haptic Codecs for the Tactile Internet. Primary goal of the task group is to define/develop codecs for both closed-loop (kinesthetic information exchange) and open-loop (tactile information exchange) communications. In this paper, we propose a reference hardware and software setup for the evaluation of kinesthetic codecs. The setup defines a typical teleoperation scenario in a virtual environment for the realization of closed-loop kinesthetic interactions. For the installation and testing of the setup, we provide detailed guidelines in the paper. The paper also provides sample data traces for both static and dynamic kinesthetic interactions. These data traces may be used for preliminary testing of kinesthetic codecs. The paper also provides links for the download of both the reference setup and the data traces.

I. INTRODUCTION

Tactile Internet (TI) is “a network or network of networks for remotely accessing, perceiving, manipulating or controlling real of virtual objects of processes in perceived real time by humans or machines” [1]. This definition of the Tactile Internet has been proposed by the IEEE P1918.1 Tactile Internet working group. When a human remotely controls a robotic system over the TI, and receives force feedback, this is typically referred to as a human-in-the-loop TI scenario. In order to achieve interoperability in this context, standardization of the exchange of information is required. The recently started standardization activity IEEE P1918.1 has been initiated to define a framework for the Tactile Internet, including descriptions of various application scenarios, definitions and terminology, functions, and technical assumptions.

Within the IEEE 1918.1 Tactile Internet working group, IEEE P1918.1.1 is a task group for the standardization of haptic codecs. This standard will 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 will define (perceptual) data reduction algorithms and schemes for both closed-loop (kinesthetic information exchange) and open-loop (tactile information exchange) communications. The standard will also specify mechanisms and handshaking protocols for the exchange of the haptic device characteristics (e.g. workspace, degrees-of-freedom (DoF), amplitude range, temporal and spatial resolution, etc.) to enable their compatibility.

The IEEE P1918.1.1 task group has already finalized the requirements for the haptic codec development. The requirement document primarily mentions the components needed for both kinesthetic (with or without delay in the communication channel) and tactile codecs, and defines the basic structure of both objective and subjective quality metrics required for the evaluation of the codecs. Additionally, the task group also mentions the need for reference hardware and software setups for the evaluation of the codecs. The finalized requirement document is available at the following link: <https://cloud.lmt.ei.tum.de/s/2YfvLKmxG17GkM7>.

Recently, the task group has announced a call for contributions (CfC) for the kinesthetic codec development. The CfC document is available at the following link: <https://cloud.lmt.ei.tum.de/s/FsuqwMXoq8d6p09>. In order to evaluate the performance of the contributions, there is a need for a reference setup. In this paper, we propose such a reference setup. This reference setup is available to the community so that possible contributors for the kinesthetic codec development may test their solutions. Along with this reference setup, we also provide sample data (force, position/velocity) traces, assisting the contributors in developing their kinesthetic codec.

This paper is organized as follows. In Section II, we provide the description of the reference setup. Guidelines for installing and testing the setup are explained in Section III. In Section

IV, we explain and show the recorded data traces. The paper is concluded in Section V.

II. REFERENCE SETUP DESCRIPTION

The aim of this paper is to develop a candidate reference hardware and software setup for the evaluation of the kinesthetic codec contributions to IEEE P1918.1.1. A kinesthetic codec is primarily required in a teleoperation scenario for reducing the packet rate of the haptic modalities. Thus, the reference setup should provide a scenario of a typical teleoperation where haptic modalities (position/velocity, force/torque) along with other modalities (audio/video) are exchanged between an operator (master) and a teleoperator/robot (slave). During a closed-loop kinesthetic interaction with a real or virtual environment, typically position/velocity signals are sent from the operator to the teleoperator (forward channel), and the resulting force feedback signals are sent from the teleoperator to the operator (backward channel). Before the transmission of the kinesthetic signals between the operator and teleoperator, the signals are encoded by a kinesthetic codec.

We have developed a reference hardware and software setup for the above mentioned purpose. In the setup, a scenario of a typical teleoperation is realized in a virtual environment with the help of a widely used haptic application development platform called *chai3d*¹. The virtual environment consists of a rigid cube (movable) lying on a rigid planar surface, as shown in Figure 1. The stiffness properties of the objects in the scene may be updated as per the requirement. The small gray colored ball in the figure represents the haptic device in the virtual space. To realize the teleoperation scenario in the virtual environment, we consider the virtual tool as the end-effector of the teleoperator. The operator controls the position and velocity of the end-effector through a haptic device. While interacting with the environment, 3 degree of freedom (DoF) force feedback is computed, and received by the operator whenever the virtual tool is in contacts with the objects in the environment. Based on this setup, one may realize both static and dynamic kinesthetic interactions: interactions with the rigid planar surface are termed as static, and interactions with the movable cube as dynamic.

