TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Kommunikationsnetze
Fachgebiet Medientechnik

Compression and Transmission of Haptic Data in Telepresence and Teleaction Systems

Dipl.-Ing. (Univ.) Peter Hinterseer

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Sandra Hirche
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. Eckehard Steinbach
	2. apl. Prof. Dr.-Ing., Dr.-Ing. habil. Hugo Fastl

Die Dissertation wurde am 20.10.2008 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 17.04.2009 angenommen.

Acknowledgement

This dissertation evolved from a very enjoyable period of my life working as research scientist and teaching assistant at the Media Technology Group of the Institute of Communication Networks at the Technische Universität München. There were quite some people during this time, without whom this work would not have been possible as it is presented here.

First and foremost, my sincere thanks go out to my supervisor Eckehard Steinbach, who made it possible for me to work on a very challenging but also quite unexplored topic, which is really not easy to find in the current research landscape. Both this main research interest and the occasional but always modest teaching assignments formed the most fruitful experiences during the last years which will certainly come in handy in my future professional career. Looking back, Mr. Steinbach always had an open ear for interesting ideas and new approaches and despite his very crammed schedule always found time to take care of his assistants while always leaving enough liberty and leeway that own ideas could freely grow and develop.

I would also like to thank Prof. Dr.-Ing. Hugo Fastl for accepting to be the second auditor of this dissertation and Prof. Dr.-Ing. Sandra Hirche for heading the committee.

Always a source of inspiration were my colleagues in the institute and in the research project, the Sonderforschungsbereich 453, funded by the DFG. Ingo Bauermann, although not directly involved with my topic, always brought in new interesring and inspiring perspectives to consider. The cooperation with Julius Kammerl was even more fruitful. Large parts of Chapter 4 originate from a close cooperation with him. Together with Sandra Hirche the main contribution of this dissertation was developed and initially tested at the Institute for Automatic Control Engineering, which is also responsible for the opportunity to use sophisticated hardware including a real telepresence and teleaction system for my experiments. Prof. Subhasis Chaudhuri was also very helpful in the development of the theoretical groundwork presented in this work.

Work would not have been half as enjoyable without my other colleagues and the administrative staff at the Institute. Thanks for all the technical and personal help, the always pleasant climate, and the joyful conversations and activities during the not-so-busy hours of the day.

Finally, I want to thank my parents for always supporting me in every possible way and last but certainly not least I thank my lovely girlfriend Freya who is always there for me, especially when I need her most.

Munich, October 2007 *Peter Hinterseer*

Abstract

Complex multimedia systems like telepresence and teleaction systems are rapidly gaining relevance in both modern science and technology. The challenging integration of various multimodal data streams is a main topic in current research activities. While the processing and coding of video and audio data as the two most important modalities for the human being has been excessively thematized in the past years, the haptic modality was neglected in comparison. Haptics, representing all kinds of information the human being is able to perceive via the means of touch, movement, force, and orientation, pose numerous interesting challenges as far as data reduction and real-time transmission are concerned. This dissertation presents a broad spectrum of information about the haptic modality focusing on possibilities to reduce the amount of data transmitted in haptic teleinteraction systems without impairing the functionalities or even endangering the user. After presenting signal statistics based compression approaches the main contribution of this thesis, the so called deadband approach, is described in detail. This approach exploits the perception capabilities of human beings to reduce the amount of packets transmitted in haptic teleinteraction systems. The deadband approach and its extensions are thoroughly tested in various experiments, ranging from simple one degree-of-freedom perception threshold examinations over three degree-of-freedom haptic interaction with virtual remote environments to the examination of its influence on task performance in a real telepresence and teleaction system. It is found that it provides significant data reduction gains while maintaining the performance of the systems. Also a theoretical point of view is given by the mathematical derivation of the output signal behavior when using the deadband approach on different kinds of input signals leading to further understanding of the theoretical foundation the approach is based on. In summary, this dissertation proves the legitimacy of the presented data reduction approach for haptic data and also defines its limitations and rules of application.

Kurzfassung

Telepräsenz- und Teleaktionssysteme finden zunehmend Anwendung in wichtigen Zukunftsgebieten wie der Medizintechnik, der Raumfahrttechnik, der Produktionstechnik und der Sicherheitstechnik. Eine grosse Herausforderung in diesem Zusammenhang ist dabei die Integration der vielen multimodalen Datenströme, die in einem solchen Szenario auftreten. Während die Verarbeitung, Codierung und Übertragung von Video- und Audiodaten in den letzten Jahren intensiv behandelt wurde, wurde die haptische Modalität im Vergleich stark vernachlässigt. Haptische Daten, also zusammenfassend alle Arten von Informationen, welche der Mensch durch Berührung, Bewegung, Kraftausübung und seine Orientierung im Raum wahrnehmen kann, stellen interessante Anforderungen an die verwendeten Verfahren bei der Echtzeitübertragung und Kompression. Diese Dissertation bietet ein weites Spektrum an Informationen über die haptische Modalität als solches und legt dabei ihren Schwerpunkt auf Möglichkeiten, die Menge an übertragenen Daten in Teleinteraktionssystemen zu reduzieren ohne dabei deren Funktion oder Sicherheit zu beeinträchtigen. Neben Ansätzen, welche die Signalstatistik für die Kompression ausnutzen, wird der sogenannte Deadband-Ansatz vorgestellt. Dieser Ansatz nutzt die Schwächen und Grenzen der menschlichen Wahrnehmung aus, um die Anzahl der übertragenen Pakete in haptischen Teleinteraktionssystemen so gering wie möglich zu halten. Der Deadband Ansatz und seine Erweiterungen werden sowohl durch Simulation als auch experimentell untersucht. Die Experimente reichen von einfachen Aufbauten mit nur einem Freiheitsgrad über die Interaktion mit einer virtuellen entfernten Umgebung in drei Freiheitsgraden bis hin zur Untersuchung des Einflusses der Datenreduktionsverfahren auf die Aufgabenperformanz in einem realen Telepräsenz- und Teleaktionssystem. Ein Hauptergebnis dieser Arbeit ist, daß der vorgestellte Deadband-Ansatz die übertragene Datenmenge signifikant reduziert, ohne die Wirklichkeitsnähe des Systems maßgeblich zu beeinträchtigen.

Neben praktischen Untersuchungen, geht diese Arbeit auch auf die theoretischen Hintergründe des Deadband-Ansatzes ein. Das Signalverhalten bei der Anwendung des Verfahrens auf verschiedene Arten von Eingangssignalen wird mathematisch modelliert und analysiert. Die Ergebnisse führen zu einem grundlegenderen Verständnis der Leistungsfähigkeit und Anwendbarkeit des untersuchten Ansatzes. Zusammengefasst lässt sich sagen, daß diese Dissertation die Einsatzfähigkeit des präsentierten Datenreduktionsansatzes für haptische Daten erfolgreich verifiziert und dabei ebenfalls über seine Schwächen und möglichen Einsatzgebiete aufklärt.

Contents

Chapter 1

Introduction

1.1 Telepresence and Teleaction Systems

Telepresence and teleaction (TPTA) systems have been and still are the subject of intensive interdisciplinary research covering the areas of network communication, computer science, robotics, system theory, control theory and psychology. TPTA systems allow a human operator to be present in and interact with a remote environment that can for example be distant, scaled to macro or nano dimensions, hazardous for a human being or even virtual reality. This is achieved by the sensing, transmitting, and displaying of multiple modalities the human operator needs for natural interaction and the feeling of presence.

A TPTA system as visualized in Figure 1.1 consists of three main components: the human system interface (HSI), the teleoperator (TOP) and the communication link connecting those two. The HSI consists of input devices, typically haptic devices for position and orientation input, and output devices for multiple modalities (e.g. head mounted device for stereo-video, headphones for audio) and the haptic devices for force and torque feedback. By means of the HSI the human operator (OP) commands the position and velocity of the teleoperator during the observation of and interaction with the remote environment through multi-modal feedback. The teleoperator itself is a robot equipped with multiple sensors (e.g., video-camera, microphones, and force/torque sensors) and, in order to be able to interact with the remote environment, it is also equipped with haptic actuators (e.g., grippers or more anthropomorphic limbs). The multimodal sensor data including the interaction force with the environment is fed back to the HSI and displayed to the operator. The communication link bidirectionally transports the multimodal data streams. Under the assumption of an ideal TPTA system, the operator feels as if he were in place of the robot interacting with the remote environment or, in other words, feels completely

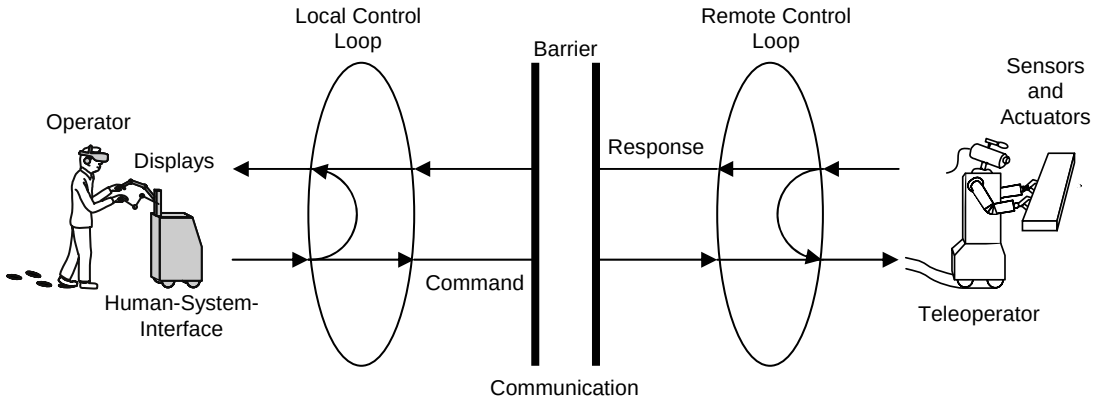

Figure 1.1: General overview of a TPTA system.

immersed.

The main focus of this dissertation lies on the haptic (force feedback) subsystem, specifically on the data reduction and transmission of haptic data streams within a TPTA system.

1.1.1 Application Scenarios

Today, only a limited number of haptic teleoperation systems are available. This is first and foremost because of the immense technological and interdisciplinary challenges those systems pose. Nevertheless, in the near future technology will be advanced enough to enable TPTA systems for the use in a vast number of application scenarios. In the following we take a closer look at some of the most promising approaches.

1.1.1.1 Telemaintenance

In complex industrial scenarios maintenance of machines and other complex technological systems is one very important factor. In case of failure an expert for the respective system has to be found and this expert in most cases has to personally inspect and evaluate the problem before being able to fix it. Personal attendance of highly trained personnel is often necessary and therefore requires additional travel time. This constant traveling significantly decreases the work efficiency of the expert.

In one possible scenario in the near future the expert stays at one place or even at home and only operates his HSI which he uses to connect to a corresponding maintenance-TOP at the site where the failure occurred. He is then able to quickly evaluate the problem

using the sensors and actuators of the TOP and eventually fix it right away or can initiate additional tasks like ordering spare parts. Once he is done at one site he can disconnect und reconnect to the next client within minutes. No need to travel for many hours or even days. This is both very positive for the expert himself, who no longer needs to travel, and the client, who gains much shorter response times to maintenance requests.

Somehow related to telemaintenance, telepresence in hazardous environments concentrates on the work in environments where a human being is not able to perform tasks or is at least strongly handicapped by protective equipment. Examples for this are the maintenance work inside a radiation exposed nuclear reactor core, operation under high pressures of the deep sea or tasks to perform in the cold and gravityless regions of space.

Systems as described in [56, 57, 48, 64, 60, 59] show the research in the direction that possibly makes this vision reality in the not-so-distant future. Features like bimanual haptic interaction and the possibility to explore wide remote environments build the foundation for an immersive and transparent TPTA system for telemaintenance.

1.1.1.2 (Tele-)Education

Teleeducation is already done in multiple forms like students attending lectures over the Internet via Audio-/Video-Streaming. The special case of haptic teleeducation is not yet used. For this type of teleeducation, a student can be taught in multiple ways: By using an affordable haptic display device at home, the student can try to fulfill a mechanic task like making a knot using surgical instruments. A supervisor at the university can watch the progress of the task and give tips and hints how to improve their performance. It is even possible to have the supervisor haptically guide the student as if he held the students hand. The student then feels the guidance forces and they help him/her in the learning process.

Furthermore it is possible to store haptic interaction data for educational purposes. For example the trajectory and contact forces of a complicated procedure executed by the supervisor can be stored and used later on to guide the students in learning the procedure. The student then is either guided along the trajectory of the procedure or presented with the contact forces while watching the procedure's video recording or a combination of both.

Examples for haptic (tele-)education can be found in [71, 74, 35, 18] but this is only a small sample of research activities in this field.

1.1.1.3 Microassembly

Due to the ongoing trend of downsizing technological systems in order to make them easier to handle, increase their efficiency and lowering their material cost it becomes more and more difficult to handle the mechanical parts of such systems. A human being is not able to handle objects below a certain size without technological help. And even with help of microscopes and tweezers the miniaturization of processed objects is limited and handling becomes more difficult and tiresome for a human being.

TPTA systems in the context of microassembly tasks offer the possibility to scale sensed haptic data like forces and positions from a micro-environment to a manageable size for a human. With such a system, given appropriate sensors and actuators exist in the respective size, it would be possible to directly and comfortably interact with objects the size of microns or even nanometers.

Exemplary work in the direction of microassembly can be found in [72, 42, 58] where both micro and nanomanipulation systems are developed and evaluated. Of course a combination of microassembly and telemaintenance or teleeducation is possible but also complicates the involved systems.

1.1.2 Haptic Data

1.1.2.1 Classification

The main novelty of TPTA systems in comparison to well known remote controlled systems is the transmission of haptic data. Haptic data (from the Greek *haptesthai*: to touch) in general consist of any type of mechanical quantity a human being is able to acquire with its sense of feeling. This ranges from positions and velocities over forces and torques to temperature, roughness and viscosity only to name popular examples.

Generally a differentiation between kinesthetic, proprioceptive and tactile information is made:

- Kinesthetic data can be described as the kind of data that is sensed by special organs in muscles, tendons, joints, and bones and is stimulated by body movement and stress on extremities and limbs. Kinesthetic data can be positions, orientations and derivatives thereof (velocity, angular velocity, acceleration, angular acceleration), forces and torques.
- Proprioceptive data is sensed within the human body and gives information about body posture mainly based on data by sensors in skeletal joints, the inner ear, and

the central nervous system. This possible types of data are similar to the examples given for kinesthetic data but in this case they affect the whole body instead of only extremities and limbs.

- Tactile data is mainly sensed by the skin. A number of different mechanoreceptors near the surface of the skin provide sensory to acquire environmental contacts in the frequency range of 50-350Hz and can thereby differentiate several properties like temperature, roughness, geometry and slippage.

1.1.2.2 Haptic Data in TPTA Systems

Haptic data signals as used in TPTA systems represent the progression of mechanic quantities like position, angle, velocity, angular velocity, acceleration, angular acceleration as well as force and torque over time. These quantities are sampled by respective sensors at a constant rate, typically in the range 500–1000Hz with a typical resolution of 16bit per Degree of Freedom (DoF). The high sampling rate is necessary to encompass the whole range of human haptic perception as well as, in the context of TPTA systems, to ensure the tracking performance and stability of the local control loops at the HSI and the TOP. Transmitted data is mostly of kinesthetic nature because this type requires the least technical effort to sense and display. Proprioceptive displays as used in flight simulators or theme park thrill rides are rarely used in TPTA systems as well as tactile displays, which usually require a large amount of micromechanical effort and are therefore hard to place at the most important locations of human interaction, the hands and fingertips.

Currently available TPTA systems like bimanual systems with additional finger force feedback easily have more than 20 DoF. At every sampling instance, i.e., 500 to 1000 times per second, a data packet consisting of all current sample values is generated and sent over the communication link. The bidirectionally transmitted position/velocity and force/torque data packets serve as reference input for the local control loops at the HSI and the TOP. As can be seen from the structure of a TPTA system, where the HSI is used to control positions at the TOP which eventually lead to force feedback at the HSI which again influences position input, a global control loop is closed by the communication system. It is clearly evident that the communication introduces transmission delay in this control loop. Furthermore, delay in a closed loop system leads to instability if not treated by appropriate control measures [17]. The more delay is introduced the more conservative the control has to be designed in order to guarantee stable operations at any time. This on the other hand leads to a strong deterioration of immersiveness, and in many cases to the inoperability of the TPTA system [39, 38, 55, 54, 70, 13]. Thus, hard real-time constraints apply for the haptic data stream, and this constitutes the fundamental difference to standard streaming

multimedia such as video and audio. The primary design goal for a communication system for TPTA systems is to keep the introduced delays and the amount of data being sent as small as possible.

Usually dedicated lines or high speed local area networks are used for the transmission of haptic data. Coming from its ubiquitous nature it becomes more and more desirable to use the Internet as the communication infrastructure. Unfortunately, the transmission characteristics of the Internet are far from being optimal. For efficient transmission of haptic data over the Internet, data reduction techniques have to be used to lower the amount of data and possibly even the rate at which packets are transmitted. Reduced packet rates put less demand on the Internet connection and therefore lower the probability of congestion along the transmission path. High packet rates (1000Hz) as we can observe them for traditional transmission of haptic data are hard to maintain over the Internet [50]. One possible solution to this challenge is the real-time coding of haptic data.

1.1.3 Frequency Characteristics of Haptic Data

By analyzing the position, velocity, force and torque signals transmitted in a TPTA-system in the frequency domain, some characteristic properties can be observed. During free space movement using the HSI, i.e., no environment contact takes place, quick changes in direction and sudden accelerations or decelerations rarely occur. The inertia of human extremities in combination with the used force feedback device itself limits the velocity and rapidity of human movements. Thus, the corresponding velocity signal mainly consists of low frequencies. As soon as the TOP hits an object in the remote/virtual environment, this resulting force feedback is signaled back to the OP. In this case, movement quickly comes to a halt which directly leads to a transient velocity signal with broad energy distribution in the frequency domain. Similar characteristics can be recognized in the force signal transmitted from the TOP which is displayed to the human OP by the HSI. As long as the TOP does not encounter direct contact to objects in the remote environment, there will be no force feedback signaled and displayed to the OP. Hitting a stiff object in the remote environment results in a sudden peak in the generated force signal which is also characterized by a broad energy distribution in the frequency domain. In order to keep in contact with the encountered object, the OP usually maintains a low contact force against the object, either to make sure not to lose contact or to explore its surface. The OP's applied contact force and the corresponding force feedback signal may vary over time but usually show smooth signal behavior in this state of manipulation. Hence, the corresponding force feedback signal shows most energy in low frequency regions. Figure 1.2 depicts a few seconds of a typical force feedback signal recorded during a TPTA experiment. The sudden force replies from the TOP when contact with a remote object in the distant

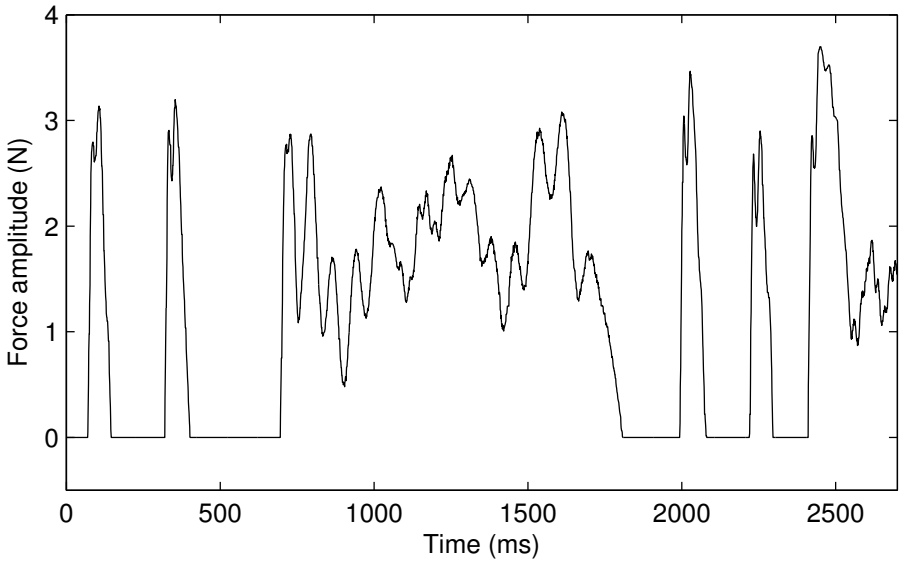

Figure 1.2: An examplary haptic force signal.

environment occurs are clearly visible. Also, when the operator is keeping contact with the touched object we see the smoothly varying change in the amplitude of the force signal.

Summarizing these observations, the velocity and force signals in telepresence systems consist mainly of low frequency content. Only at the event of sudden contact in the remote environment, the signals tend to become transient which leads to a broad energy distribution in the frequency domain. This however is only true if the used velocity and force/torque sensors deliver a signal with almost no noise. While this is true for most force/torque sensors of better quality, the velocity signal is almost always generated as the first order derivate of the position signal and therefore contains quite high noise levels.

1.1.4 Coding of Haptic Data

The multimodal data transmitted in TPTA systems consists of both well known types of data like video streams (both mono and stereoscopic) and audio streams and also of haptic data generated by force/torque or position/velocity sensors. In the past, but also today significant research efforts in the fields of audio and video coding can be observed. In contrast, the coding of haptic data has received only limited attention. With recent advances in virtual reality, telerobotics, and telepresence and teleaction however, the topic is rapidly gaining relevance. Once haptic data has to be transmitted or stored an interest in compressing this kind of data emerges.

Fundamentally, a differentiation of real-time or in-the-loop coding and offline coding is

necessary. The aforementioned constraints on the communication of haptic data only apply to real-time coding. But of course some research effort has been made on the offline coding of haptic data as well. Apart from that differentiation it has to be mentioned that in principle the coding of data in general can be either lossy or lossless. For obvious reasons, lossy compression achieves much higher compression gains compared to lossless coding but introduces errors in the signal. The delicate part of lossy coding is to introduce a kind of error which does not or only marginally influence the performance of the system the data is used in.

First efforts in offline coding can be found in [66, 67] where a number of different approaches of effectively sampling haptic data and first applications of DPCM and ADPCM are shown for mostly lossless offline coding purposes. In some scenarios so called haptic sessions are recorded and stored. Instead of using conventional compression techniques like ZIP or RAR, techniques especially tailored to the coding of haptic data yield significantly higher coding gains as shown for example in [62] where, among other techniques, the main contribution of this dissertation is used to achieve lossy compression ratios of 127:1 and above for so called "haptic media files", which basically are compilations of recorded haptic data streams with some side information.

Much more interesting, especially in the context of TPTA systems, is the real-time coding of haptic information. Traditional block-based compression techniques are not applicable because of the additional delay introduced by building blocks of data. Stream-based techniques like differential coding followed by entropy coding (e.g., [13] or Section 2.1) work quite well with up to 90% packet payload reduction. Approaches using DPCM, ADPCM and Huffman Coding [52, 37] for real-time coding also work well, but they all suffer from a bad packet header to payload ratio. For instance, if a haptic data stream carries 3-DoF data with 16bit resolution the payload of one packet is 6byte. In comparison, the UDP/IP header of this packet is 28byte. Additionally, approaches suffering from the negative effects of differential coding like vulnerability to packet loss are obviously less suitable for packet-based communication in TPTA systems. But coding of haptic data is not only of interest in TPTA systems. In [21] compression based on predictive coding is applied in order to improve the communication between a haptic device and the controlling host with respect to sampling rate and data rate. In this context the efficient coding of haptic data is beneficial to the fidelity and transparency of a haptic input/output device.

All the mentioned approaches have in common that they only exploit statistical signal properties for data reduction. In comparison, the approach presented in this dissertation exploits the characteristics of human haptic perception in order to reduce the packet transmission rate in networked TPTA systems. To our best knowledge, this is the first psychophysics-based haptic data reduction approach in scientific literature.

1.1.5 The Human Haptic System

Since the main goal of the work presented in this dissertation is the effective coding of haptic data based on the exploitation of limitations of human haptic perception, we need to take a look at the mechanisms a human being uses to perceive and generate haptic information.

1.1.5.1 Cutaneous Perception

The skin is the largest sensor of the human body. It is responsible for cutaneous perception and is therefore equipped with a number of different receptors. The combination of the stimuli sensed by these receptors results in haptic or tactile sensation.

First we take a look at the mechanoreceptors which are responsible for sensing mechanical stimuli. The four types in the skin are [31, 22, 69]:

Merkel receptor
: This receptor has the shape of a disc and is mostly placed between the upper skin (epidermis) and lower skin (dermis). It senses pressure stimuli in the frequency range from 0.3Hz to 10Hz in small, well defined areas. It slowly adapts its perception behavior to applied stimuli and is therefore able to sense intensities of stimuli.

Meissner corpuscle
: This is a conglomerate of a number of flat-shaped cells. It is located in the dermis right beneath the epidermis and senses low frequency vibrations from 3Hz to 50Hz. Its field of perception is small and well defined and adapts itself rapidly to stimuli.

Ruffini cylinder
: This receptor consists of many fibers located within a hull shaped roughly like a cylinder. It is located within the dermis and mainly senses stretching and static force stimuli in the frequency range from 0Hz to 10Hz in a large indistinct area. Its adaptation to stimuli is rather slow.

Pacinian corpuscle
: Shaped like an onion, this is a capsule around a nerve fiber located deep within the skin in the subcutaneous fat areas. It is basically an acceleration and vibration detector in the frequency range from 10Hz to 500Hz. It covers a large, indistinct area of skin and rapidly adapts to stimuli.

The adaptation behavior of receptors can be described as follows: Slow adaptation means that the receptor senses the stimulus for a very long time before the perception fades while the stimulus is still there. For example a watch is not consciously felt at the wrist. After

some time we don't feel it anymore although it is still there. Rapid adaptation of course is the exact opposite. Only the initial event once a stimulus occurs is sensed by a rapidly adapting receptor. One could say that a slowly adapting receptor senses the state of a stimulus whereas a rapidly adapting receptor senses the change of a stimulus.

Note that the four receptor types cover all possible combinations of large and small area of reception and slow and rapid adaptation to stimuli.

Since the human skin is also able to feel temperature there are also so called thermoreceptors [31, 22, 45]. There are separate receptors for warm and cold:

Warm fibers
: These receptors respond to increasing temperatures in the range from $30°C$ to $48°C$ where higher temperature leads to more intensive sensation. It does not respond to mechanical stimulation and is located in the dermis.

Cold fibers
: Located directly beneath or within the epidermis, the sensation created by these receptors gets higher with decreasing temperatures. The detection range of these receptors starts at $5°C$ and ends at $43°C$.

In the temperature range from $30°C$ to $36°C$ a human being usually does not perceive any temperature chages of the skin. Outside of this range, perception is essentially provided by the associated receptor type.

If mechanical stimulation is so large that tissue damage is imminent or the skin is exposed to temperatures above $45°C$ or below $-15°C$ another type of receptor comes into play: the pain receptors or nocireceptors. Nocireceptors have receptive fields of around 25mm^2 and lead to a pain response in the brain when triggered.

The spatial resolution of thermoreceptors is lower than those of nocireceptors or mechanoreceptors but is increased by combining the sensations of the three.

The combination of all those receptor types enables the high resolution of the tactile perception of a human being.

1.1.5.2 Kinesthesia

Kinesthesia describes the ability of a human being to sense the position and movements of the body and its parts. This information is taken and consolidated from a couple of different receptor types [22].

Joint position receptors

These are mainly the already known Ruffini cylinders and Pacinian corpuscles which are located within the joints together with free nerve endings. During body movement the pressure on those receptors changes according to the joint position. This sensation of pressure is interpreted by the central nervous system as joint position. Consequently the change in pressure is evaluated as angular motion of the joint.

Muscle tension receptors

There are two types of receptors within muscles. Muscle spindles are located in between muscle fibers and can detect their respective movements created by both passive and active stretching. Golgi tendon organs are located between a muscle and its respective tendon and measure muscle tension. These two receptors in combination can sense both tension strength and rate of increase/decrease of tension and are therefore the most important force receptors during bodily interaction with the environment.

In combination with the cutaneous senses a perception of forces and positions in a very large range is possible.

1.1.5.3 Sensory Motor Control

The interaction of the human motor system, consisting of muscles, joints, tendons, and bones and the sensory system described above constitutes the human sensory motor system. Since haptic interaction with objects requires delicate control of forces and positions, the human developed a very powerful sensory motor control mechanism. The simple example of grasping and lifting a wet glass illustrates the function principle of sensory motor control.

First the visual system localizes the glass and we move our hand in the right position to grasp the glass with all four fingers and the thumb. As soon as our fingertips touch the surface, the combined sensation of thermoreceptors and mechanoreceptors tells the brain that the object is wet and that slippage may be a problem. In the next step force is applied by the fingers to build up enough friction that the glass can be lifted. Here both the tactile receptors in the fingertips and the kinesthetic receptors in the muscles and joints serve as sensors in the force control loop to keep the applied force stable and at an adequate level. Next the hand is moved upwards to lift the glass. As soon as gravitation pulls the glass down, shear forces and possible slippage are detected and measured in the fingertips by the respective mechanoreceptors. In the following the grasping force is adaptively increased to reliably prevent slippage. This force control usually takes place reflex based at an unconscious level to prevent unnecessary delays in the control loop. After this a steady grasp is established and the glass can be manipulated further.

Of course this description is quite simplified but still shows the complex interaction of sensors and actuators of the human body in order to achieve a relatively simple task.

The main interaction tools of the human are its hands. The human hand with its four fingers and the opposing thumb is a very versatile gripping and interacting device. It allows both high grasping forces and stability using the so called power grasps where both palm and the whole fingers are used to encompass an object as well as highly dexterous manipulation using the so called precision grasps where only the thumb and one or more fingertips are used to precisely manipulate an object.

1.1.6 Limitations of the Human Haptic System

Human perception has been investigated intensively during the past two centuries. In this context, the respective perceptual properties for all kinds of stimuli put on the human body have been studied. Despite the very sophisticated nature of human haptic perception and interaction there are still limitations in various aspects of haptic sensations and motor control. We are especially interested in those limitations because we want to exploit them in the coding process.

1.1.6.1 Resolution and Bandwidth

First we take a look at the resolution and bandwidth requirements of human haptic interaction since these are the first candidates of properties a data reduction system could base upon. According to the properties of the receptors used to sense haptic information the frequency resolution of human haptic perception ranges from 0 to 500Hz. Although [68] mentions that frequencies of up to 10000Hz can be required for skillful manipulative tasks, current teleoperation hardware runs at sampling frequencies of 1000Hz which directly reflects the capabilities of the receptors in the skin.

It is also important to note that human input and output capabilities are very asymmetric what haptic information is concerned. While being able to sense frequency stimuli of up to 500Hz the human haptic system is only able to produce frequencies of up to 10Hz.

1.1.6.2 Detection Thresholds: Weber's Law

Although absolute perception thresholds, e.g., the smallest stimuli a human being is able to detect, are known for haptic perception, this property is hardly usable for coding purposes. Relative detection thresholds, e.g., the amount by which a stimulus has to change before

this change becomes perceivable, on the other hand represent a good starting point for perceptual coding.

Just like a lot of other perceptual modalities, human haptic perception can be well approximated by Weber's Law. Ernst Weber was an experimental physiologist who in 1834 first discovered the linear relationship

$$\frac{\Delta I}{I} = k \quad \text{or} \quad \Delta I = kI \tag{1.1}$$

between the stimulus intensity I and the smallest still perceivable change of the stimulus intensity ΔI. The constant k is in modern psychophysics literature referred to as Weber constant or Just Noticeable Difference (JND) (in older literature ΔI has been called JND) or Differential Threshold. The JND is measured in psychophysical experiments and represents a statistical rather than an exact quantity. The JND usually reported is the difference that a person notices in 50% of the trials.

For haptic perception, i.e., force, limb position, and velocity, the JND is in the range from 5% to 15%, depending on the type of stimulus and the limb/joint where it is applied ([44, 22, 19]). This means that if, for example, the displayed force at the HSI changes its magnitude by less than the JND, the operator would not notice this change.

1.2 Chapter Summary

Various interesting application scenarios for TPTA systems with even more to come in the near future are the basis for more intensive research in the field of coding and transmission of haptic data which plays an important role in such systems. The unique set of properties and challenges this kind of multimedia data presents, leads to approaches which are different from standard audio and video coding.

In order to be able to exploit limitations of human haptic perception and sensory motor control a deeper knowledge of the underlying mechanisms is very useful and can directly lead to first data reduction approaches when it is combined with the knowledge of the characteristics of examplary haptic signals.

Chapter 2

Compression and Data Reduction of Haptic Data

In the following a number of possible approaches for the data reduction and compression of haptic data is presented. We will use the following definitions of terms compression and data reduction throughout the rest of this work. The term compression is usually used for approaches which try to code information using less bits than in the original representation. Source coding is another term for data compression. Data compression can be both lossy or lossless. In contrast to that the term data reduction describes a lossy approach of minimizing the amount of data by omitting part of the information. This omission takes place in the original representation of the data.

First a lossless compression approach for haptic data is evaluated. After that the main contribution of this dissertation is presented: a psychophysically motivated data reduction approach for real-time haptic data streams, the so called deadband approach.

2.1 Differential Entropy Coding

Due to the relatively high sampling frequency required for haptic data streams and the frequency characteristics described in Section 1.1.3, the changes in sample values of a degree of freedom from one sample to the next are relatively small. During operation with the telepresence system presented in [13] subsequent samples are either identical or only change by a small amount. (see Figure 2.1). Such a distribution is well compressible using a DPCM-based compression scheme. Only the changes from one sample to the following are transmitted or in other words the current sample serves as the first order prediction of the following sample and only the prediction error is encoded. As a sampling resolution

of 16bit per DoF is enough in most cases the resulting range after differential coding is about 7-10bit wide. Entropy coding of the prediction error allows for the use of adaptive codeword lengths depending on the relative frequency of occurrence of the difference to code. For example the most frequent input symbol, in this case the 0, would yield a very short codeword of the length of only one or two bit. Less frequent input symbols are represented by longer codewords.

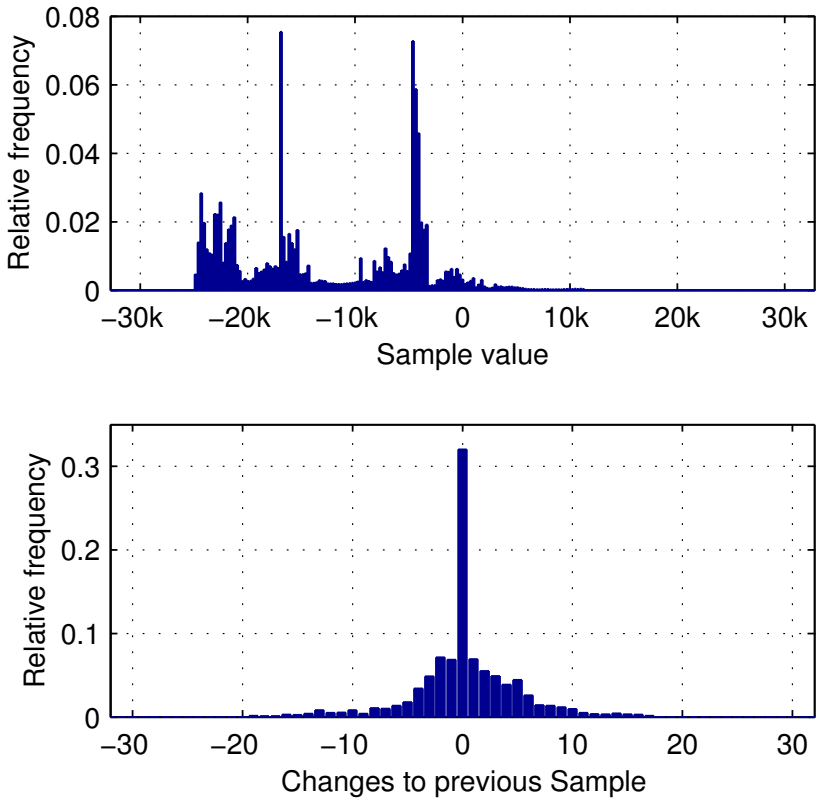

Figure 2.1: Relative frequency of positions and their changes from sample to sample in the TPTA-System presented in [13].

2.1.1 Non-Adaptive Huffman Coding

The first approach presented in literature uses Huffman coding [41] as the entropy coding scheme. More specifically, the non-adaptive variant of this source coding method is used. In order to encode the input symbols a probability table has to be created first. This is done by recording the relative frequencies of the signal to code as shown for example in Figure 2.1. Of course, the coding efficiency is higher if the exact signal statistics are known. Unfortunately, this is not the case for the real-time application scenario since there

is no time to analyze the data to encode beforehand. The only possibility in this case is the statistical analysis of a large number of sample data sets for the specific application scenario.

The generated probability table, in which the relative frequencies of all possible input symbols are recorded, is then used to generate the so called Huffman Tree (see [41] for details) which offers a very efficient way of encoding and decoding of the input symbols. Depending on the implementation only very little computational resources are necessary for the en/decoding. Usually a certain number of DoFs are sampled every sampling instance and have to be transmitted immediately because of the restrictions of haptic real-time communication (see 1.1.2.2).

2.1.1.1 Procedure

The procedure in a real-time TPTA system is as follows. For every sampling instance a set of n quantized sample values is read from the sensors of a n-DoF system. In the next step differential coding for every DoF is performed. After that, the n differences of the set are encoded one after the other using a different Huffman Tree. Individual probability tables are used for each DoF of the signal so that a total number of n different tables are necessary. The signal properties of the different DoFs of a haptic data stream can vary considerably. Therefore it is not a good idea to use only one table for all DoFs.

Since the resulting Huffman code is prefix free, which means a symbol is decodable without knowing its length beforehand, it is possible to directly concatenate the encoded symbols of the different degrees of freedom even if they use different Huffman trees for encoding. The only thing the decoder needs to know is the order in which the DoFs are encoded. In this way very efficient bit streams can be built because almost no side information is needed to decode when the sequence and the respective probability tables for the DoFs are known at the receiver side.

2.1.1.2 Table Misses

As mentioned, the tables we use to encode differential haptic data are 7 to 10bit wide. Nevertheless it is still possible that values occur after differential coding that do not fit into this table since differential coding by itself does not decrease the size of the source alphabet. If a value after differential coding is not covered by the probability table we call this a table miss. In order to be able to treat table misses we introduce a special symbol in the source alphabet which marks table misses. If a table miss occurs, the special symbol

is encoded and the absolute value of the table miss before differential coding (full 16bit length) is appended directly after the special symbol.

This special treatment of table misses is not necessary by default. If we used probability tables double the size of the source alphabet before differential coding, every possible value would be covered. But for the special properties of haptic data this method works well because table misses are very rarely encountered. The reduced memory consumption and reduced complexity by having small source alphabets outweighs the very small increase in redundancy introduced by the way table misses are treated here.

2.1.1.3 Results

Since haptic data has very favorable characteristics for differential Huffman coding some experimental measurements have been made. For example for haptic data with 16 bit resolution and 500 Hz sampling rate as encountered during operation of the TPTA system in [13], the differential Huffman coding scheme reduced the amount of data communicated to about 10% to 25% of the uncompressed rate.

2.1.2 Adaptive Arithmetic Coding

Arithmetic Coding (AC) is an efficient approach for redundancy reduction in data streams. Just like Huffman coding it is a method of entropy coding. In comparison to Huffman coding, AC overcomes the limitation of needing a fixed number of bits as the coded representation of an input symbol. Therefore it is possible to get very close to the entropy of the source signal which represents the theoretical lower bound. AC has been proposed by *Abramson* and *Elias* in the 1960s. In 1976 first finite precision implementations were shown and in 1979 and 1980 the algorithm as it is still used today has been proposed in [61] and [34]. It is well described in [20] and since it is known to be a very good candidate for entropy coding it was further investigated for its usability in TPTA systems.

2.1.2.1 Properties

One important property of AC is that it produces an incremental, non-final code. That means that if a symbol sequence abc is encoded to a bit sequence xyz, this bit sequence xyz may only be decoded to ab. The symbol c may be decodable as soon as the next symbol is encoded or even later. In order to be able to decode the complete sequence, either the number of encoded symbols has to be known in advance or a special symbol which marks the end of the sequence has to be introduced.

Non-final code also means that we cannot determine which bits in the encoded bitstream represent a certain input symbol. We always need the complete preceding bit sequence in order to decode the stream at a certain position.

Once we want to use AC to compress haptic data in real-time, a number of additional challenges arise.

2.1.2.2 Procedure

The arithmetic coding of haptic data works almost the same as for Huffman coding (see 2.1.1.1). Of course the n different DoFs after being differentially coded are processed by an arithmetic coder with the corresponding probability table. The result is only one bitstream, where all the n differences are encoded and multiplexed. This stream has to be sent to the receiver and can be decoded there using the respective decoder and probability tables.

2.1.2.3 Adaptivity

AC can be applied either with or without adaptive probability tables. In comparison to adaptive Huffman coding, where the Huffman-tree has to be rebuilt from time to time to adapt to changing signal properties, with AC it is very simple to implement signal dependent behavior. One can even start off with equally distributed symbol probabilities and build the table from scratch.

For haptic data, it makes sense to keep track of the last couple of seconds and build the table from the relative occurrences of values in that time. For this, a FIFO queue with the most recent 1 to 10 thousand samples (depending how localized the adaptation should work) is kept in memory. The probability table is the table of the relative number of occurrences of values in this queue. With every incoming new value, the respective counter is increased by one. At the same time the counter of the value which falls out at the end of the queue is decreased by one.

In case static tables shall be used, a table can be built from the recordings of a number of experiments.

2.1.2.4 Table Misses

Table misses if the source symbol lies outside the 7-10bit wide probability table are treated almost identically as in the case of Huffman coding. The only difference is that the 16bit-

sample before differential coding is appended after the bit in the bitstream which leads to a decodable out-of-table special symbol.

2.1.2.5 Bit Sequence

Because of the non-final code structure it is impossible to know where a transmitted bit sequence ends. In packet based networks the smallest data unit used is 8bit long (byte). If a transmitted bit sequence ends somewhere in the middle of a byte and the rest of the byte is filled with zeroes, the receiver is by no means able to distinguish whether the zeroes at the end of the stream belong to the code or whether they are only left empty. One way to get around this problem is to send the number of valid bits along with the sequence. This clearly introduces additional redundancy in the data stream. The other possible way is to only send complete bytes of code. We consider this to be the better solution although it introduces at least one sampling step of additional delay.