The physics computations of the interactions are performed by ODE (Open Dynamics Engine) which is already available in *chai3d*. For visual and haptic rendering, we run two threads in parallel with sampling rates of 50 Hz and 1000 Hz, respectively. To control the motion of the end-effector, the setup provides two different modes: position, and velocity control. In the position control mode, the position of the slave at a given instant of time is estimated from the position information received from the master while in the velocity control mode, the position is estimated from the velocity information received from the master. During the teleoperation, the control mode can be adjusted online. Additionally, communication delay may also be simulated in the forward and backward channels by buffering the signal before transmission. All these

¹available at the following link: <http://www.chai3d.org/>

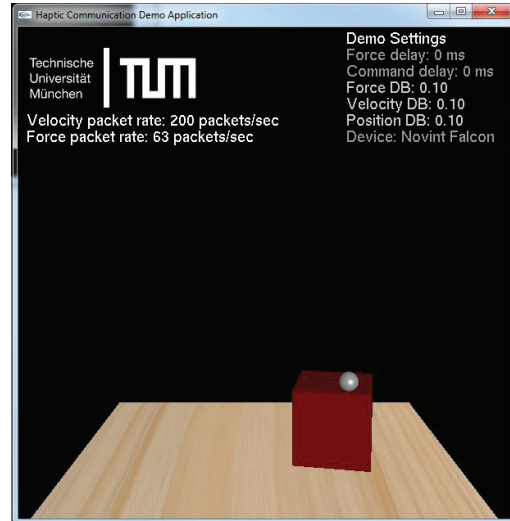


Fig. 1. A snapshot of the virtual environment designed for the reference setup.

parameters can be configured in the configuration file of the setup. For example, the configuration file defines the control mode (flag=0 for position control, and flag=1 for velocity control), delay to be introduced in forward and backward channel, and whether or not a noise filter on the velocity signals is enabled (flag=0: disabled, flag=1: enabled). In the configuration file, we have also given an option to record the signals. A simple MATLAB script is provided to plot the results of the teleoperation session.

It should be noted that a control-architecture is required to maintain the stability of the global control loop of the teleoperation in the presence of communication delay. For further details, the readers are referred to [2], [3], [4], [5]. In the current version, the reference setup does not include any stabilizing control architecture. Thus, the setup should only be used when no communication delay is introduced. However, the setup may be used in explaining how the delay effects the stability and transparency of the teleoperation. A suitable control architecture will be included in the next version of the reference setup, which will be used for the evaluation of contributions for kinesthetic codec for delayed teleoperation scenarios.

In order to compress haptic (position/velocity, force/torque) information, a kinesthetic codec is required in the setup. This means that before the exchange of kinesthetic signals between the master and slave, a suitable compression scheme is employed for encoding the signals. On the receiver side, the signals are decoded with the help of a suitable reconstruction scheme. For example, the force signals are reconstructed at the master side, and the position signals are reconstructed at the slave side. The signals should be encoded and decoded in such a way that perceptual transparency of the teleoperation is maintained.

For illustrating the principle of a kinesthetic codec in the setup, we include a well-known kinesthetic data compression

scheme which is based on the concept of perceptual deadbands (PD) [6]. This compression scheme is based on Weber’s law. According to the perceptual deadband approach, only if the relative difference between two stimuli exceeds a certain perceptual threshold (also called Weber fraction or just noticeable difference or perceptual deadband (DB) parameter), the signal will be transmitted. Therefore, only perceptually significant samples are transmitted. At the receiver side, a zero-order hold strategy is implemented to interpolate the signals at the rate of 1KHz (as per the requirement of the local control loops). For further details about this compression scheme, the users are referred to [6], [7].

In the snapshot of the virtual environment shown in Figure 1, the parameters of the perceptual deadband based codec are seen under “Demo Settings”. Here “Force delay” and “Command delay” correspond to delays introduced in backward and forward channel, respectively. And, “Force DB”, “Velocity DB” and “Position DB” denote the deadband parameters for the force, velocity and position signals, respectively. The figure also shows the velocity packet rate (forward channel), and force packet rate (backward channel) generated by the perceptual deadband based kinesthetic codec.