2.1.2.6 Finishing Sequence

Generally the code generated from a number of symbols can not be decoded back to the original symbols right away. Some information of the last symbols is still in the encoder, represented by the current encoding interval bounds (see [20] for details). Once the next symbols are encoded the first set of symbols becomes decodable. However, it is possible to make the sequence decodable right away. To do this, the final code has to be produced by flushing all residual data from the encoder. This is done by generating a number of bits which clearly define a value inside the final interval in the encoder. Once this finishing sequence is appended to the produced code, it becomes decodable. The drawback of this approach is the redundancy added by the finishing sequence and that the encoder and decoder have to be restarted afterwards.

2.1.2.7 Delay

If we use the method of sending only complete bytes of the encoded bitstream we face the problem that we cannot send data directly after encoding. Even if the encoded bitstream is exactly a multiple of 8bit long we would not be able to decode it at the receiver because it is non-final code.

For example if one set of sample data is encoded and results in 27bit of code, we can not send those 27bit to the receiver yet. We would have to use zero-padding to fill the last byte up and the receiver would not be able to distinguish between code and padding bits.

But even if one set of input data would result in 32bit of encoded bitstream it would be of no use to send those 4byte right away because the last symbol of the set would not be decodable yet.

One possibility to circumvent any delays is the application of the finishing sequence after every set of data. But because of the redundancy the finishing sequence and the zero padding introduce we do not use this method.

Once the following set of data is encoded and appended to the bitstream, we can send a number of complete bytes and can be sure that the previous set of values can be decoded. We can say we have an inherent one step delay in the system. At a sampling frequency of 1000Hz which is generally used for haptic data this one time step equals 1ms. This delay is considered not critical.

2.1.2.8 Delay Constraint

Because of the fact that AC makes it possible to encode very frequent source symbols at an average of less than one bit per symbol it is possible that even over multiple time steps the encoder doesn't produce enough output code to fill a whole byte of data. In this case transmission should be forced in order to stay below a certain delay bound. There are two possible solutions for this case:

The first possibility is to introduce another special symbol into the alphabet which represents forced transmission. For forced transmission the encoder encodes this special symbol right after the last payload symbol and also appends the finishing sequence. Once the decoder receives and decodes this special symbol it restarts decoding from scratch with the next received packet of bitstream.

The second possibility is to use the finishing sequence directly after the last payload symbol and send the data right away. In this case it is necessary to tell the decoder that the bits following after the last payload symbol are not part of the code and that it has to restart decoding with the next packet. This can be achieved by a one bit field in the header of each packet (since some header information is almost always necessary) which is either zero if the decoder should restart decoding with the contained data or one if it is just code continuation.

The first approach should be taken if delay violations happen seldom. This can be the case because either the number of encoded values in every time step is large or the delay constraint is not very strict. The second approach should either be taken if one bit is available in the header anyway or forced transmissions are expected very frequently. This is mostly the case for small amounts of values in one packet or a very strict delay constraint.

2.1.2.9 Results

We can see that using adaptive arithmetic coding for the real-time transmission of haptic data is not trivial. Table 2.1 shows that some additional gain over Huffman coding is possible using this scheme. Transmitted data sets were always the same and both the Huffman coder and the AC without adaptation worked with the optimal probability table calculated from exactly the encoded data set. The adaptive AC always used the most recent 5000 samples to build its current table.

Type	average bit/symbol
Huffman	5.161
Static arithmetic	5.157
Adaptive arithmetic	4.660

Table 2.1: Compression effectiveness under optimal circumstances.

2.1.3 Compression based on Signal Statistics: Conclusion

Despite this significant compression gain achievable, differential entropy coding suffers from multiple downsides in the field of real-time haptic data transmission.

One drawback is that it leads to a problem in case of packet loss on the channel. The difference values affected by this disruption of the data stream do not reach the destination. Therefore a constant offset appears in the subsequently reconstructed values. To compensate for this offset, it is necessary to transmit absolute values every n samples or in case a packet loss was detected.

But the main reason why these approaches are not as efficient as hoped for is the bad packet header to payload ratio of a haptic data stream as already mentioned in the Introduction. Even very good compression on the payload itself is useless if a big share of the necessary network bitrate is caused by packet headers. Therefore, the compression schemes in the following sections aim at packet rate reduction rather than payload compression.

2.2 1-DoF Deadband Approach

The basic idea of the psychophysically motivated data reduction approach presented here is to transmit data only if the operator is likely to detect the change compared to previously transmitted data. Related schemes are the deadband approach applied in networked control

systems [53] where signal changes are not transmitted unless they exceed a certain fixed threshold and the well-known Δ-modulation ([65, 32]). In our approach, the threshold is not fixed but depends on the magnitude of the current output signal. The deadband that is formed by applying this threshold value with respect to the most recently transmitted magnitude value will be called p in the following. p is a percental value. The deadband principle as we apply it here is illustrated in Figure 2.2. The parameter k in Weber's Law (Equation 1.1) basically states the upper bound of the deadband p. If, for example, the user is presented with a force of 1N and the deadband is given with $p = 10\%$ the next force sample value is only transmitted once it goes either below 0.9N or above 1.1N. Every force change in the interval from 0.9N to 1.1N is considered imperceptible by the human operator and therefore not necessary to be transmitted. Once p is larger than k the deviation between the transmitted and original signal is likely to become perceivable to the user and interaction may feel distorted and the quality of immersion is reduced.

To apply the deadband algorithm, the magnitude of the difference d between an initial value x_i and a current value x_c has to be computed. This is done by calculating the absolute difference between those two sample values and comparing it to a threshold value (the deadband p multiplied by the initial value x_i).

$$\begin{aligned} d &= |x_i - x_c| \\ d &\leq |p \cdot x_i| \implies \text{Do nothing} \\ d &> |p \cdot x_i| \implies \text{Transmit new value } x_c \end{aligned} \tag{2.1}$$

As the control loops at the HSI and the TOP require an input signal at a constant high sampling rate, samples which are not transmitted have to be reconstructed at the receiver side. It is straightforward to apply a zero-order-hold strategy, where the value of the most recently received sample is held until a new sample arrives (hold last sample).

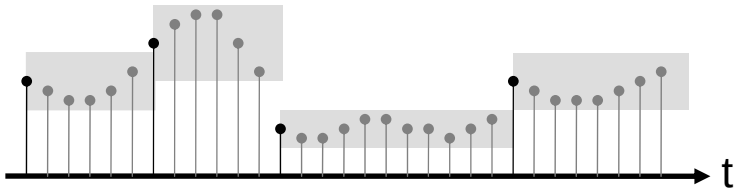

Figure 2.2: 1-DoF deadband applied to a signal. Grey samples are not sent. Black samples are signaled to the receiver.

Note that every data reduction in a closed loop system has an influence on the system dynamics, and as such possibly on the stability. Stability under lossy data reduction as well as effects of additional communication delay have been investigated in other works

of the author ([9, 8]), and will not be treated here. Furthermore, we will consider a position/velocity-force architecture here, where the position/velocity is transmitted from the HSI to the TOP and the force from the TOP to the HSI. The velocity as well as the force is processed using the deadband approach. In order to prevent a drift between the positions of the HSI and the TOP resulting from the lossy nature of the data reduction algorithm, additionally a position update is sent with every velocity packet.

The deadband data reduction approach will be abbreviated by DBDR in the rest of this work.

2.3 3-DoF Deadband Approach

In TPTA applications we often encounter haptic devices with multiple degrees of freedom (DoF), often 3. A 3-DoF device typically uses the three Cartesian components (or another representation of 3D space) of the current velocity or force. Applying the 1-DoF DBDR approach to every single component of the Cartesian representation is a straightforward extension, which, however, turns out to be very inefficient with respect to the data transmission rate.

If random movements with identically distributed directions and magnitudes of forces and velocities are examined, the component with the lowest magnitude and therefore the smallest deadband is mostly responsible for packet generation. The probability of having a component with low magnitude therefore increases with the number of components used. It becomes obvious that the more DoFs a system has, the less efficient the DBDR approach becomes if separately applied to each DoF.

To overcome the aforementioned limitation we propose a multidimensional deadband approach. This approach is motivated by recent psychophysical results [25] indicating the validity of the straightforward extension of Weber's Law to n dimensions. In the following we explain the extension of the one-dimensional deadband (a numeric interval, see Equation 2.1) to two dimensions where the deadband becomes a circular area. In 3D a spherical volume element serves as deadzone (we will denote a multidimensional deadband as deadzone from now on) and the extension is similar to the 2D case.

From now on vector values are considered and are denoted in bold letters $\mathbf{x}_i$ and $\mathbf{x}_c$, and d represents the magnitude of their difference. Accordingly, the deadzone algorithm reads as follows

$$d = |\mathbf{x}_i - \mathbf{x}_c|$$

$$\begin{aligned} d &\leq p \cdot |\mathbf{x}_i| \implies \text{Do nothing} \\ d &> p \cdot |\mathbf{x}_i| \implies \text{Transmit new vector } \mathbf{x}_c \end{aligned} \tag{2.2}$$

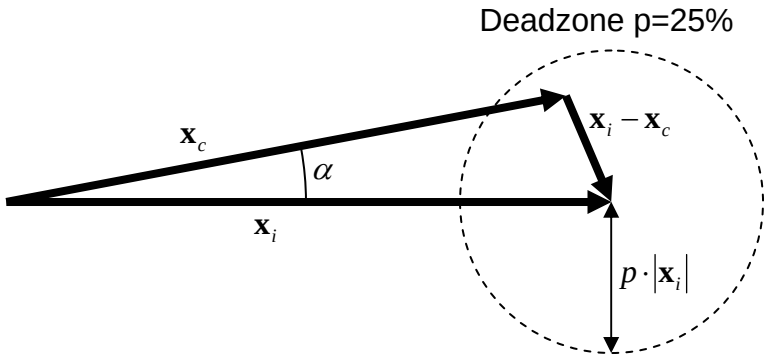

Figure 2.3: Geometrical description of a 2-DoF deadzone.

Figure 2.3 illustrates the resulting deadzone. The deadzone is represented by a circle around the tip of vector $\mathbf{x}_i$ with radius $p \cdot |\mathbf{x}_i|$. The angle between $\mathbf{x}_i$ and $\mathbf{x}_c$ is denoted by α. The multi-DoF deadband principle is visualized in Figure 2.4. If the tip of vector $\mathbf{x}_c$ lies within the deadzone circle, the deadband is not violated and thus no new value is transmitted. If the tip lies outside the deadzone circle, updated sample values are sent.

The circular shape of the deadzone makes it computationally easy to calculate whether the deadzone is violated or not. The size of the deadzone circle depends only on the length of vector $\mathbf{x}_i$ whereas the maximum of the angle α depends only on the deadband factor p.

The angle α reaches its maximum, when

$$\mathbf{x}_c \perp \mathbf{x}_i - \mathbf{x}_c \tag{2.3}$$

and

$$|\mathbf{x}_i - \mathbf{x}_c| = p \cdot |\mathbf{x}_i| \tag{2.4}$$

i.e., $\mathbf{x}_c$ is tangential to the deadzone circle. p is assumed to be significantly smaller than 1. So α_{max} can be calculated as follows:

$$\begin{aligned} \sin \alpha_{max} &= \frac{|\mathbf{x}_i - \mathbf{x}_c|}{|\mathbf{x}_i|} = \\ &= \frac{p \cdot |\mathbf{x}_i|}{|\mathbf{x}_i|} = \\ &= p \end{aligned} \tag{2.5}$$

$$\alpha_{max} = \arcsin p \tag{2.6}$$

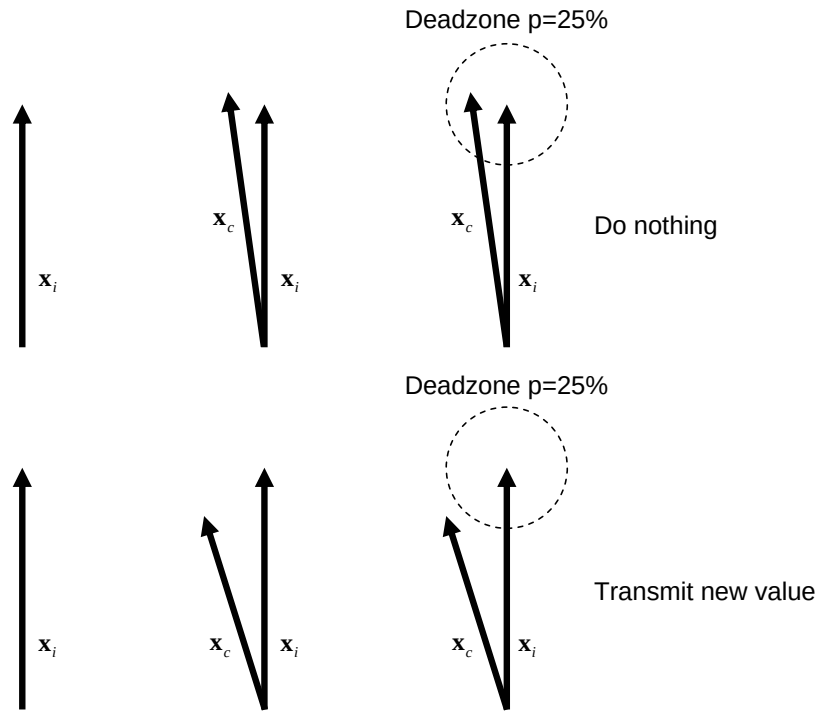

Figure 2.4: Criterion for transmission of new values in the 2-DoF case.

In this case $|\mathbf{x}_{c_{\alpha max}}|$ would be

$$\begin{aligned} \cos\alpha_{max} &= \frac{|\mathbf{x}_{c_{\alpha max}}|}{|\mathbf{x}_i|} \\ |\mathbf{x}_{c_{\alpha max}}| &= \cos\alpha_{max} \cdot |\mathbf{x}_i| \\ &= \cos(\arcsin p) \cdot |\mathbf{x}_i| \end{aligned} \tag{2.7}$$

This means that no matter how large a sampled 2-DoF variable (velocity, force, ...) is, once it changes its direction by α_{max} an updated value will be sent to the receiver. The multidimensional deadband algorithm hence provides constant directional sensitivity independent from the deadband parameter p. The implications of this important property of the isotropic deadzone are yet unknown and would be an interesting subject to further experiments.

The extension of this approach to 3D is straight forward. The vectors $\mathbf{x}_i$ and $\mathbf{x}_c$ become 3-dimensional, the circular deadzone becomes a spherical deadzone. The tip of $\mathbf{x}_c$ has to lie outside this sphere to trigger an update value. The values of α_{max} and $|\mathbf{x}_{c_{\alpha max}}|$ stay the same, because the vectors $\mathbf{x}_i$ and $\mathbf{x}_c$ define a plane in which the above calculations hold.

2.4 Extensions of the Deadband Approach

2.4.1 Model-based Prediction

In order to further reduce the number of transmitted packets, signal prediction is used on both sides of the system as shown in Figure 2.5. On the OP side a force predictor is used to estimate future force values from the incoming force values. On the TOP side the same predictor is fed with the values sent to the OP side. The fact that the predictors on OP and TOP side are strictly coherent (neglecting the unavoidable delay between the models because of transmission delays) enables us to only send packets over the network if the current actual signal differs from the predicted signal by more than the deadband/deadzone. A similar prediction is performed for the velocity values which are transmitted in the opposite direction.

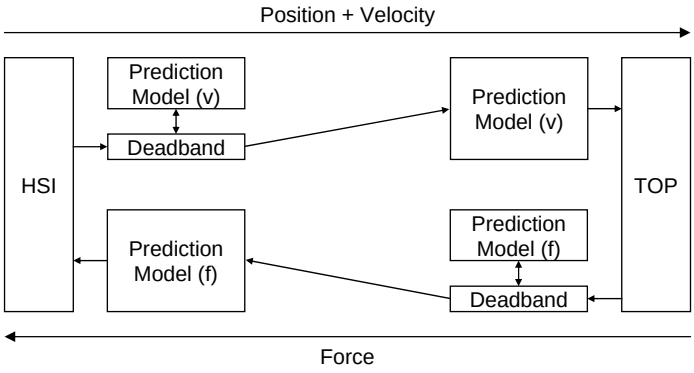

Figure 2.5: System with model-based prediction for higher packet rate reduction.

As an example of such a real time signal prediction a relatively simple linear predictor is implemented and experimentally analyzed in this work.

$$\mathbf{x}_i = \begin{cases} \mathbf{x}_{new} & \text{value sent/arrived} \\ \mathbf{x}_{i-1} + \frac{\mathbf{x}_{new-1} - \mathbf{x}_{new-2}}{t_{new-1} - t_{new-2}}(t_i - t_{i-1}) & \text{else} \end{cases} \tag{2.8}$$

where $\{\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{x}_{i-2}, \ldots\}$ are the most current values output by the model and $\{t_i, t_{i-1}, t_{i-2}, \ldots\}$ are the corresponding time instances. $\{\mathbf{x}_{new}, \mathbf{x}_{new-1}, \mathbf{x}_{new-2}, \ldots\}$ and $\{t_{new}, t_{new-1}, t_{new-2}, \ldots\}$ are the last sent/received values and the corresponding time instances.

With this predictor the signal is estimated by following the slope given by the last two received signal values. Once the predicted signal differs too much from the actual signal,

a new correct value is transmitted and the new prediction starts from there. More specifically, if the actual value falls outside the psychophysically motivated deadband around the predicted value we consider the prediction error to be noticeable and correct it by sending the actual value.

The control loops which secure the safe operation of both OP and TOP are normally updated at a fixed sampling rate. Having a prediction model as shown above enables us to use almost arbitrary sampling rates for those control loops. A strict match between OP and TOP sampling rates is no longer necessary, because the prediction model can be evaluated at any sampling rate and is updated as soon as it differs from the desired values.

The DBDR approach combined with model based prediction will be abbreviated by DBDR-P in the remainder of this work.

2.4.2 Filtering of Input Signals

Polynomial extrapolation, here a first order extrapolation, is known to be sensitive to high-frequency disturbances. Any high frequency sensor noise on the input signal results in large prediction errors and as a result in an unnecessary high utilization of the communication link. In real TPTA systems, especially the velocity signal is very noisy as it is not measured directly but derived from a discrete time difference approximation of the measured quantized position signal. Noise naturally also occurs in force measuring but does not have the spiky behavior of the velocity signal. This is one of the reasons why the compression of force signals works significantly better than the compression of velocity signals for the approaches described in Sections 2.3 and 2.4.1 as will be later demonstrated in our experimental results.

In order to be able to predict future samples more accurately and therefore increase compression performance we denoise the input signal using a fast Kalman filter. A snapshot of a velocity signal in one of our experiments with and without filtering can be seen in Figure 2.6.

How the Kalman filter is used in our system is shown in Figure 2.7. To be computationally efficient and to cause as little delay as possible we need a filter which has a low complexity and can be applied to a signal sample by sample. The Kalman filter [46] in a simplified form fulfills those needs. We use a scalar Kalman filter for every DoF separately to keep computational complexity low.

A discrete one dimensional signal of sample values x_k (the real signal without noise) is approximated by an estimated signal $\hat{x}_k$. Therefore measurements z_k are taken every sampling instant and the following algorithm is applied:

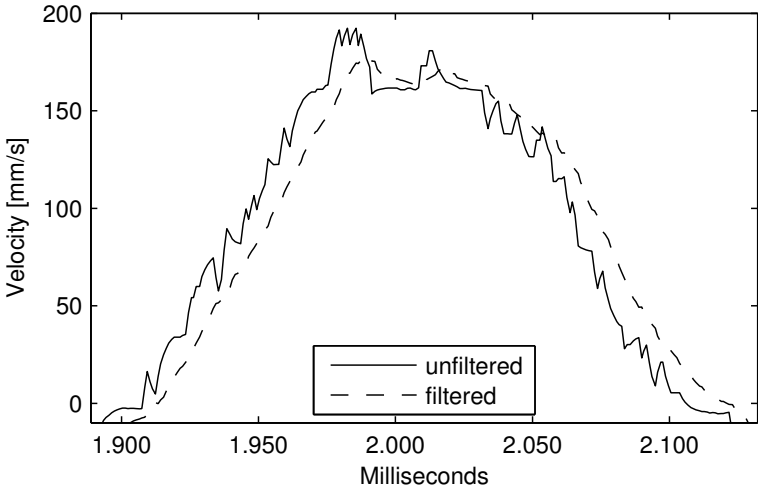

Figure 2.6: Velocity signal before and after pre-filtering.

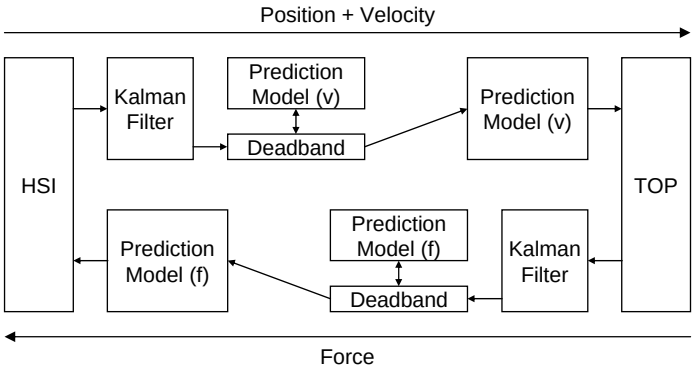

Figure 2.7: Application of the Kalman filter in the system.

1. Calculation of the innovation I, which is the difference between the measurement and the estimation:

$$I = z_k - \hat{x}_{k-1} \tag{2.9}$$

2. Calculation of the variance of the innovation I:

$$S = P + R \tag{2.10}$$

where P is the variance of the prediction error (calculated below) and R, the variance of the measurement noise, which is also the only parameter we use to adjust the filter characteristics.

3. After that we calculate the gain K for the estimation step:

$$K = \frac{P}{S} \tag{2.11}$$

4. Then finally the estimation becomes:

$$\hat{x}_k = \hat{x}_{k-1} + K \cdot I \tag{2.12}$$

5. Calculation of the new variance of the prediction error:

$$P_{new} = P_{old} + Q - K \cdot P_{old} \tag{2.13}$$

where Q is the variance of the process noise. We set it to 1 to have only one filter parameter.

Steps 1 through 5 are executed every sampling instant and the filtered signal values $\hat{x}_k$ are taken as input signals instead of the noisy measurements.

The filter characteristics can be adjusted by changing the parameter R, the variance of the measurement noise, which weighs the measured values against the predicted value which in our filter is simply holding the last value. A low R leads to a high confidence that measurements are correct, therefore measurements are weighted more for the filter output than the prediction and vice versa. The example in Figure 2.6 was recorded using $R = 100$, which was chosen as a good tradeoff between noise reduction and system response during the preparation phase of the experiment. This was also the value which was used for the velocity signal in the psychophysical experiment in Section 2.5.2.3.

Because the main reason of filtering the input signals is to get rid of signal noise, other kinds of filters like standard low-pass filters can also be used. The only important things are that the filter delay is kept at a minimum and the high frequency noise is suppressed effectively. The Kalman filter behaves very well in both properties and additionally performs best when gaussian noise is encountered.

The DBDR-P approach combined with Kalman filter based preprocessing of the input signals will be abbreviated by DBDR-PF in the remainder of this work.

2.5 Psychophysical Evaluation

Several experiments have been conducted in order to verify the proposed data reduction techniques. A 1-DoF experiment is used to determine the detection threshold of the dead-band parameter p and to relate it to the JND k. In the 3-DoF experiments the user utility of the proposed 3-DoF approaches is determined.

2.5.1 1-DoF Approach

The main challenge of the approaches presented in the last sections is to minimize the network traffic while maintaining maximum possible immersiveness. As a first step psychophysical experiments were conducted for the 1-DoF case in order to determine the maximum value of the deadband parameter p where a degradation of the immersiveness is not perceivable. Furthermore, the effect on the network traffic is studied.

2.5.1.1 Setup

The experimental setup consists of two identical 1-DOF haptic devices connected to a PC and a stiff wall as the environment (see Figure 2.8). The angle is measured by an incremental encoder, the force by a strain gauge. The sensor data is processed in the PC where all control algorithms including the DBDR algorithm are implemented. The velocity/position is transmitted to the TOP acting as the set value for the local control loop of the TOP. The TOP tracks the movement of the HSI and communicates the measured contact force back to the HSI as the set value for the force control loop.

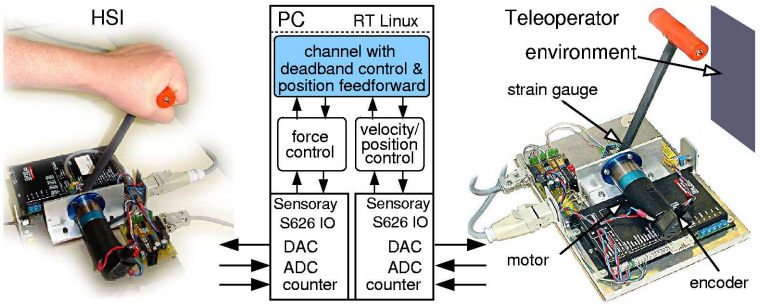

Figure 2.8: Experimental setup with two 1-DOF haptic devices [1].

2.5.1.2 Procedure

Altogether 14 subjects (3 female, 11 male, aged 20–50) were tested for their detection threshold of the deadband parameter p. The subjects sat in front of the HSI lever and were told to operate it with their preferred hand. They were equipped with earphones to mask the sound the device motors generate. During a familiarization phase the subjects were told to feel the hard contact, a stiff wall by which the lever movement was restricted at the TOP side, through the system at a sampling rate of 1000Hz and without any DBDR algorithm applied. As soon as they felt familiar with the system the measurement phase began. The deadband parameter detection thresholds were determined using a three interval forced

choice (3IFC) paradigm. The subjects were presented with three consecutive 20s intervals in which they should operate the system. Only in one of the three intervals, which was randomly selected, the DBDR algorithm with a certain value p was applied. The other two were presented without DBDR. Every three intervals the subject had to tell which of the three felt different from the other two. The experiment started with a deadband parameter $p = 2.5\%$ and was increased after every incorrect answer up to a maximum of 25%. When an answer was correct, the same value was used again until three consecutive right answers were given. After this first pass, the subjects were told what the distortion feels like and with what kind of technique they should be able to perceive it best. Then the same procedure as before was applied (2nd pass). After another three consecutive right answers p was reduced by 50% without telling the subjects and the procedure was repeated in order to verify the detection threshold one more time (3rd pass). The mean value of the three p values at which the consecutive right answers occurred were taken as the deadband detection threshold for the specific subject.

2.5.1.3 Results

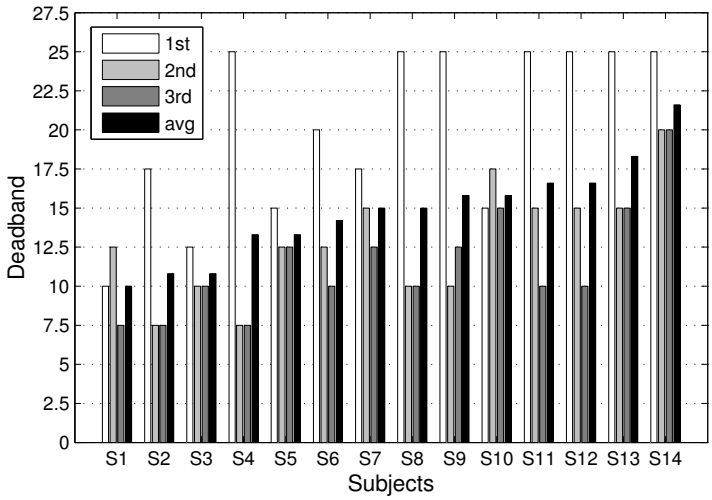

Figure 2.9: Subject-specific detection thresholds for p in the 1-DoF experiment.

The specific results for every subject are shown in Figure 2.9. Comparing the results of the three passes for the individual subject, all subjects had a significantly higher detection threshold in the first pass when they did not know what kind of distortion they had to expect. Hence, the distortion introduced by the DBDR approach is not necessarily perceived as disturbing or impairing the contact impression. The subject specific detection thresholds are in the range between 10% and 22.5%. Only one subject managed to detect the distortion introduced by $p = 10\%$. For the remaining 13 subjects corresponding to 93% a higher

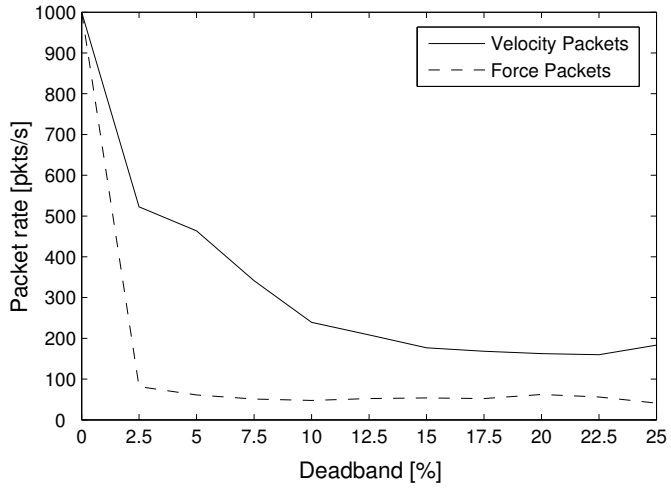

Figure 2.10: Influence of the deadband width on the packet rate: Average number of transmitted packets as a function of the deadband parameter p.

threshold was determined. Eleven subjects (79%) had a detection threshold $p > 11\%$. The measured detection thresholds are in the range of the JNDs reported for velocity and force perception [22, 44]. It should be noted, however, that JNDs are typically determined in static conditions. Here a temporal change of the signal is considered. The relation between the JNDs obtained by psychophysical experiments for static conditions and our deadband results for dynamic manipulation conditions requires further investigation.

In order to investigate the effect of DBDR on the packet rate the induced network traffic was recorded during the experimental user study. The mean percentage of transmitted packets as a function of the deadband parameter p is shown in Figure 2.10. 100% represent the standard approach with 1000 packets/s on the forward and the backward path, respectively. As expected, higher deadband parameters lead to higher traffic reduction. The traffic volume induced by velocity packets is already at 25% at a deadband size of $p = 10\%$ and keeps falling with increasing deadband size. The impact on the number of force packets transmitted is even higher. Already at $p = 2.5\%$ the network traffic volume in the backward path is less than 10% of the standard approach. At $p = 10\%$ only 15% of the original number of packets is transmitted. This means an average network traffic reduction by 85%. 93% of the subjects were not able to feel the distortion introduced by the corresponding deadband parameter.

2.5.2 3-DoF Approaches in Virtual Environments

In order to verify the presented 3-DoF DBDR approaches (3-DoF DBDR, 3-DoF DBDR-P, and 3-DoF DBDR-PF), several experiments were conducted using a commercially available 3-DoF haptic device. In contrast to the previous experiment, not a single detection threshold is determined, but the quality of immersiveness is rated over a range of deadband values as a first step towards a user utility function. The influence of the deadband is further investigated separately for the force and the velocity data.

Figure 2.11: The SensAble Phantom Omni device used for the experiments [www.sensable.com].

The conducted experiment for the 3-DoF DBDR approaches was a haptic interaction task with a virtual remote environment. The hardware and software setup is as follows: On the OP side the haptic display device SensAble PHANTOM Omni (see Figure 2.11) serves as the HSI. Over a 100Mbit/s Ethernet LAN connection this OP side transmits current position and velocity samples to a simulated haptic environment on another machine in the same LAN.

2.5.2.1 Setup of the OP Side

Haptic Display Device

The haptic device is capable of 6-DoF input and 3-DoF output. This means that both the endeffector's position in space as well as its orientation can be read from the device drivers. In contrast to that it is only possible to output forces in 3-DoFs namely the three directions in space. The torques necessary for altering the endeffector's orientation cannot

be produced. In our experiment, the additional 3-DoFs of endeffector orientation are only used to display the 3D-cursor of the graphical display correctly. They are neither sent to the TOP side nor do they have any other influence.

Graphical Display

The graphical display consists of an OpenGL-based 3D visualization of the workspace. Both the current cursor position and the position of the haptically manipulated object are displayed. See Figure 2.12 for an impression of the HSI graphical display.

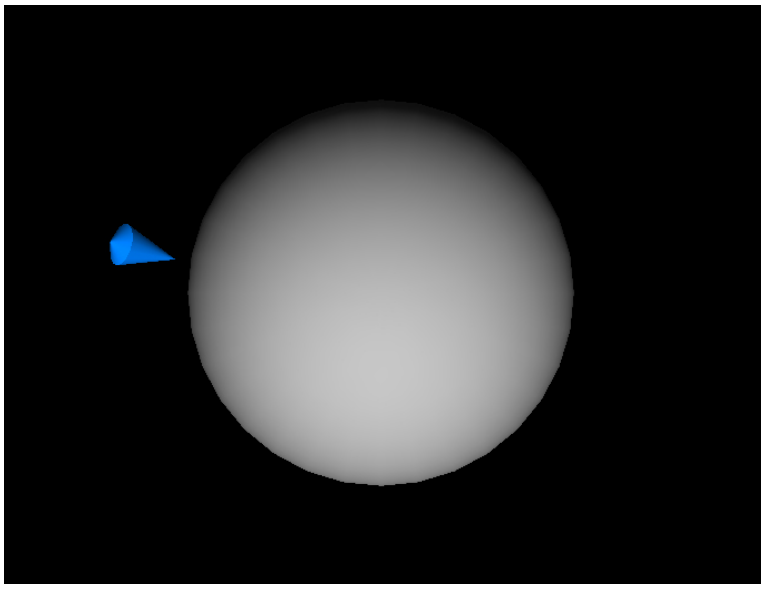

Figure 2.12: The graphical display of the OP side (HSI).

This display shows a grey sphere in the middle of the workspace of the haptic display device along with the blue cursor which signifies the current position and orientation of the device.

The pose of the haptic display device is sampled at 1000Hz. The graphical display is refreshed with the standard refresh rate of 60Hz.

Deadzone Implementation

The three components of the current device velocity are combined in a 3D vector. According to Equations 2.2 in Section 2.3 the size of the deadzone is calculated. The reference (or initial) vector is always the one which was last transmitted to the receiver. Every 1ms a new value of the 3D velocity vector is read from the device drivers and it is decided whether its tip lies in the deadzone or not. According to the result of the decision, a new

vector is sent to the receiver or the new vector is discarded. In case DBDR-P is used, a respective prediction step is introduced and the decision is based on the prediction error (see Section 2.4.1). If DBDR-PF is used an additional Kalman-prefiltering of the velocity signal taken from the device is done (see Section 2.4.2).

Position Update
In such a deadband system the sending of velocity values only would result in a more or less severe degradation of position tracking. It is therefore necessary to send the actual position values along with the current velocity values so that the TOP can take care of position errors. Since the packet sizes are very small this additional amount of data is negligible. The packet rate is not increased.

2.5.2.2 Setup of TOP Side

Virtual Environment
The virtual environment is implemented by a C++-Class which manages the positions and properties of virtual objects which are to be manipulated as well as the positions of one or more users interacting with the environment. It is capable of 3-DoF input and 3-DoF output. This means it is fed with a 3-DoF velocity input (along with a 3-DoF position input for reasons of position tracking) and calculates the resulting forces for this position. The environment as well as the haptic display device at the OP side are refreshed at a rate of 1000Hz. Unlike the systems in [36] and [43] which also implement a virtual haptic environment it is intentional that all computations concerning the haptic feedback are done centralized on one machine like in [23]. This central approach is chosen in order to have a system which is as comparable as possible to a real TPTA system. Also different from [26] where only one packet is in transit at all times for stability reasons, the presented system communicates in both directions at the same time. Stability problems were not observed.

Sphere Object
The only object in the virtual environment in this experiment is a sphere in the middle of the virtual workspace. This sphere is registered with the sphere in the graphical display (see Section 2.5.2.1) so that contacts between the cursor and the sphere in the graphical display exactly correspond with contacts in the virtual environment. The virtual sphere is fixed at the center of the workspace and can be touched with the virtual cursor. The resulting force during the interaction is calculated by Hooke's Law

$$F = u \cdot b \tag{2.14}$$

where u is the stiffness of the sphere and b the amount of penetration into the sphere body. F is the resulting force magnitude. The direction of the force always points from the sphere center to the actual cursor position. It is calculated as

$$\mathbf{F} = \frac{\mathbf{x} - \mathbf{s}}{|\mathbf{x} - \mathbf{s}|} \cdot (r - |\mathbf{x} - \mathbf{s}|) \cdot u \tag{2.15}$$

where the resulting force vector $\mathbf{F}$ is determined from the current position of the user $\mathbf{x}$, the sphere position in space $\mathbf{s}$, the sphere radius r, and the stiffness u.

Deadzone Implementation
The initial vector for the deadzone calculation is the force vector which was last sent to the OP side. Every time the virtual haptic model is updated it either sets the most current position and velocity values for the user position (in case an update packet has arrived) or calculates a new position from the last known user position, the last known user velocity, and the exact time since the last update. This updated position is then used to calculate an updated force which then is used as the current vector for the deadzone calculations. In case the deadzone is violated by the new vector, a new packet containing the updated force vector is sent and the sent vector serves as the new initial vector. In case DBDR-P is applied an additional prediction step for the reference force is introduced before the deadzone decision. DBDR-PF additionally introduces prefiltering of the actual force signal taken from the virtual environment model.

2.5.2.3 Subjective Evaluation

Ten test subjects underwent the experimental procedure described in the following to determine suitable values for the deadband parameters so that no degradation of immersiveness can be noticed.

The following cases were considered:

- 3-DoF DBDR on velocity values only as described in Section 2.3
- 3-DoF DBDR on force values only as described in Section 2.3
- 3-DoF DBDR-P on velocity as described in Section 2.4.1
- 3-DoF DBDR-P on force as described in Section 2.4.1
- 3-DoF DBDR-PF on velocity as described in Section 2.4.2
- 3-DoF DBDR-PF on force as described in Section 2.4.2

The subjects are first presented with a system without DBDR to get used to handling the device and to experience what it feels like. Then a heavily distorted system is shown to the subjects in which they can clearly feel the kind of distortion which is introduced into the system by the DBDR algorithm with very high deadband parameters p. This phase is called the familiarization phase.

After the subjects feel familiar with the system and they know the kind of distortion they are presented with, two test runs are conducted each consisting of twelve 30-second intervals (24 intervals in 12 minutes total) in which the subjects were told to haptically explore the virtual environment and to assess the quality of the haptic presentation. In the first run with twelve intervals the DBDR approach is only used for the velocity values which are sent from the OP to the TOP. In the second run the deadband is only used on force values which are sent from the TOP to the OP. During the tests, the subjects wore headphones so they could concentrate on their haptic sensations.

In the 12 intervals of each run we apply a randomly chosen order of the following possible deadband values: 0%, 2.5%, 5%, 7.5%, 10%, 12.5%, 15%, 20%, 25%, 30%, 35%, and 40%. The subjects did not know either which value was currently used or in which communication direction the DBDR was applied.

After every interval the subject was required to rate the presentation. If it felt exactly like the undistorted signal from the familiarization phase, they should give a rating of 10 points. If it felt just as bad as the heavily distorted signal from the familiarization phase, they should give a rating of 1 point. The ratings in between could be chosen according to the quality of the presentation where higher ratings signify better quality.

2.6 Performance

The results for the mentioned 3-DoF approaches can be seen in Figures 2.13 and 2.14.

2.6.1 3-DoF without prediction and filtering

The results are represented by the solid lines in Figures 2.13 and 2.14.

From Figure 2.13 it can be observed that a deadzone usage on velocity values seems to be far less perceptible than on force values. One can see that the velocity deadband can be increased to up to 20% while still reaching an average rating of almost 7 points, which most subjects described as barely perceivable distortion. In comparison, the force deadband should not be far above 5% for the average rating to also stay above 7 points.

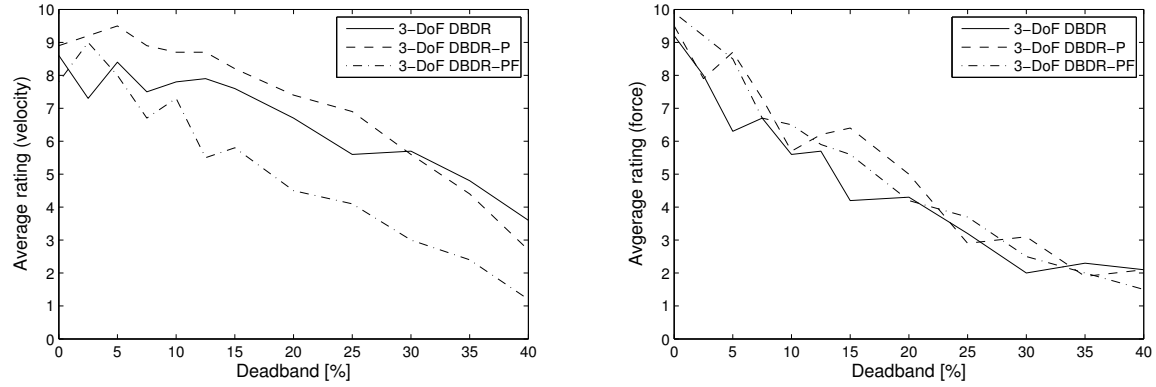

Figure 2.13: User ratings for the deadband presentations in the 3-DoF approaches.

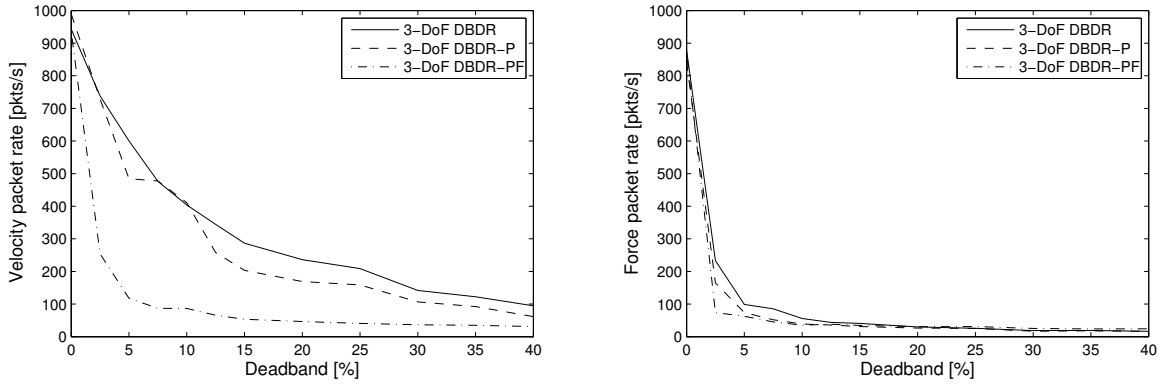

Figure 2.14: Resulting packet rates of the 3-DoF approaches.

This behavior has two reasons. The first reason lies in the fact that the used TOP is a VR environment. Position errors can be corrected by just setting the actually transmitted position value as the current position. In a real TPTA system it is more difficult to correct this position error because the endeffector has to be moved towards the correct position. This introduces additional distortion whereas in VR environments the position can be updated instantaneously.