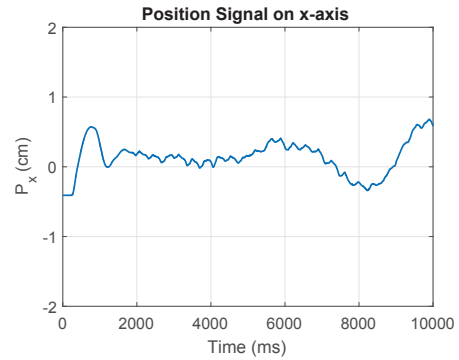


Fig. 2. An illustration of the hardware setup: user interacting with the virtual environment by a Falcon haptic device.

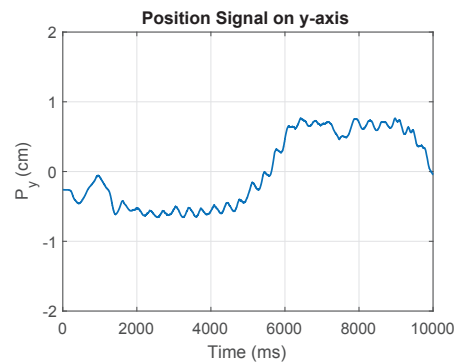
In Figure 2, we show an illustration of the hardware setup. Here a user is seen interacting with the virtual environment by a Falcon haptic device [8] (the default haptic device for the setup). The proposed reference setup is independent of the design/structure of kinesthetic codecs being used. The users may test/evaluate the performance of their developed kinesthetic codec with the setup. In the next section, we provide guidelines for the installation and testing of the reference setup.

III. GUIDELINES

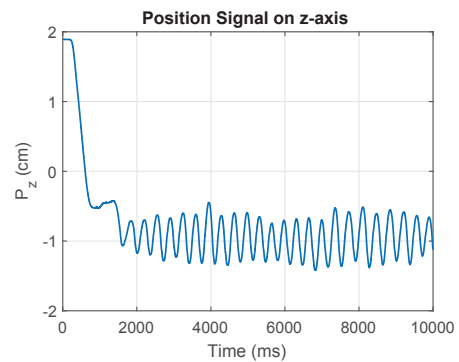
The setup can be downloaded from the following link: <https://cloud.lmt.ei.tum.de/s/xZLJtqe08eguNxW>. The application has been developed with chai3d-2.0.0 and Microsoft



(a)



(b)

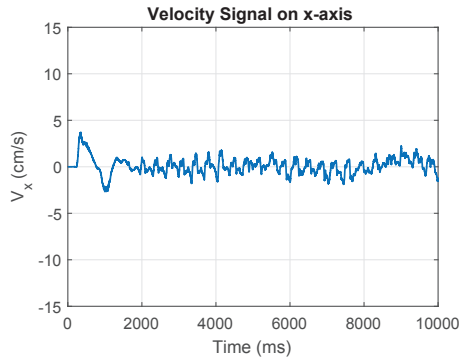


(c)

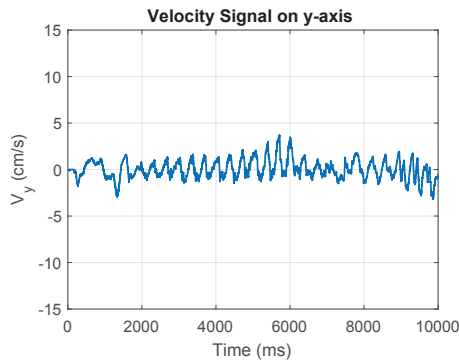
Fig. 3. Position data traces of the operator for the first scenario of static interaction.

Visual Studio 2013 on a Windows 7 platform. There is a “Readme.txt” file in the main project document which defines the instructions and guidelines of the setup in detail. Here we briefly mention the key instructions and guidelines. The users are requested to follow the guidelines for the successful testing of the setup.

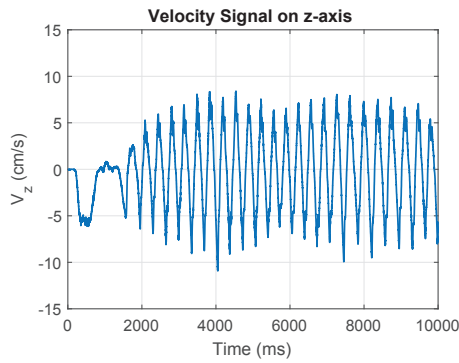
- 1) *Installation of haptic device:* The setup has been tested with a Falcon haptic device. The main project document folder contains the driver of the haptic device. In order to install the driver, run “setup.Falcon.v4.0.28.0_100707.exe” and follow the installation instructions of the haptic device.



(a)

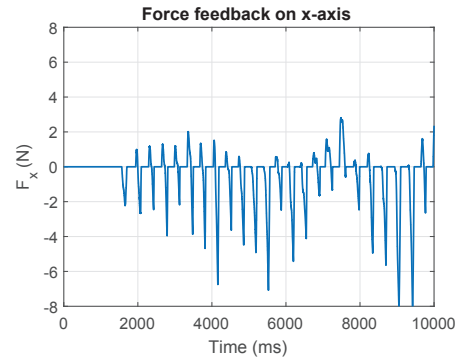


(b)

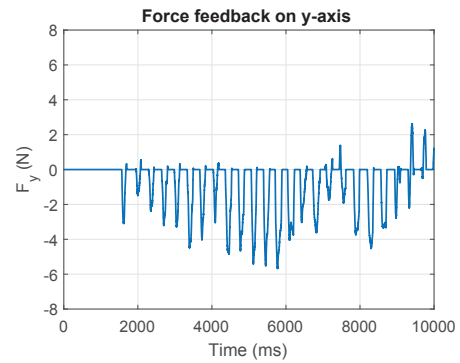


(c)

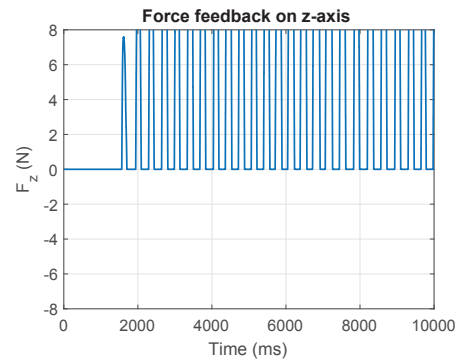
Fig. 4. Velocity data traces of the operator for the first scenario of static interaction.



(a)



(b)

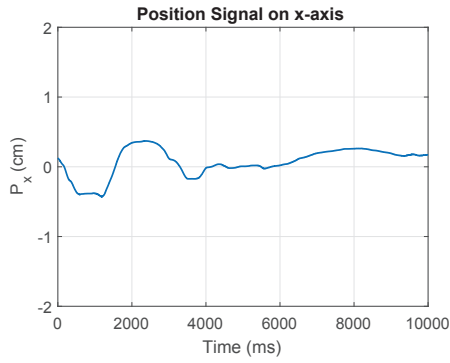


(c)

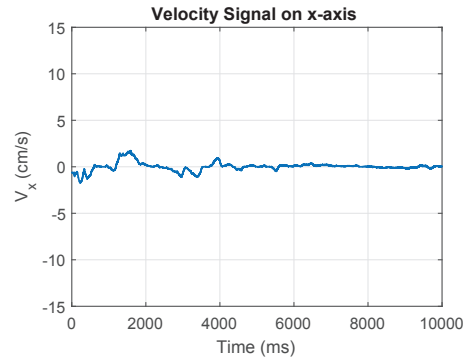
Fig. 5. Force data traces of the teleoperator for the first scenario of static interaction.

- 2) *Device calibration*: When the falcon device is run for the first time, it needs to be calibrated to display the force feedback. Otherwise, no force feedback will be perceived. If the device is not calibrated, a red color is seen on the front light emitting diode (LED) of the device. To calibrate, move the device so that all joints reach their limits. When the LED turns blue, the device is successfully calibrated.
- 3) *Installation and compilation of chai3d*: The project document is under the following path: “\examples/msvc9/CHAI3D.sln”. Click on the solution to open it with MS Visual Studio 2013. There are two projects:

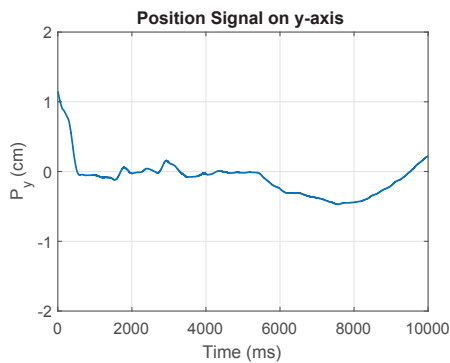
- CHAI3D and HapticCommunicationDemo. Build the solution, and set the HapticCommunicationDemo project as “StartUp Project” by right clicking on the project.
- 4) *Running demo application*: Run the application using the play button in Visual Studio (debug mode). In real-time, there is an option to change the codec parameters. Please follow the instruction screen of the application for the relevant keys as defined in the “Readme.txt” file.
- 5) *Workspace adjustment*: If the limits of the physical workspace of the device are reached, press the middle button of the Falcon device and stop the movement of



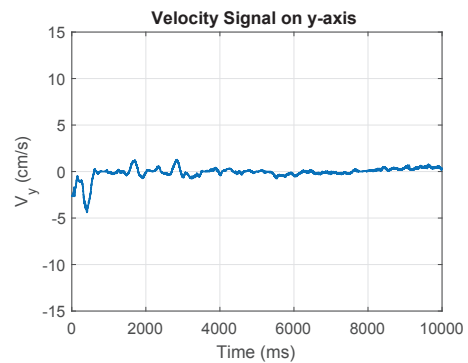
(a)



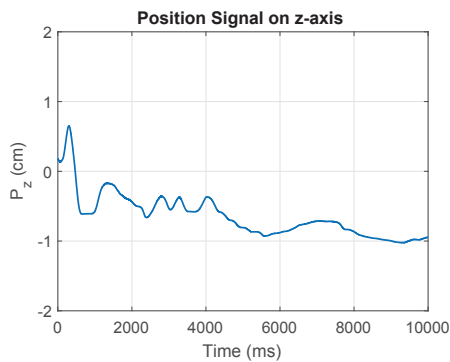
(a)



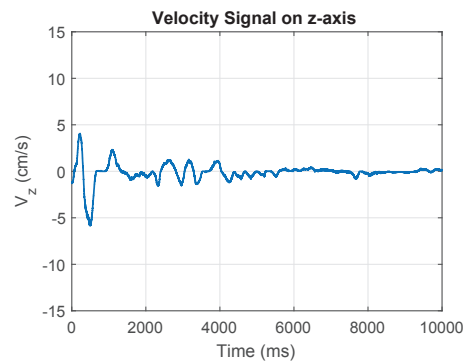
(b)



(b)



(c)



(c)

Fig. 6. Position data traces of the operator for the first scenario of dynamic interaction.

Fig. 7. Velocity data traces of the operator for the first scenario of dynamic interaction.

the virtual tool. Readjust the device handle in the center of the workspace and press the button again to restart controlling the virtual tool. This enables the user to explore all regions of the virtual workspace.

- 6) *Reset button and exit application:* The application provides two keys to control the virtual environment in real time. For moving the cube to its initial position, press the key “[r]” (reset button), and for stopping the application, press the key “[x]” (exit button). The application also provides other control keys to update the parameters of the codec in real time. More details can be found in the “Readme.text” file.

- 7) *Analyzing the results:* If “RecordSignals” flag is enabled in “config.cfg”, kinesthetic signals (position, velocity, force) for both the operator and teleoperator are recorded. Under the project path, there exists the following MATLAB script to plot the recorded signals: “/HapticCommunicationDemo/AnalyzeResultsDB.m”

The path of the haptic communication library is “/examples/msvc9/HapticCommunicationDemo”. All necessary source files are located under “/src” folder. The configuration file “config.cfg” is located under “/cfg” folder. Before running the application, adjust the desired codec parameters in “config.cfg”.