The second reason lies in the DBDR principle itself. As we have mentioned in Section 2.2, a new value is only transmitted if the user can sense the introduced change. This is of course true for the direction from the TOP to the OP. The transmitted forces are directly sensed by the human being. In contrast to this in the other direction no human sensory system is involved. So there is basically no reason to transmit new velocity values adapted to human perception. Velocity updates are processed by a non-human system which uses it merely to generate new force values for the HSI. Therefore environment dynamics affect the possible degree of deadband application. Consequently we can see that it is often possible

(at least in the presented case, which somehow stands for most VR haptic environments) to use deadband transmission far beyond human haptic sensory capabilities for the direction from OP to TOP.

With respect to the resulting packet rates the first observation is that, in general, more velocity packets than force packets are generated as can be seen in Figure 2.14. This also has multiple reasons. First, velocity packets have to be sent all the time for tracking the endeffector whereas force packets only need to be sent when contact with the environment takes place. Secondly, force usually reaches higher magnitudes (and therefore higher deadbands) more quickly than velocities. In the case of this experiment the test subjects are in contact with the environment almost all the time, and so velocity is mostly small whereas force is quite high in most cases. The third reason lies in the differentiation of already noisy position values in order to get the desired velocity signal. This differentiation amplifies the noise and this high amount of noise is therefore another reason for triggering the deadband, especially when it is small in magnitude.

It can be observed that with 0% deadband less than 1000 packets per second are sent. This comes from the fact that even with 0% deadband a change in the measured variable must occur to trigger a new packet transmission. In the case of calculated forces of the VR environment, force is exactly zero while no contact to the environment is made. Therefore only in case of contact packets are sent. Knowing that, we can conclude that during the experiments with 0% deadband the subjects had contact with the environment about 87% of the time.

In comparison to our results for the 1-DoF case in Section 2.5.1.3 we can conclude that the deadband usage in three dimensions leads to similar tendencies in packet rate reduction as the 1-DoF approach but is not quite as effective. Packet rates for velocity packets are reduced by almost 75% when using a barely perceivable 20% deadband. For force packets a reduction by almost 90% is possible by chosing the also barely perceivable 5% deadband.

2.6.2 3-DoF with linear prediction

The dashed lines in Figures 2.13 and 2.14 show the corresponding results.

The ratings in Figure 2.13 given by the test subjects are almost always above those of the previously mentioned case. Hence it is possible to use even larger deadbands when prediction is applied in this case.

We can see that even the simple linear prediction model in 2.4.1 reduces the packet rates in comparison to the 3-DoF approach in the previous paragraph. In Figure 2.14 we observe an improvement of 28% and 25% for a 20% velocity deadband and a 5% force deadband,

respectively. The total savings in comparison to transmission without deadband are 83% and 93% for velocity and force packets, respectively.

One possibility to further improve this approach is to use more sophisticated prediction methods. However, the limiting factor is signal noise as the following experiment shows.

2.6.3 3-DoF with linear prediction and filtered input signals

The results are represented by the dash-dotted lines in Figures 2.13 and 2.14.

In Figure 2.13 we can see an almost linear decrease in the ratings for both types of data with increasing deadband value. We assume that values of 7 and higher represent a good feeling of immersion in the system. Consequently, we can say that 10% deadband for both velocity and force should not be exceeded so as not to sacrifice immersion.

With respect to packet rates, compared to the results from the previous paragraph where no pre-filtering was applied, we can observe a drastic decrease in velocity packet rates, especially for small deadband values. This is exactly the benefit the pre-filtering was supposed to give. This has two main reasons. First, during motion phases with small velocities the velocity noise triggered unnecessary packet transmissions. With reduced noise in the signal this happens considerably less often. Secondly, less noise makes it easier to estimate and predict signal slopes.

For force packet rates the improvements are not as significant as in the velocity case. The reason for this is the fact that we have considerably lower noise levels on the force signal to begin with. The pre-filtering step is therefore not as efficient here as for velocity signals. Still we can observe a significant improvement (55%) in packet rate at 2.5% deadband in comparison to the LP case. This means almost 93% packet rate reduction in comparison to the original rate with only a minimal 2.5% deadband applied.

Finally we can state that the proposed pre-filtering step for prediction based deadband transmission of 3D haptic data works well for velocity and force data. At a combination of 7.5% deadband for velocity and 2.5% deadband for force we achieve a reduction of packet rate to 8.7% of the original rate for velocity and 7.4% for force with barely noticeable influence on immersiveness.

2.7 Chapter Summary

In this chapter first steps in the compression of haptic media data and the main contribution of this thesis, the DBDR approach with its extensions, are presented and discussed.

Intensive experimental uses studies were conducted to prove the effectiveness of the DBDR approaches.

It is very likely that the presented results can be transferred to more complex scenarios and tasks. The operation of more complex or more dynamic scenes should not be very different. The results from [44, 22] are for static conditions. However, all our results point in the direction that thresholds similar to the static JND are valid for the dynamic case. To the best of our knowledge, research in psychophysics has not yet considered spatio-temporal behavior of the JND in literature.

Perceptual thresholds in combination with the resulting packet rates allow us to choose optimal trade offs. For force samples it is generally not necessary to choose thresholds higher than a few percent because packet rates are already very low at this point. For velocity it is highly dependent on the amount of noise in the signal, but generally we can say that 10 to 20 percent should be possible in most cases.

With increasing system complexity in terms of the number of DoFs used for interaction, the approaches become less efficient. This is because only three directional or angular DoFs can be combined in a reasonable way. If more DoFs are used they have to be grouped and every group may trigger a packet transmission at every sampling instant. Finding reasonable combinations of more than 3 DoFs in the case that data generation is synchronized over these DoFs is subject to future research.

Chapter 3

Influence of Deadband Compression on Task Performance in a real TPTA System

The previously described approaches for multi-DoF haptic data reduction were so far only employed in virtual environments. The obvious next step is to examine their viability in real TPTA systems, where both HSI and TOP are real robots with all necessary sensors and actuators and control systems to keep each side and the whole system stable. Therefore, an experiment was conducted in order to study the influence of the DBDR and DBDR-P schemes on the performance of such a real TPTA system.

3.1 Hardware Setup

The hardware setup of the performed experiments consists of the DeKiFed (HSI) and DeKiTop (TOP) devices described in [47]. The 4-DoF design of the described system was truncated to a pure 3-DoF architecture with purely translational actuation and sensory. The SCARA type arms use DC-motors with Harmonic Drives in the joints to provide high-performance and low-friction tracking and force feedback. Both the HSI and the TOP are equipped with a force/torque sensor at the endeffector (TOP) and the handle (HSI) making it possible to exactly track the forces encountered with the environment. At the TOP these are the contact forces with the remote environment, at the HSI the actual force feedback to the human operator is measured.

The control systems are implemented in MATLAB/SimuLink and compiled for the real-

time capable Linux derivative RT-Linux. Separate PCs with reasonably dimensioned hardware and RT-Linux as the operating system are running the control applications for the HSI and TOP, respectively. These PCs are part of the 100MBit/s institute Ethernet network, which was used for UDP/IP communication. The robot hardware itself is connected to the PCs using special I/O-PCI cards.

3.1.1 HSI Side

The HSI of the system is also called master side in the following. The HSI, as shown in Figure 3.1, is a 3-DoF translational input device with four active joints of which only three were used in the experiment. The two joints with vertical rotation axis are used for sensing and actuating the x-y-plane of the workspace. The joint with horizontal rotation axis in combination with a parallelogram mechanism senses and actuates the z-direction. The handle is mounted on a force/torque sensor and the unused joint on the off-joint side of the parallelogram mechanism. A power grasp is used to hold the handle and to control the position of the HSI and, through the TPTA system, the TOP. The original setup of the HSI is bimanual but only the right hand side is used in the presented experiment.

3.1.2 TOP Side

The TOP of the system is also called slave side in the following. Since the HSI and the TOP have mainly identical configuration, the HSI description fits also for the TOP at least concerning the x-y-actuation. The z-direction has an inverse parallelogram configuration to the HSI, where the unused joint and the force/torque sensor are mounted on the bottom of the parallelogram mechanism. Comparing Figures 3.1 and 3.2 quickly reveals the similarities and differences. As the actual endeffector a steel stick with a wooden tip is used which is mounted directly into the force/torque sensor.

3.1.3 Video Transmission

In order to allow for visual feedback of the remote environment a camera is mounted at one specific angle which allows for a good view of the whole interaction scene as shown in Figure 3.3. The camera resolution is 640x480 pixels with 30 frames per second. The video stream was encoded using an MPEG4 codec and is transmitted via the Ethernet network to another PC where the picture is decoded and displayed to the test subjects.

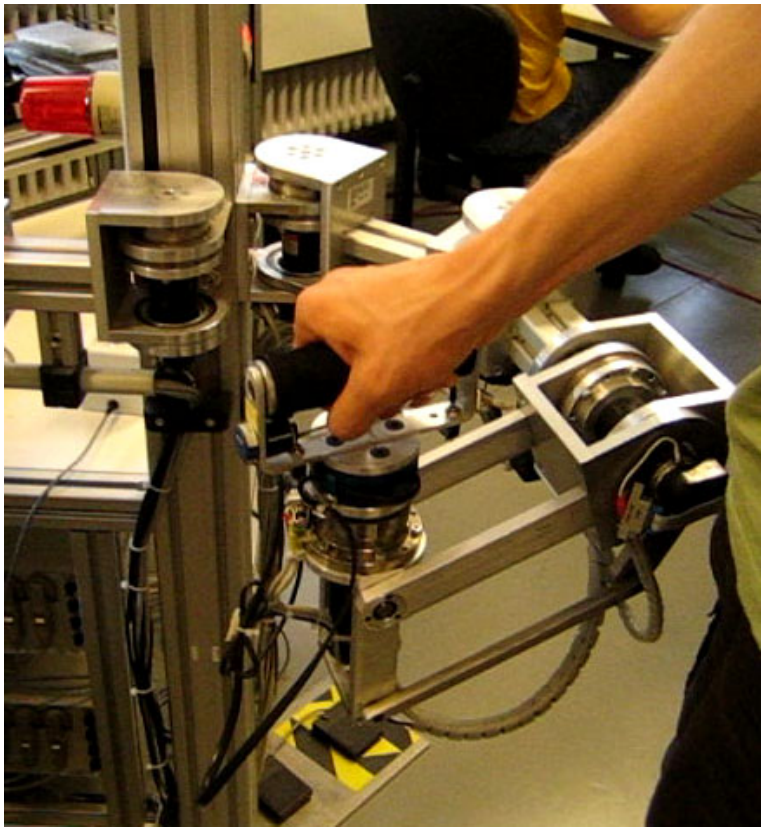

Figure 3.1: The Human System Interface of the TPTA system.

The video presentation itself was intentionally held as simplistic as possible. The test subjects should mainly rely on their haptic feedback of the remote environment instead of heavily relying on the visual feedback.

3.2 Control Scheme and Stability Measures

Since the experiment was conducted without delay on the communication channel in order to evaluate the influence of the DBDR schemes on task performance alone, no special control measures were taken to compensate for transmission delays. The system itself uses an update rate of 1000Hz in the HSI and TOP control loops and to communicate current sensor data between HSI and TOP, thus generating 1000 packets per second in each direction of the network connection in the uncompressed state.

First tests during the preparation phase of the experiment used velocity control at the TOP side to follow the users trajectory at the HSI. This of course works well if no compression takes place. However, if DBDR schemes are in place, this leads to a position drift between

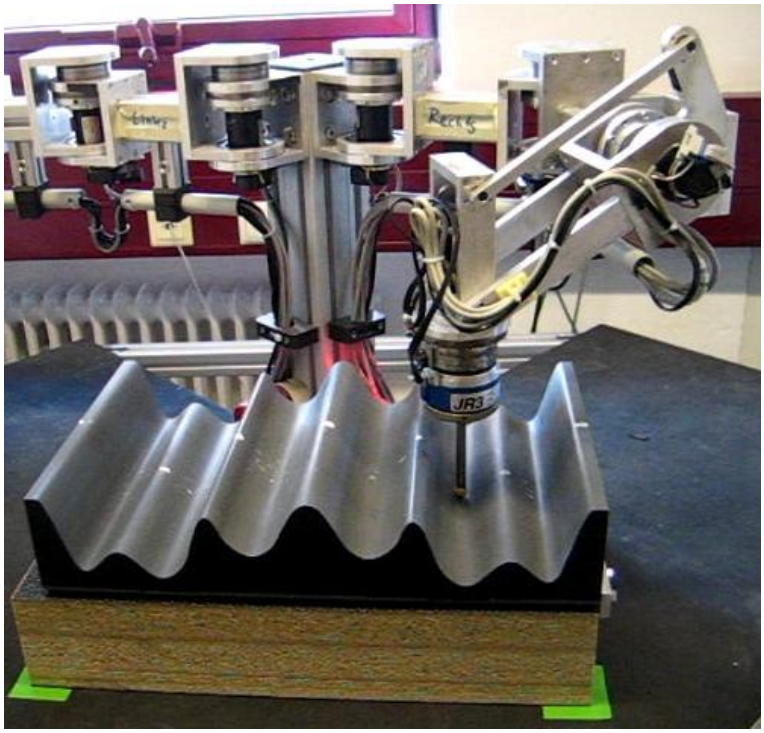

Figure 3.2: The Teleoperator of the TPTA system.

HSI and TOP which increases with longer execution times and increasing deadband size. In consequence the control scheme on the TOP side was changed to position control. By sending both the current velocity and the position information from the HSI to the TOP with every update packet, the current position is always known. In times when no new update packet arrives, the current position is predicted from the most recently transmitted velocity and position vectors and the time since the last update by using the following linear extrapolation:

$$p_{n+1} = p_n + v_c t_s \tag{3.1}$$

where p_{n+1} is the new position in the current sampling step, p_n is the position in the previous sampling step, v_c is the most recent velocity value received from the HSI and t_s is the sampling time. Since this approach basically predicts the current position of the HSI, the unavoidable prediction error has to be corrected when the next packet arrives. Then the actual position of the HSI is known again and the TOP has to be positioned accordingly. This can lead to rapid TOP movements if the prediction error becomes too large. Since this only occurred for quite high deadband values and worked flawlessly for practical deadband values, this approach was used in the experiment.

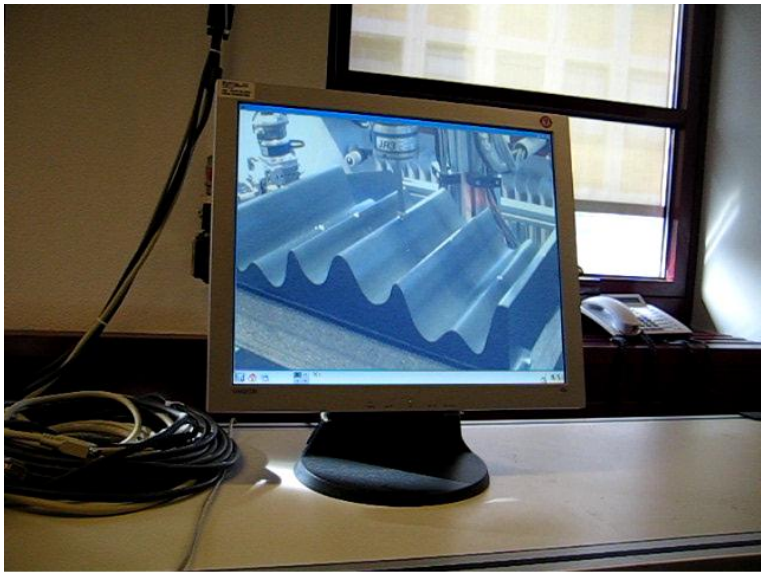

Figure 3.3: The video display at the HSI.

At the HSI force control was used to present the user with the transmitted force values of the TOP side environment interaction. Since the quite simple combination of position control on the TOP side and force control at the HSI tends to become unstable when hard contacts are encountered, a force gain of 0.5 was introduced in the system as a stabilization measure. This means that the human operator only feels half the force the sensors on the TOP side record. This kind of force scaling does not overly influence the perception of the remote environment but greatly helps with stability. Since the manipulated object was very hard and stiff, this measure was used in the experiment.

3.3 Data Reduction Scheme

In order to implement the data reduction scheme into the system, a special MATLAB/SimuLink Block had to be developed, which supports the transparent data reduction using the DBDR and DBDR-P schemes. Since the TPTA system allows for telemanipulation in 3 degrees of freedom, the 3-DoF DBDR approach with isotropic spherical deadzone was used as described in Section 2.3. The evaluation was done for deadband values of 0%, 2.5%, 5%, 7.5%, 10%, 12.5%, and 15%. Additionally, the prediction approach as described in Section 2.4.1 was used in combination with all deadband values, except for 0% where prediction is unnecessary, resulting in a total of 13 parameter sets. A Kalman filtering of input signals was not necessary in this setup, since the control schemes themselves already implement low-pass filtering on the velocity signal.

In the direction from the HSI to the TOP packets consisting of 3-DoF position and 3-DoF velocity samples are sent. These values are quantized with 16bit resolution resulting in a payload size of 12bytes per packet. The transmission of update packets is purely based on the velocity signal by using the 3-DoF DBDR approach on it. The position information is only sent due to necessary position correction at the TOP side as explained above. The model based prediction approach was also only used on the velocity signal. Since the payload size is very small compared to the header size of 30bytes (20bytes IP, 8bytes UDP, and 2bytes application protocol), the additional position information is negligible especially if taken into account that the DBDR approach aims at packet rate reduction rather than total data rate reduction.

In the direction from TOP to HSI packets consist of 3-DoF force information, also sampled at 16bit resolution resulting in total payload size of 6bytes and also 30byte header information. No additional information (like the position information in the opposite direction) is necessary here because there are no drift effects for force feedback. The 3-DoF DBDR and DBDR-P approaches are used directly on those force values.

3.4 Task and Manipulated Environment

The manipulated object is a massive plastic profile which has the shape of a one dimensional waveform on the upper side. It is shown in Figure 3.2. The profile itself weighs about 3kg and is very rigid and stiff. It is aligned with the Cartesian axes of the TOP so that the x-axis of the TOP lies orthogonal to the wave profile, the y-axis lies parallel to the wave profile. In z-direction (the vertical axis), the profile is mounted on a wooden block, so that the whole depth of the profile could be reached by the endeffector without leaving the optimal workspace of the TOP. The profile itself is made of black plastic and its center along the x-axis is marked by white dots on the maxima of the waveform to give a guidance of the optimal manipulation route.

The task the test subjects had to perform during the experiment is described in the following. After the initialization of the HSI and the TOP, which leads to both of them being positioned at a fixed starting position, the subject has to move the TOP to either the right or the left edge of the profile as instructed by the conducting person and touch the profile at the white point in the middle of the edge. Starting from there, the subject has to follow the surface along the marked center line across all the waves until he/she reaches the other side of the profile. We call this the contact phase. After reaching this point the subjects are to return to the starting point on the other side of the profile in free space over the profile. We call this the free space phase. The subjects are instructed to try to keep contact to the profile surface for the whole time during the contact phase and to keep the

contact force as constant as possible. They are also told to perform the task as precisely and as quickly as possible.

The video feedback during the experiment was supposed to provide the necessary information about reaching the starting point and keeping the correct heading across the wave surface since the white markers were clearly visible although a real 3D impression was not provided because neither camera movements nor stereo vision was allowed. The intentionally limited video view of the remote environment forced the test subjects to rely their actions more on their haptic feedback through the TPTA system as it is normally necessary for interacting with their environment. This way, a disturbance of the haptic feedback should be detectable more easily, which was the intention of the experiment.

3.5 Experimental Procedure

Test subjects were first introduced into the experiment with a short summary about the procedure that awaited them. Their task was described as the evaluation of the quality of an interface for human-robot interaction. After that they were led into the actual experiment site which was shielded against external influences by three cardboard walls. Only the haptic HSI and the monitor displaying the video feed were visible to the subjects. Then the subjects were introduced to the operation of the haptic HSI. They were told that touching it is possible after it has reached it's initial position. Touching it before that would lead to system malfunction. After that the so called familiarization phase took place. The subjects were able to freely use the TPTA system to explore the remote environment without the influence of any compression technique in order to get comfortable with its operation. They were also shown what the procedure, which they were to complete in the upcoming tests, looked like and could try it out beforehand. Subjects were also told to try to complete the task as quickly and as precisely as possible and to try to stay in contact with the surface of the test object at all times.

The subjective quality of the human-robot interaction interface was assessed using Scheuchenpflugs "Fragebogen zu Präsenz und Immersiver Tendenz in virtuellen Realitäten" (Questionnaire on presence and immersive tendencies in virtual realities) [63], which is a renowned way of evaluating this kind of system from a psychologist's point of view. The subjects were told to fill in the first page of the questionnaire where questions about handedness, experience with human-robot interfaces, and experience with 3D computer games were asked as the next step of the experiment. The following pages had to be filled in after every pass in the experiment.

After that the subjects were told that they have to wear headphones with white noise from

now on to shield them from external auditory influences.

In each of the 13 passes each test subject had to complete, they first waited for the HSI to reach the initial position. Then they were shown a paper card indicating the side (left or right) from where the next pass of the experiment should start from. After trying to complete the task under the given circumstances of DBDR and DBDR-P as fast and as precise as possible, the subject filled in another page of the questionnaire, where three questions concerning the quality of presentation in the last pass were posed.

The 13 passes of the procedure with the parameters described above were used in totally randomized order. Each pass took between 30s and 90s to complete, depending on the speed the respective test subject felt comfortable with. If one of the above mentioned task goals were obviously not met by the participant, for example no contact with the edge at the beginning or the end of the test run, not reaching the other edge of the profile, or other obvious misunderstandings of the task took place, the test run was repeated. Any abnormal test situations, especially repeated test runs, were noted down in the experiment protocol to be able to explain possible problems during the data interpretation later in the process.

After the test subjects finished their 13 test runs, each one had to pass an evaluation of his/her motor-sensory coordination skills using the Motor-Sensory Coordination test of the Wiener Test System. This was done in order to be able to identify test subjects which have difficulty with the skill the given task requires or exceed the normal population's abilities by far and consequently to be able to exclude those subjects from the analysis.

The complete test procedure took 45 to 60 Minutes per test subject.

3.6 Test Subjects

32 test subjects were recruited from both the staff of the involved institutes and the general public to take part in the experiment. Their average age was 28.03 years with a standard deviation of 5.30 years. 9 of the subjects were female, 23 were male. 4 of the subjects were left handed, 28 were right handed. Due to a large number of incomplete or missing data sets, one subject had to be excluded from the analysis. The test for motor-sensory skill showed that none of the 31 remaining subjects showed noticeable deviations from the average population in this respect. 6 of the subjects said that they had much experience in the usage of robotic systems whereas the other 25 stated that they had little to no experience in that respect. All test subjects did not differ much in their experience with 3D computer games (average time per week: 0.81 hours with an standard deviation of 1.54 hours).

3.7 Experimental Goal

There are three main questions which were supposed to be answered by conducting this experiment.

1. Influence of the DBDR in 7 steps, each with and without model based prediction (DBDR-P), on the packet rates in the TPTA system and if an increase of the deadband parameter and switching on prediction always results in a significantly lower packet rates.

2. Influence of the deadband and prediction parameters on the perceived quality of the haptic human-robot interface. Quality assessment was done based on the questions of the aforementioned psychological questionnaire.

3. Influence of the DBDR and DBDR-P approaches on the task performance during the experiment. Task performance was assessed through task completion time, contact loss, contact ratio, contact force behavior and track deviation during the experiment. All necessary experimental data was recorded in millisecond accuracy.

The variables encountered in the experiment are classified into independent and dependent variables. Independent variables are varied according to the experimental setup and procedure in order to find dependencies between them and the dependent variables, which are measured or otherwise assessed during the experiment.

Independent variables:

- Deadband parameter (varied in 7 steps from 0% to 15%)
- Prediction parameter (varied in 2 steps, on and off)

Dependent variables:

- Network packet rate (measured)
- Subjective Quality of the Interface (assessed by questionnaire)
- Task completion time (measured)
- Number of contact losses (measured)
- Contact ratio (measured)
- Contact force behavior (measured)
- Position tracking accuracy (measured)

3.8 Statistical Methodology

3.8.1 ANOVA

The statistical analysis in the following is based on the assumption that the assessed dependent variables are normally distributed. Based on this assumption it is determined if certain dependencies between independent and dependent variables, so called effects, are statistically significant or not. Additionally, it is determined how much confidence we can place in a dependency by determining the significance level of an effect.

Since no dependencies between the different dependent variables were found, univariate variance analyses (ANalysis Of VArinace, ANOVA, see [49, 27, 40]) between the dependent and the independent variables are done.

In all cases ANOVA is performed it is determined if a significant main effect can be detected. If no significant main effect is detected, no further analysis of the different levels of independent variables is necessary. If this is not the case, i. e., a significant main effect is present, a more granular assessment of the different levels of independent variables is necessary to see which one caused the effect.

To determine both the main effect and the so called single contrasts between the different levels of the independent variables and the reference case a so called F-test is performed. It is part of the ANOVA and determines whether the ratio of the between-group-variability, the variability between test and reference cases, and the within-group-variability, the variability within the test cases, is significant or not. It is given in the notation

$$F(x, y) = v, p = w \tag{3.2}$$

where x is the amount of degrees of freedom of the between-group-variability, y is the amount of degrees of freedom of the within-group-variability, v is the so called F-value which directly leads to w by looking up v in the probability table of the F-distribution. w is the probability that the test cases and the reference case are significantly different. In our case we consider effects lower than $p = 0.05$ as significant as usual in psychophysical studies.

When many test cases are compared to one reference case (in our case all the test runs with deadband parameters of 2.5% and more are compared to the reference case with no deadband) using ANOVA, this is called “repeated measures ANOVA”.

3.8.2 Sphericity

When using repeated measures ANOVA as the analysis tool, the data has to be checked for sphericity first. Sphericity is a measure for the equality of the variances encountered in the different runs in a repeated measures ANOVA. Mauchly's Test [51], a well known test for sphericity, is used in our case to verify this requirement. In case the sphericity test fails, so called Greenhouse Geisser corrections [33, 28] are applied to be able to reach correct conclusions from the ANOVA. The amount of degrees of freedom changes in case such a correction is applied.

3.8.3 Friedman Variance Test

In case the underlying experimental data is not normally distributed, ANOVA does not work by definition. One solution in such a case is the Friedman Variance Test [29, 30] proposed by Milton Friedman which is used to handle such tests without the underlying assumption of normality.

3.8.4 Wilcoxon Signed-Rank Test

If paired comparisons between two test cases have to be made and the data of those test cases is not normally distributed, the Wilcoxon Signed-Rank Test [73] can be used to determine if the differences between the cases are significant or not.

3.9 Experimental Results

3.9.1 Experimental Data

Since the very detailed recording of all involved experimental data in millisecond accuracy led to about 350 Megabytes of result data per test subject, an automated approach to extract all interesting variables from the raw data was used. A MATLAB script was used to parse both the data from the master and the slave of one experimental run and to extract the following variables in a form that could be read into the SPSS statistics suite for further analysis:

dataset
: Name of the dataset directory.

deadband
Deadband parameter used for this dataset.

prediction
Whether prediction was used in this dataset or not.

testnum
Number of this test run in the 13 runs each subject had to complete.

direction
Whether the test run started from the right or the left side of the test object.

packetsmaster
Number of packets sent by the master side.

packetsmastertime
Number of milliseconds that it took to send the master packets.

packetsmasterratio
Average number of packets per second encountered at the master side.

packetsslave
Number of packets sent by the slave side.

packetsslavetime
Number of milliseconds that it took to send the slave packets.

packetsslaveratio
Average number of packets per second encountered at the slave side.

taskcompletiontime
Time in milliseconds from the first contact on the starting edge to the final contact at the same edge after crossing the plastic profile and moving the TOP back to the starting point.

tctpenalty
When some part of the task was not or erroneously fulfilled this is different from 0 (which means everything went OK). Possible errors were, firstly, that the subject did not return to the starting position after crossing the profile, and secondly, the target area at the final position was not reached correctly but the free-space movement was done.

contactdur
Number of milliseconds the contact phase with the object lasted while crossing it.

forcemean
The mean value of the measured contact force during the contact phase.

forcevar
The variance of the measured contact force during the contact phase.

trackdeviationymean
The mean value of the tracking deviation orthogonal to the movement direction during the contact phase.

trackdeviationyvar
The variance of the tracking deviation orthogonal to the movement direction during the contact phase.

numlostcontact
Number of contact losses during the contact phase which lasted longer than 50ms.

contactdurratio
Ratio of the duration of the contact phase and the total duration of contact losses.

In the following the measured influence of the independent variables on the dependent variables is presented. For every dependent variable two plots show the effect of the DBDR approach and the DBDR-P approach on the respective variable. The plots are followed by the respective statistical analysis for the two cases and after that a brief interpretation is given if possible.

3.9.2 Achieved Packet Rate Reduction

Obviously, the most important goal of the DBDR approaches in the conducted experiment is the packet rate reduction performance. If the approach is to be employed, it is always necessary to weigh the packet rate reduction performance against the encountered quality deterioration. One major purpose of the presented experiment is to give examples of which performance parameters deteriorate with increasing data compression and which do not.

First we take a look at the encountered packet rates for the different compression parameters on the master and the slave side. Master packets carry position and velocity information while slave packets carry force information. Figures 3.4 and 3.5 show the mean packet rates and their standard deviations.

Since testing the packet rate data sets for normal distribution yielded negative results, Friedman variance tests were used instead of the standard parametric ANOVA. It was first examined if the encountered packet rates varied significantly with the compression

parameters. If a significance was detected, Wilcoxon Signed-Rank Tests were conducted to do pairwise comparisons between each deadband parameter step and their respective predecessors (e.g. 2.5% compared to 0%, 5% compared to 2.5%, and so on). After that, additional pairwise comparisons were done between each deadband parameter without prediction and with prediction in order to determine when the application of the prediction algorithm makes sense in general.

3.9.2.1 Master Side

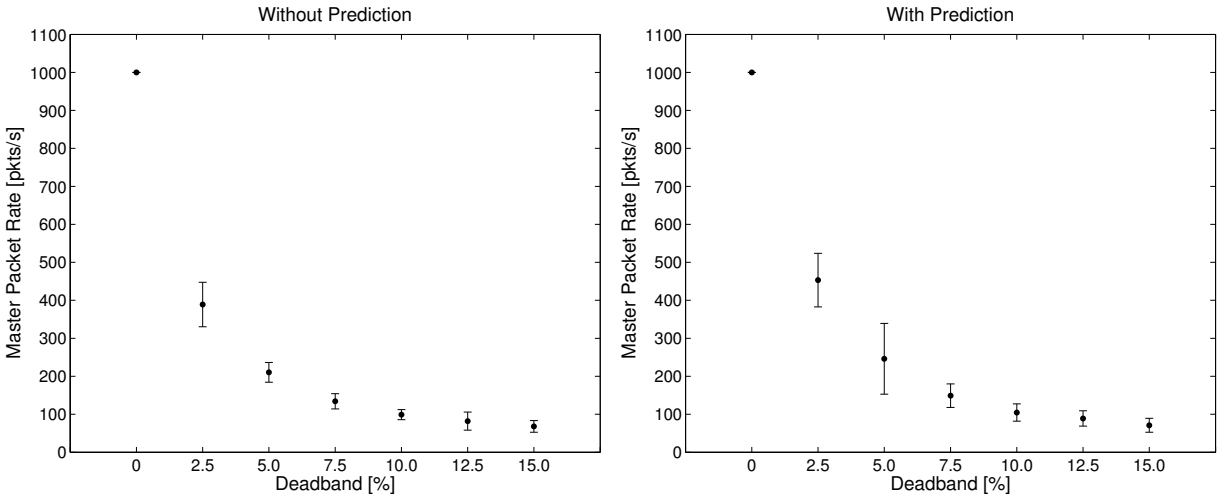

Figure 3.4: Measured packet rates and their standard deviations for different data reduction parameters without (left) and with (right) prediction for the master side.

The Friedman variance test on the packet rates on the master side yielded strong significance in both the cases without prediction ($\chi^2(6) = 177.61, p < .001$) and the cases with prediction ($\chi^2(6) = 148.89, p < .001$).

The pairwise comparison using the Wilcoxon Signed Rank Tests for the significance of packet rate reduction between deadband parameter steps and their respective predecessor yielded high significance ($p < .001$) in all cases. That means that every increase of the deadband parameter by 2.5% in the experiment yielded another significant reduction of the packet rate on the master side.

The evaluation if prediction leads to further significant packet rate reduction was done by pairwise comparison of the rates of the same deadband parameter steps with and without prediction. By looking at the results for the master side, we can see that the cases with prediction led, inversely to our expectations, to an increase in packet rates in all cases. This is a first surprising result, since there seems to be a greater difference of prediction

behavior to the VR-case described in Section 2.4.1 as anticipated. In a real TPTA-system, with the control approach used in this experiment, prediction is not worthwhile, at least for velocity and position data.

3.9.2.2 Slave Side

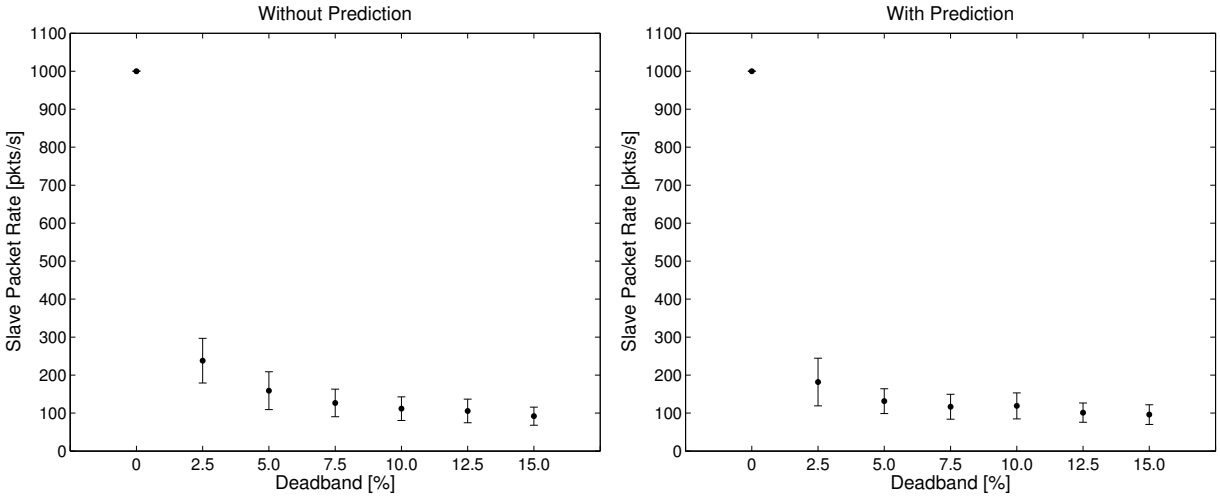

Figure 3.5: Measured packet rates and their standard deviations for different data reduction parameters without (left) and with (right) prediction for the slave side.

The Friedman variance test on the packet rate generated by the slave side of the experiment (containing force information) yielded strong significance for both the cases without prediction ($\chi^2(6) = 148.89, p < .001$) and with prediction ($\chi^2(6) = 132.34, p < .001$).

Pairwise comparisons in the case without prediction between deadband parameter steps and their respective predecessors yielded the results shown in Table 3.1. We can see very strong significance for the first three deadband parameter steps from 2.5% to 7.5% and borderline non-significance for the fourth step at 10% (due to Bonferroni-corrections the significance level begins at .008 here) and another significant change at 15%. We can state that increase of the deadband parameter in the range up to 7.5% leads to significant changes. After that changes are sometimes not significant any more. By looking at the respective figure we can confirm this, because for parameters greater 7.5% the packet rate is already very low and decreases only marginally for higher values.

The same analysis is done for the case with prediction. Results are shown in Table 3.2. Here again significant changes are recorded up to a deadband of 7.5% and again for 12.5%. Since the packet rates are already very low for 12.5% we can say that worthwhile packet rate reductions only take place for deadbands up to 7.5%.

Comparison of deadband %	Probability	significant
0 and 2.5	$p < .001$	yes
2.5 and 5	$p < .001$	yes
5 and 7.5	$p < .001$	yes
7.5 and 10	$p = .01$	no
10 and 12.5	$p = .27$	no
12.5 and 15	$p < .008$	yes

Table 3.1: Significance of packet rate reduction difference between subsequent deadband parameter steps on the slave side without prediction.

Comparison of deadband %	Probability	significant
0 and 2.5	$p < .001$	yes
2.5 and 5	$p < .008$	yes
5 and 7.5	$p < .008$	yes
7.5 and 10	$p = .89$	no
10 and 12.5	$p < .001$	yes
12.5 and 15	$p = .20$	no

Table 3.2: Significance of packet rate reduction difference between subsequent deadband parameter steps on the slave side with prediction.

Deadband in %	Probability	significant
2.5	$p < .001$	yes
5	$p = .06$	no
7.5	$p = .14$	no
10	$p < .05$	yes
12.5	$p = .31$	no
15	$p = .10$	no

Table 3.3: Significance of packet rate reduction difference between cases with and without prediction on the slave side.

The results of the pairwise comparisons between the cases of one deadband parameter with and without prediction are shown in Table 3.3. We can see that only deadband parameters of 2.5% and 10% prediction lead to significant gains whereas for the other parameters prediction is not worthwhile. The only really strong significance was detected at 2.5% so that we can reason that prediction above this value should not be used for force transmission, especially when taking the results of the following task performance analysis into account. But at least for the force transmission case, prediction leads to packet rate reduction in contrast to velocity/position transmission discussed above.

3.9.3 Subjective Quality

The subjective quality assessment of the human system interface used for the experimental task is a very important measure for the performance of the data reduction algorithms. Subjects had to answer 3 questions concerning the subjective quality after every test run of the experiment. The scale for all three questions ranged from 1 (best quality) to 7 (worst quality). For the evaluation, the average score of these 3 questions was used as the main quality measure.

In Figure 3.6 the average ratings and their standard deviations for all combinations of independent variables, i.e., all 7 deadband settings respectively with and without prediction are shown.

In the case where DBDR was used to reduce the amount of data communicated in the system an ANOVA was conducted to evaluate whether the encountered differences in the quality ratings are significant or not. The analysis showed no significant dependency ($F(6, 180) = 1.17, p = .33$) between the assessed quality ratings and the size of the used deadband parameter which seems reasonable when looking at the left diagram in Figure 3.6 where no real dependency is visible.

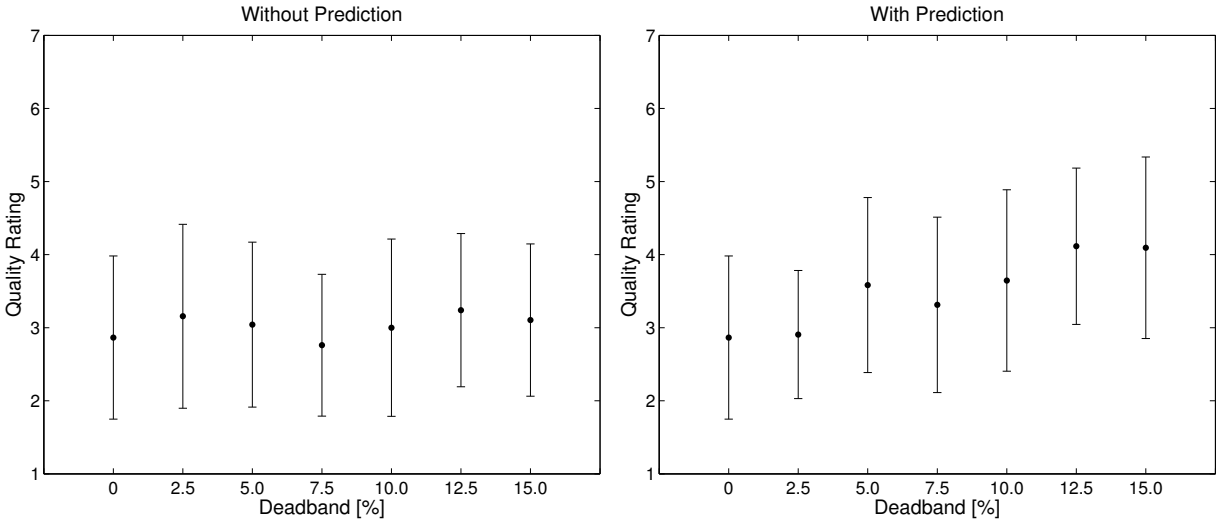

Figure 3.6: Subjective quality ratings (1=best, 7=worst) and their standard deviations the test subjects gave for different data reduction parameters without (left) and with (right) prediction.

Deadband in %	F-statistics	significant
2.5	$F(1,30) = 0.14 \quad p = .71$	no
5	$F(1,30) = 9.67 \quad p < .05$	yes
7.5	$F(1,30) = 3.83 \quad p = .06$	no
10	$F(1,30) = 13.97 \quad p < .05$	yes
12.5	$F(1,30) = 39.28 \quad p < .001$	yes
15	$F(1,30) = 25.18 \quad p < .001$	yes

Table 3.4: Significance of effects in subjective quality assessment while using the DBDR-P approach.

The ANOVA performed for the case where DBDR-P was used showed a very significant main effect ($F(6,180) = 10.22, p < .001$) which could have also been anticipated from Figure 3.6. The single contrasts for this case are shown in Table 3.4. We can see that only in the 2.5%-case a non significant quality deterioration was measured. The 7.5%-case is only borderline non significant with its $p = .06$. Therefore we can state that the DBDR-P approach did not perform very well in this real TPTA-system unless it is used in conjunction with very low deadband values.

3.9.4 Task Performance

3.9.4.1 Task Completion Time

The assessment procedure for the task completion time is described in detail in Section 3.4. The test subjects had to haptically follow the plastic profile and subsequently return to the starting position in free space movement. Since the subjects were told to fulfill the task as quickly and as precisely as possible, some of the subjects emphasized the speed whereas others took more care of the precision of their task execution. This is probably the reason for the relatively high level of the encountered standard deviations. The exact measured task completion times are shown in Figure 3.7.