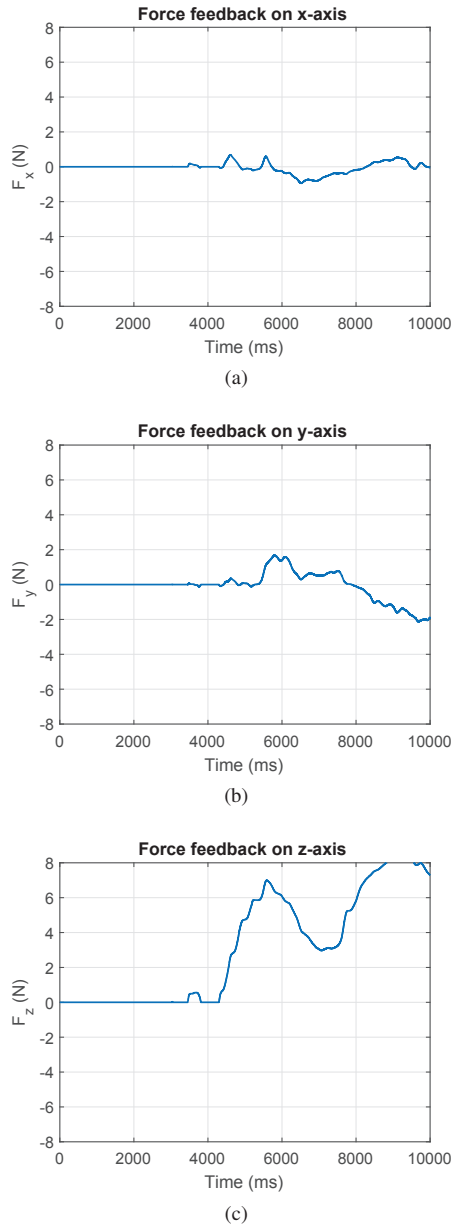


Fig. 8. Force data traces of the operator for the first scenario of dynamic interaction.

IV. DATA TRACES

We have recorded raw data traces (position and velocity signals of master, force signal of slave) for both static and dynamic interactions. While recording the data traces, the perceptual deadband based kinesthetic codec was "OFF" (i.e., Force DB=0, Velocity DB=0 and Position DB=0, as shown in Figure 1). The data traces are recorded at a sampling rate of 1KHz.

For static interaction, data traces have been recorded under the following two scenarios: (1) tapping on the fixed surface with the tool, and (2) dragging the tool on the fixed surface. Similarly for dynamic interaction, we have recorded the data

traces under the following two scenarios: (1) moving the cube on the fixed surface by pressing on its top face with the tool, and (2) moving the cube on the fixed surface by pushing one of its side faces with the tool. For both scenarios of static and dynamic interactions, we have recorded the data traces for 25 seconds. In Figures 3, 4 and 5, we show a portion of the three dimensional position, velocity and force traces recorded for the first scenario of static interaction, respectively. Similarly, Figures 6, 7 and 8 show a portion of three dimensional position, velocity and force traces recorded for the first scenario of dynamic interaction, respectively. The complete recorded data traces can be found at this link: <https://cloud.lmt.ei.tum.de/s/4FmHUCsoUvwR1e3>. These recorded data traces may be used for preliminary testing of the contributions for the kinesthetic codec.

V. CONCLUSION AND FUTURE WORK

In this paper, a reference setup has been proposed for the evaluation of the contributions for the kinesthetic codec development in IEEE P1918.1.1. We have also recorded data traces for both static and dynamic interactions. Both the reference software and the data traces can be downloaded from the links mentioned in the paper. In the future, we will include a suitable control architecture in the setup, which will ensure the stability of the control loops in the presence of delay in the communication channel, and the resulting setup will be used in evaluating kinesthetic codecs for delayed teleoperation scenarios.

REFERENCES

- [1] Tactile Internet Emerging Technical Subcommittee. [Online]. Available: <http://ti.committees.comsoc.org/>
- [2] R. Anderson and M. Spong, "Bilateral control of teleoperators with time delay," *IEEE Trans. on Automatic Control*, vol. 34, no. 5, pp. 494–501, 5 1989.
- [3] J.-H. Ryu, J. Artigas, and C. Preusche, "A passive bilateral control scheme for a teleoperator with time-varying communication delay," *Mechatronics*, vol. 20, no. 7, pp. 812–823, 10 2010.
- [4] X. Xu, B. Cizmeci, A. Al-Nuaimi, and E. Steinbach, "Point cloud-based model-mediated teleoperation with dynamic and perception-based model updating," *IEEE Trans. on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2558 – 2569, 5 2014.
- [5] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach, "Haptic data reduction for time-delayed teleoperation using the time domain passivity approach," in *IEEE World Haptics Conference, Chicago, USA*, Jun 2015.
- [6] P. Hinterseer, S. Hirche, S. Chaudhuri, E. Steinbach, and M. Buss, "Perception-Based Data Reduction and Transmission of Haptic Data in Telepresence and Teleaction Systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 588–597, 2 2008.
- [7] E. Steinbach, S. Hirche, J. Kammerl, I. Vittorias, and R. Chaudhari, "Haptic data compression and communication for telepresence and teleaction," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 87–96, 1 2011.
- [8] Novint Falcon I. [Online]. Available: <http://www.novint.com/index.php/novintfalcon>