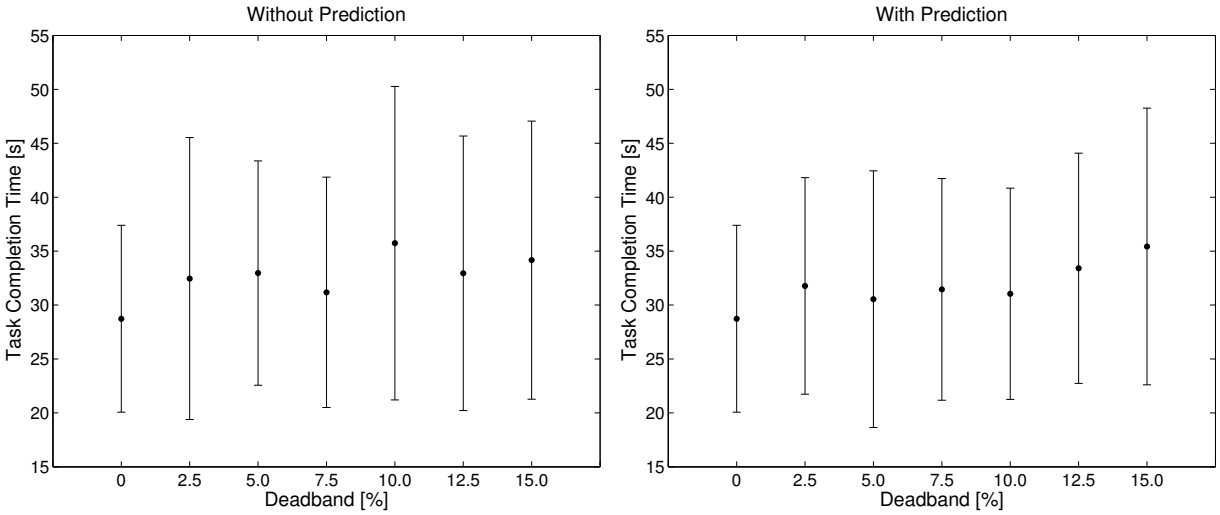

Figure 3.7: Measured task completion times and their standard deviations (in s) for different data reduction parameters without (left) and with (right) prediction.

Since Mauchly's Test showed that the conditions for sphericity (see 3.8.2) were not fulfilled in this case, the according corrections were applied to the degrees of freedom of the ANOVA.

In the case without prediction a significant main effect was detected ($F(4.05, 121.45) = 2.75, p < .05$). In order to further evaluate when the greatest dependencies were detected, the single contrasts are shown in Table 3.5. The analysis shows that all deadband sizes had at least an almost significant effect (highest $p = .08$ for 7.5%) on the task completion time in comparison to the uncompressed 0% case.

The analysis of the cases including the prediction approach also shows a significant main effect ($F(4.55, 136.60) = 3.73, p < .05$). Table 3.6 gives details on the single effects. We can see that the 2.5%, 12.5%, and 15% case show significant effects whereas the 5%, 7.5%, and 10% cases don't.

Deadband in %	F-statistics	significant
2.5	$F(1,30) = 3.75 \quad p = .06$	no
5	$F(1,30) = 12.80 \quad p < .01$	yes
7.5	$F(1,30) = 3.38 \quad p = .08$	no
10	$F(1,30) = 10.52 \quad p < .01$	yes
12.5	$F(1,30) = 5.39 \quad p < .05$	yes
15	$F(1,30) = 7.58 \quad p < .05$	yes

Table 3.5: Significance of effects on task completion time while using the DBDR approach.

Deadband in %	F-statistics	significant
2.5	$F(1,30) = 10.42 \quad p < .05$	yes
5	$F(1,30) = 0.59 \quad p = .45$	no
7.5	$F(1,30) = 2.86 \quad p = .10$	no
10	$F(1,30) = 2.14 \quad p = .15$	no
12.5	$F(1,30) = 11.66 \quad p < .05$	yes
15	$F(1,30) = 10.42 \quad p < .05$	yes

Table 3.6: Significance of effects on task completion time while using the DBDR-P approach.

The results from the test for influences of the compression parameters on the task completion time are somewhat inconclusive. On the one hand it seems that the DBDR approach without prediction, even in the low 2.5% case, causes strong deterioration in task completion time, whereas in the case with prediction three parameter sets did not cause significant deterioration. Especially the 5% case is odd because it goes from very strong significance without prediction to no significance at all with prediction. It seems that the vague experimental guideline to complete the task as quickly and as precisely as possible may have led to a behavior of test subjects to do some of the test runs slower but more precise and others quicker and less precise in a completely random manner.

3.9.4.2 Number of Contact Losses

During the contact phase of the experiment test subjects often had a hard time maintaining contact to the profile surface because of the very sharp turns that were necessary to follow it. Especially if they focused on quick task completion it became more difficult to maintain steady contact. One of the measured performance parameters was the number of contact losses longer than 50ms during the contact phase. In Figure 3.8 the recorded average number of contact losses and their standard deviations are shown.

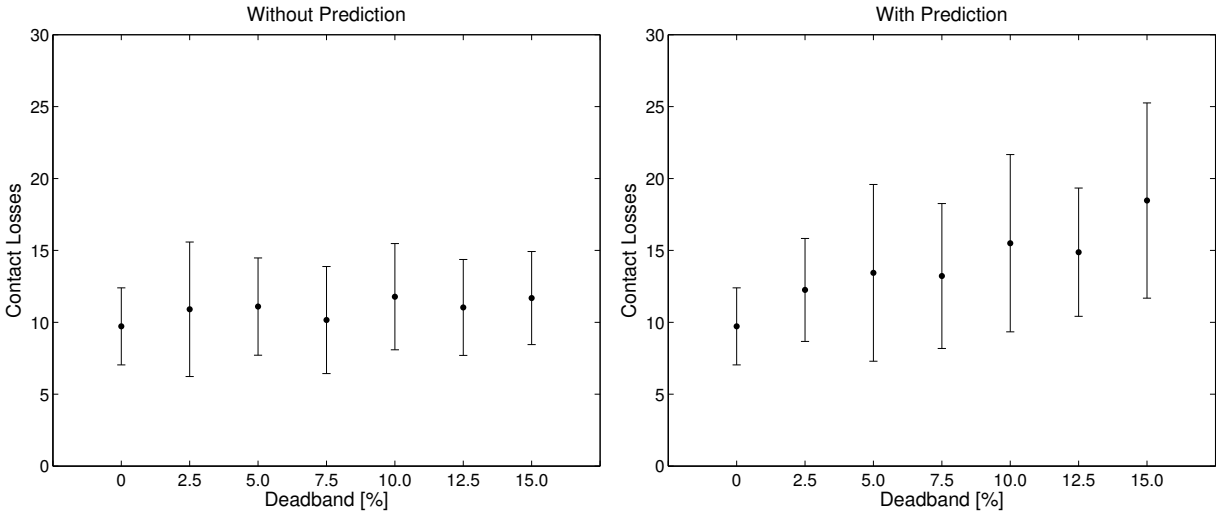

Figure 3.8: Measured number of contact losses and their standard deviations for different data reduction parameters without (left) and with (right) prediction.

Deadband in %	F-statistics	significant
2.5	$F(1,30) = 2.40 \quad p = .13$	no
5	$F(1,30) = 3.67 \quad p = .07$	no
7.5	$F(1,30) = 0.78 \quad p = .39$	no
10	$F(1,30) = 7.95 \quad p < .01$	yes
12.5	$F(1,30) = 3.84 \quad p = .06$	no
15	$F(1,30) = 7.90 \quad p < .01$	yes

Table 3.7: Significance of effects on the number of contact losses while using the DBDR approach.

Since Mauchly's Test showed that the conditions for sphericity (see 3.8.2) were not fulfilled for the contact loss data set, the according corrections were applied to the degrees of freedom of the ANOVA.

In both cases with and without prediction a significant main effect was detected ($F(4.49, 134.80) = 2.38, p < .05$ for the case without prediction and $F(3.72, 111.45) = 15.12, p < .001$ for the case with prediction). Corresponding tendencies are also visible in the respective figures.

For the case without prediction the respective comparison tests to the reference case of 0% deadband are shown in Table 3.7. We can see significant effects for the 10% and 15% cases and two borderline not significant results in the 5% and 12.5% cases. Interestingly, the 7.5% case has the least significance.

Deadband in %	F-statistics	significant
2.5	$F(1,30) = 14.58 \quad p < .05$	yes
5	$F(1,30) = 10.04 \quad p < .05$	yes
7.5	$F(1,30) = 12.85 \quad p < .05$	yes
10	$F(1,30) = 18.85 \quad p < .001$	yes
12.5	$F(1,30) = 31.48 \quad p < .001$	yes
15	$F(1,30) = 48.29 \quad p < .001$	yes

Table 3.8: Significance of effects on the number of contact losses while using the DBDR-P approach.

In the case with prediction the single contrasts to the 0% deadband reference shows significance for all test cases, where the first three (2.5%, 5%, and 7.5%) show smaller significance than the other three (10%, 12.5% and 15%).

For the parameter of encountered contact losses during the contact phase of the experiment we can see that deadbands lower than 10% lead to no significant deterioration of contact behavior if no prediction is used. With activated prediction, all deadbands lead to significant deterioration of contact losses which again shows that the prediction approach seems not a valid solution in real TPTA systems with the used control structure in comparison to virtual TOPs.

3.9.4.3 Contact Ratio

Another interesting performance parameter is the contact ratio which describes what relative part of the time during the contact phase was actually spent in contact with the environment. It is calculated as follows:

$$\text{Contact Ratio} = \frac{\text{Time in contact}}{\text{Total duration of contact phase}} \tag{3.3}$$

Consequently, a value of 1 or 100% means, that the test person maintained contact to the surface the whole time without a single interruption. This would be the optimal case. The contact ratio reflects the ability of the test person to control the TOP with sufficient precision to maintain contact with the environment using the TPTA system. Figure 3.9 shows the recorded mean duration ratios and their standard deviations for both the cases with and without prediction.

As in many of the cases before, Mauchly's Test showed that the conditions for sphericity were not fulfilled for the contact ratio data set. Accordingly, corrections were applied to

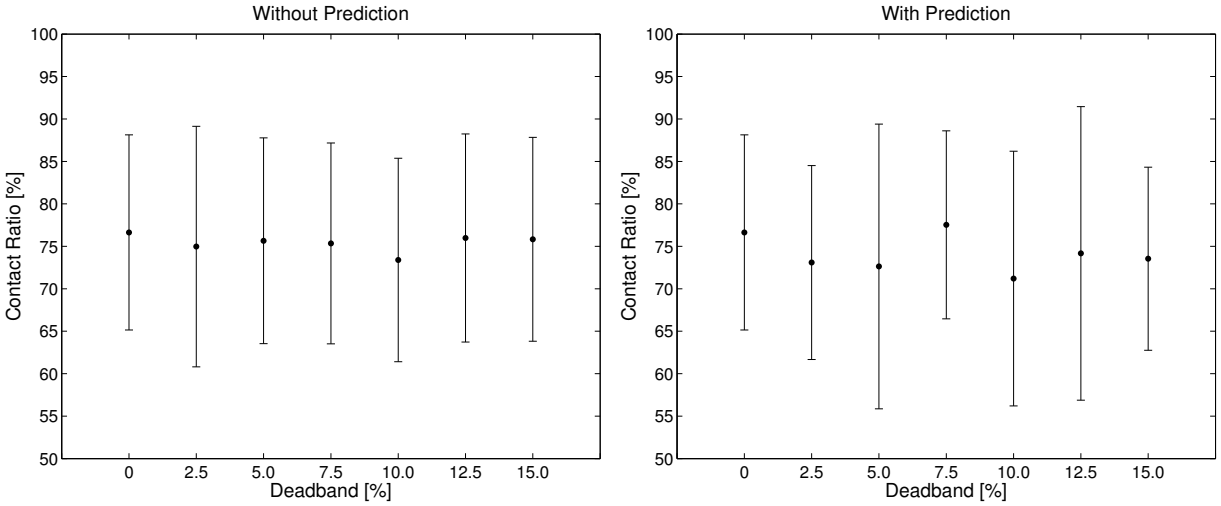

Figure 3.9: Measured ratio between the actual contact time and the length of the contact phase for different data reduction parameters without (left) and with (right) prediction.

the degrees of freedom of the ANOVA.

The statistical analysis showed no significant main effect for the case with no prediction ($F(3.92, 117.46) = 0.34, p = .85$). For the case with prediction, there also was no detectable main effect ($F(4.36, 130.88) = 1.15, p = .34$). As we can also see by looking at the figures, the case with prediction shows greater deviations than the case without prediction which is also reflected by the p-values of the F-tests.

Consequently, we can state that the applied data reduction algorithms did not have a significant effect on the contact ratio of the test subjects in the presented experiment.

3.9.4.4 Contact Force Behavior

While the test subjects performed the contact phase of the experiment, the contact force applied to the remote environment was measured. The variance of this contact force is one measure for the task performance because it reflects the amount of precision and control the subjects were able to bring across the TPTA system to the remote environment. The measured force variance values are shown in Figure 3.10 for both the cases with and without prediction.

Mauchly's Test showed that the conditions for sphericity (see 3.8.2) were not fulfilled for force variance. Therefore the according corrections were applied to the degrees of freedom of the ANOVA.

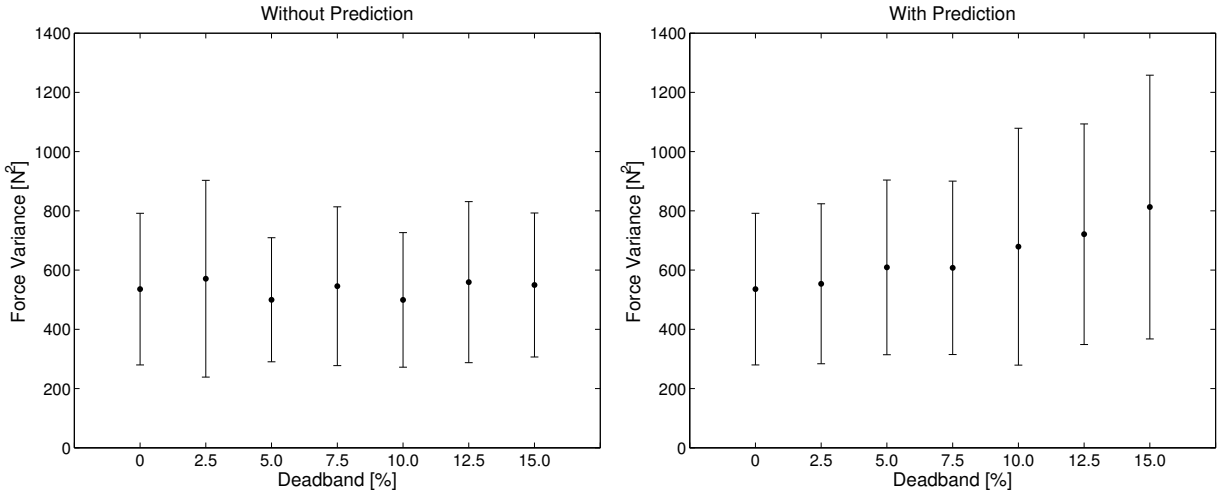

Figure 3.10: Measured force variance during the contact phase for different data reduction parameters without (left) and with (right) prediction.

The statistical analysis showed no significant main effects for neither the case without prediction ($F(1.55, 46.59) = 0.80, p = .43$) nor for the case with prediction ($F(2.23, 66.84) = 1.18, p = .32$). Still, a slight tendency for higher force variances is visible in the respective figure for the higher deadband parameters when prediction is active.

Not a real performance measure but also of interest is the applied mean force during the test runs. Figure 3.11 shows the measured mean force in all cases.

Since no significant effects were found in this case, we can state that the DBDR approach did not significantly interfere with the contact force behavior during the presented TPTA experiment.

3.9.4.5 Track Deviation

During the contact phase of the experiment the test subjects ought to follow a track across the plastic profile which was marked by white dots on the surface. One parameter to measure task performance is the deviation from this optimal path. Since there was no way to haptically sense the correctness of the current path, the subjects had to rely on the intentionally suboptimal video feedback from the remote environment. The measured variance of the track deviation is shown in Figure 3.12 for both the cases with and without prediction.

Again here Mauchly's Test showed that the conditions for sphericity (see 3.8.2) were not fulfilled, so the according corrections were applied to the degrees of freedom of the ANOVA.

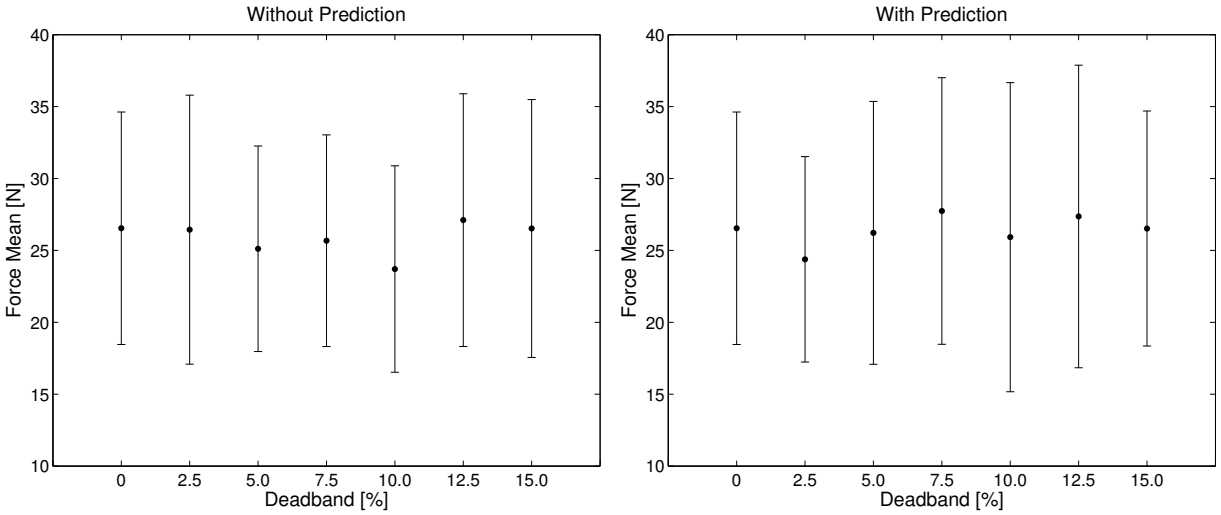

Figure 3.11: Measured mean contact force during the contact phase for different data reduction parameters without (left) and with (right) prediction.

For both the case without prediction ($F(3.62, 108.71) = 1.49, p = .22$) and the case with prediction ($F(3.70, 111.12) = .54, p = .69$) no significant main effect was detected.

Also interesting is the mean track deviation the subjects encountered during their test runs. The mean deviation from the optimal track is shown in Figure 3.13. Note that almost all mean values are negative. That most likely comes from the setup of the experiment where HSI positions closer to the body of the operator result in negative deviations from the optimal track. Since human interaction precision decreases with the distance from the body this result is not surprising.

No significant effects were detected in this case. This leads to the conclusion that the DBDR and DBDR-P approaches did not deteriorate the tracking efficiency of the test subjects in this experiment. It is most likely that the encountered tracking errors result more from the relatively bad video feedback from the remote environment.

3.10 Discussion

There are several interesting conclusions to be drawn from the experimental results. The first and most obvious one is that the usage of a linear signal model based prediction does neither lead to packet rate reduction in all cases nor does it seem otherwise suitable because it leads to strong deterioration of task performance and perceived quality.

The reason for this bad performance lies in the handling of position/velocity transmission

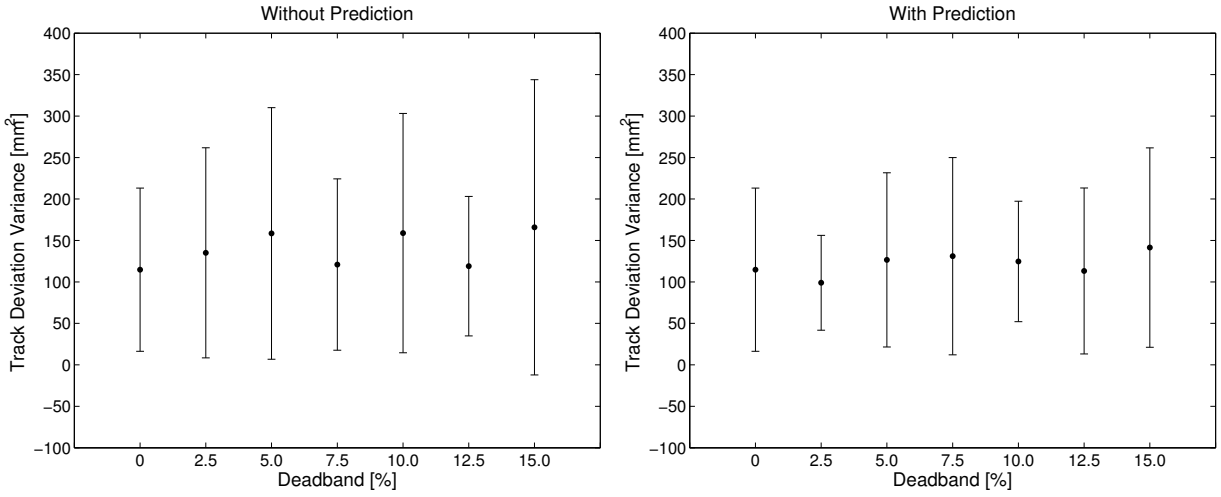

Figure 3.12: Measured variance of the track deviation during the contact phase for different data reduction parameters without (left) and with (right) prediction.

in this system setup. As stated in Section 3.2 the actual TOP position is based on a prediction according to the last transmitted position and the respective velocity at that time and the elapsed time since. So even if only DBDR is in place the TOP position is already predicted. This works well, as the experimental results show. When model based prediction is added with the DBDR-P approach, which tries to linearly predict velocity from the last two transmitted velocity samples we basically have a position prediction running based on a velocity prediction. This quickly leads to increased position errors which have to be corrected by sudden movements of the TOP when they occur. While this is not a big problem in virtual remote environments where such position errors can be compensated easily by simply setting new positions for the virtual TOP, a real TOP has to be moved to the correct position by accelerating and decelerating it in a very short time period, which always leads to disturbances in the system when it is necessary.

There are a couple of possible measures to deal with this disadvantage in position/velocity transmission. One would be to introduce a control system that allows for less sudden position corrections, possibly by adding increments to the set positions which lead to a correct position over the next n steps. Another possibility would be to introduce fixed rate updates at 10 to 50Hz to keep the necessary corrections small so they don't disturb interaction that heavily. It would also be possible to keep track of the position error at the TOP side since all information to do that are also available at the OP side. By setting a reasonable threshold for the position error, an update packet could be sent accordingly. Solutions like this should be tested in further experiments to evaluate whether they work well or lead to more challenges along the way.

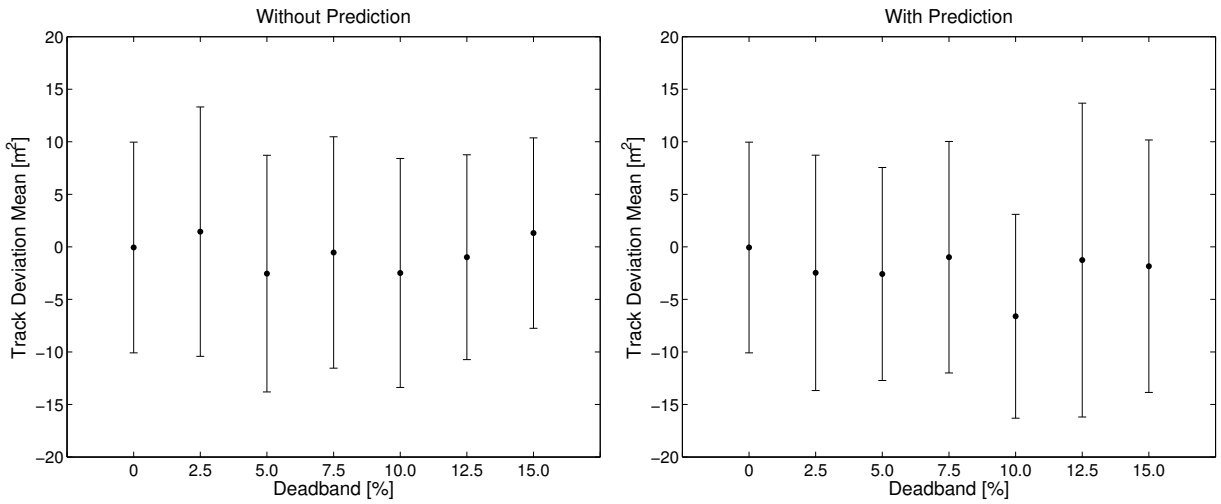

Figure 3.13: Measured mean of the track deviation during the contact phase for different data reduction parameters without (left) and with (right) prediction.

While the DBDR and DBDR-P approaches did not deteriorate most of the task performance parameters, their influence on task completion time remains inconclusive. While significant adverse effects were found for otherwise well performing parameter sets, non-significant influence while using otherwise bad performing parameter sets were also recorded. The most likely reason for this behavior is the ambiguous nature of the task explanation which required both speed and precision at the same time.

3.11 Chapter Summary

This chapter tries to determine the influence of psychophysically motivated data reduction approaches, namely the DBDR and DBDR-P approaches on the task performance in a real TPTA system. Many different aspects of task performance and also the achieved data reduction performance is assessed and analyzed in order to be able to give guidelines and useful information for the application of such algorithms in real TPTA systems.

The experiment described in this chapter was a joint effort of the author and the Human Factors Institute at the University of the Armed Forces in Munich which provided the psychological background for the statistical analysis and the professional conduction of the experiment itself. All the hardware of the experiment has been used by courtesy of the Institute of Automatic Control Engineering at TU München. The experimental design, the communication software incorporating the examined data reduction algorithms, the data extraction from the measurement logs as well as interpretation of the statistical analyses

were the main contributions of the author.

The results indicate that the combination of the DBDR-P approach on velocity data and position control does not work well in real TPTA systems while working perfectly well in virtual remote environments. But after ruling out signal model based prediction in the control architecture as used in the experiment we can state that the DBDR approach without prediction works very well in the presented TPTA system. It does not lead to significant deterioration of subjective quality perception, does not significantly increase the number contact losses in complex contact situations, does not lead to significant changes in contact force behavior, and does not significantly deteriorate tracking accuracy. On the other hand it does significantly affect packet rates across the network by up to 93% of the original rate for velocity/position packets and up to 91% of the original rate for force packets (at 15% deadband). Still at a deadband parameter of 7.5% which represents a reasonable value where the hardware is almost not stressed at all by sudden position corrections and there is no significant packet rate gain any more at the slave side, rates are already at 87% for both velocity/position and force transmission.

We can finally conclude that the 3D-deadband approach also works well in real TPTA systems.

Chapter 4

Theoretical Analysis of the Deadband Approach

After demonstrating the effectiveness of the deadband approach for various scenarios in the last chapters, a more theoretical view is now presented in order to further analyze its characteristics. In the following, the deadband approach is studied in terms of an adaptive sampler. Hence, we now refer to it as the deadband sampler.

4.1 Types of Sampling

Sampling is omnipresent in today's digital technology. Generally, sampling describes the transformation of a continuous signal into a discrete signal. Special forms of sampling like oversampling and undersampling are also used to transform one discrete signal representation into another.

While the dimension in which sampling can take place is quite arbitrary, we concentrate on the sampling in the time domain. In the vast majority of cases uniform sampling is used, where a signal is sampled in certain constant intervals, the sampling period t_s, resulting in the sampling frequency $f_s = \frac{1}{t_s}$.

In order to be theoretically able to reconstruct the continuous signal, a sampling frequency of at least $f_s = 2f_m$ has to be used to sample the signal, where f_m is the highest frequency in the continuous signal. This is known as the Nyquist-Shannon sampling theorem and is a very basic concept in signal processing.

For perfect signal reconstruction, the sampled values have to have infinite resolution. In reality quantization is used to make the sampling values manageable but this also introduces

irreversible loss of information.

In comparison with well known uniform sampling non-uniform sampling opens up a whole lot of new possibilities in signal processing but also introduces additional reconstruction problems.

Non-uniform or adaptive sampling allows for irregular sampling intervals which may be tailored to the specific signal properties at a given time. Signal parts with low frequency content can be represented using fewer samples compared to parts with higher frequency content. On the other side, signal reconstruction becomes more difficult because the actual sampling frequency at a given signal position is not always easy to determine.

Almost all the work on non-uniform sampling seeks for the goal of best possible signal reconstruction. In comparison to this goal, the deadband sampler described in this work aims at the selection of the least number of samples from a signal which then result in a perceptually identical reconstructed signal. In other words, the signal deterioration or sampling noise introduced by the sampler should be below the perceptual threshold of a human being.

4.2 Principles of the Deadband Sampler

As the deadband sampler uses the underlying principle of the deadband approach presented in Section 2.2, it is also based on Weber's Law (see 1.1.6.2). It therefore generates a constant output signal q as long as the input signal does not exceed the deadband threshold $q \pm pq$ which would optimally correspond to the actual perception threshold of a human being. Once this threshold is exceeded, the output signal is updated to the current signal value and this value is held at the output until the new threshold around the output signal is exceeded. This results in irregular sampling intervals and therefore in non-uniform sampling.

A comparison with classic predictive compression techniques as discussed in [24] yields some similarities. An approach mentioned in [24] checks whether the prediction error at the sender side exceeds a given threshold. If this is the case, the new value is sent. If it is not the case, the predicted value is used at the receiver side. The deadband sampler differs in two major aspects from this classic approach. Firstly, the deadband sampler changes the threshold linearly with the amplitude of the current output value and secondly, the linear relationship is chosen to reflect the perception capabilities of a human being.

The deadband sampler is usually used on already sampled input signals in order to non-uniformly downsample them. Note that it would still be possible to use the deadband

sampler on a continuous input signal as well.

4.2.1 Input-Output relationship

Figure 4.1: The block diagram of the deadband sampler.

The main rule how the output signal is generated is described in the following. The discrete input signal is defined as:

$$x_i \in (-\infty, +\infty) \qquad i = 1 \dots N \tag{4.1}$$

where i is the sequence number of a sequence of N input samples.

The output signal of the quantizer q_i is generated by the following rules:

$$q_i = \begin{cases} x_{i-m} & \text{if} \quad x_{i-m}(1-p) < x_i < x_{i-m}(1+p) \\ x_i & \text{else} \end{cases} \tag{4.2}$$

and

$$q_{i-1} \dots q_{i-m} = x_{i-m} \tag{4.3}$$

for

$$i - m \geq 1 \tag{4.4}$$

where m samples back in the signal the last threshold violation took place. We call x_{i-m} the reference value. p is the deadband threshold approximated by Weber's Law. As previously described, p in most cases ranges from 0.05 to 0.15 for human haptic perception.

4.2.2 Update rate behavior

For the transmission of sensor data over limited channels using the deadband approach it is very useful to know in advance which p will be necessary in order to generate a certain output rate. Since this behavior strongly depends on the input signal we have to make a few simplifications. In the following we derive the update rate behavior for the deadband sampler for different types of input signals.

4.2.2.1 Complexity Considerations

To explain the necessity of deeper theoretical considerations despite the simplicity of the deadband approach we give a short example of the complexity for the case of an independent and identically distributed (iid) input signal. Special cases for this type, namely uniformly distributed and normally distributed input signals, will be looked at later. First we present some principal ground rules of the theoretical analysis.

At first glance one might think that with an iid input signal an analysis would be very simple because of the independence of the samples. Unfortunately this is not the case as is easily proven by a simple simulation. Looking at Figure 4.2 we can see the results of a simulation run of one million normally distributed sample signals with 10 steps each, which were processed using the deadband sampler and analyzed for the probability of each sample in the signal sequence violating the currently active deadband threshold.

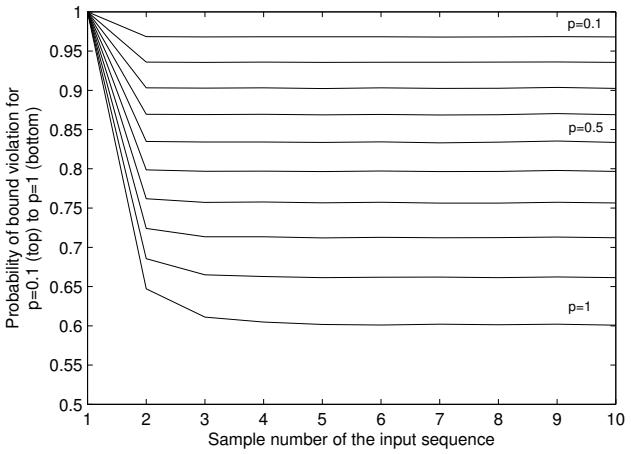

Figure 4.2: Probabilities of deadband bound violation for $p = 0.1$ (top) to $p = 1$ (bottom) for the first ten steps of a normally distributed input signal.

Assuming that the probability that one value of the input sequence lies within the deadband bound of another input value is γ we can derive the probability for the first steps in the sequence for violating the deadband bound.

For the first sample, the value x_1 has to be sent to the receiver. Therefore, the probability of bound violation is 1 in this case. This is also true in the simulation.

For the second sample, the deadband bound is violated once the value x_2 does not lie within the deadband bound around the first sample x_1. Therefore the probability for this case is $1 - \gamma$. This is also true for all p in our simulation.

For the third sample, the deadband bound can be violated in two different ways. The first possibility is that the third sample x_3 violates the deadband around x_2. However, this does only matter if x_2 itself lay outside the deadband around x_1 and became the new reference value. So the probability for this first case is $(1-\gamma)\cdot(1-\gamma)$. The second possibility is that the third sample x_3 violates the deadband bound of x_1 in the case that x_2 lay within the deadband bound of x_1 and therefore did not become the new reference. The probability for the second case is $\gamma \cdot (1-\gamma)$. To get the probability that x_3 violates the threshold, we must add up these two probabilities:

$$(1-\gamma)\cdot(1-\gamma)+\gamma\cdot(1-\gamma) = 1-2\gamma+\gamma^2+\gamma-\gamma^2 = 1-\gamma \tag{4.5}$$

We can see that the probability that x_3 violates the threshold is the same as for x_2, namely $1-\gamma$. If we compare this result with the simulated probabilities of bound violation in these steps we can clearly see that this is not correct. Especially for higher values of p the simulated probabilities for x_3 differ by quite a large margin from the probabilities for x_2. Therefore this case is not as trivial as it seems at first and the simple assumption that there is a certain γ throughout the whole signal is wrong.

The reason for this discrepancy is that γ is only the correct probability for the second step. In all following steps there are strong dependencies between the probabilities in the current step and those of the previous steps which lead to changes in the probability distribution of the output signal which is then no longer Gaussian. Therefore it is in no way possible to give a γ which is constant over the whole signal because the probability distribution of the output signal changes with every step. The probability that a new incoming signal sample violates the currently active threshold therefore depends on the whole history of bound violations in the past of the signal.

In order to further explain how the probabilities for an arbitrary signal model can be calculated, we have to formalize the behavior of the system. In general we can say that for every step it is either possible that the new incoming input value violates the active deadband or it does not. Depending on the signal model it can be necessary to know the complete past of the input and output signals in order to determine the probability for the violation or compliance in the next step. Therefore we need a way to distinguish between all possible signal behaviors as far as bound violation is concerned. The only way to represent this is a binary tree. The two leaves of every node represent bound violation and no bound violation. This tree would then look like as shown in Figure 4.3.

The superscripted bit sequences t in Figure 4.3 (e.g. q_3^{101}) denote the sequence of bound violation (1) and no bound violation (0) along the signal. The index is the sequence count as already introduced earlier. Each of the nodes in the binary tree has a specific

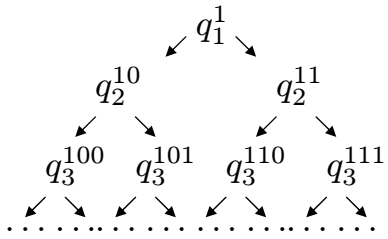

Figure 4.3: Binary tree structure representing all possible output sequences concerning bound violations.

probability that a signal reaches it. This probability depends on the probability of some or all preceding nodes in direction to the root. So for example the probability that a signal experiences a violation in step one, no violation in step two and again a violation in step three, i.e., q_3^{101}, depends on the probability of q_2^{10} and q_1^1 or at least one of the two.

In order to determine the probability of occurrence for each node in the tree, a signal model dependent algorithm has to be applied on the preceding nodes in the tree. The algorithms for some types of input signals will be given later on in detail. It is always necessary that the distribution and behavior of the input signal model is known as well as the output distributions of the preceding nodes. Following the notations from above, the distributions of the involved signals are denoted $D_{q_i^t}$ for the distribution of the deadband sampled output signal at step i of the signal. t is the bit sequence which describes the history of the signal up to the node (again 1 for bound violation, 0 for compliance). Note that all $D_{q_i^t}$ except for $D_{q_1^1}$ are not real PDFs, because their integral is not 1 as necessary for a real PDF but the probability of occurrence of the respective tree node $P_{q_i^t}$. In order to convert them to real PDFs a correction factor of $\frac{1}{P_{q_i^t}}$ has to be applied. We call those functions $D_{q_i^t}$ partial PDFs from now on, knowing we can convert them to real PDFs any time using the aforementioned correction factor.

4.2.2.2 Convergence

The first level of the tree only consists of one node, namely q_1^1. The second level of the tree consists of two nodes, q_2^{10} and q_2^{11}, level three of four nodes and so on. The n-th level consists of 2^{n-1} nodes. Of course, if all node probabilities of one level are summed up the result is always 1. If we want to determine the probability that a signal violates the deadband bound in step n, denoted as P_{Ln} in the following, we have to sum up all probabilities in level n of the tree that correspond to a violation in step n. These are 2^{n-2} values, namely all with a 1 at the end of the superscript. For the first levels this looks as follows:

$$\begin{aligned} P_{L2} &= P_{q_2^{11}} \\ P_{L3} &= P_{q_3^{101}} + P_{q_3^{111}} \\ P_{L4} &= P_{q_4^{1001}} + P_{q_4^{1011}} + P_{q_4^{1101}} + P_{q_4^{1111}} \\ &\vdots \end{aligned} \tag{4.6}$$

As we can see from the simulation shown in Figure 4.2 the probability P_{Ln} quickly converges to a certain value. We call this value the steady state probability P_{SS}. P_{SS} represents the average probability that a bound violation takes place in every step of the whole signal and therefore is exactly the measure we look for to represent the update rate behavior of the deadband sampler. By multiplying the incoming rate R_x with P_{SS} we get the average outgoing rate R_q or, expressed differently

$$\frac{R_q}{R_x} = P_{SS} \tag{4.7}$$

the data reduction ratio of the deadband sampler is P_{SS}.

Of course, steady state probability convergence depends heavily on the signal model itself. In Figure 4.4 we can see the convergence behavior for four different signal models, two iid, a uniformly distributed signal ranged from 0 to 1 and a standard Gaussian distributed signal, and two non-iid signals, a Gaussian autoregressive signal with AR-coefficient 0.5 and a Gaussian autoregressive signal with AR-coefficient 1 (Wiener Process or random walk model). Both the random increment in the AR cases and the Gaussian iid signal itself had zero mean and a variance of 1 ($\mathcal{N}(0, \sigma^2 = 1)$). We will have a closer look at these signal models in the following sections. In the figure, we can observe multiple things. We can see that for smaller p the convergence is faster than for larger p. For the Gaussian signal model the convergence is faster than in the uniform case. The Gaussian AR-case converges more slowly with growing AR-coefficient until in the case of the Wiener Process, no convergence takes place because of the unbounded nature of the signal.

In the following we will take a closer look at different input signal models and the deadband sampler's behavior for those models.

4.3 iid Input Signals

First we present the tree generation algorithm for iid input signals. To make the operating principle of the algorithm clear we start off at the first input sample. Since we assume

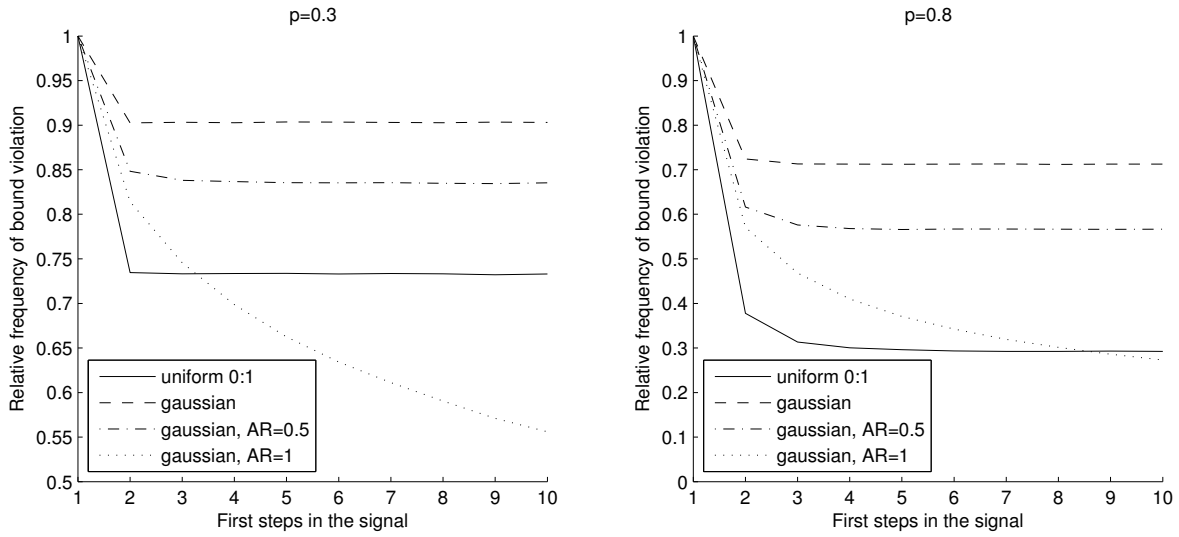

Figure 4.4: Steady state probability convergence for different signal models for $p = 0.3$ and $p = 0.8$.

that no deadband threshold has been established at the beginning of the signal or, in other words, the initial value is assumed to be zero with the deadband also being zero, the first sample definitely violates the deadband and becomes the first reference value (assumed it is not zero). Therefore the PDF of the first deadband sampler output sample is the same as the input signal.

$$D_{q_1^1} = D_{x_i} \tag{4.8}$$

The probability of occurrence of this node, denoted as $P_{q_i^t}$, can be calculated as the integral over the PDF. In general

$$P_{q_i^t} = \int_{-\infty}^{+\infty} D_{q_i^t}(x)dx \tag{4.9}$$

and in this special case

$$P_{q_1^1} = \int_{-\infty}^{+\infty} D_{q_1^1}(x)dx = 1 \tag{4.10}$$

For the next level in the tree, which corresponds to the second sample in the signal, the first time two outcomes are possible: Either the new incoming input sample value violates

the deadband bound around the current reference value and becomes the new reference and output value, or it complies to the deadband bound and the reference and output value stays the same. The ways of calculating the partial PDFs of these two cases are quite different.

We start with the calculation of $D_{q_2^{10}}$. In this case the second sample lies within the deadband bound around the first. In order to determine the partial PDF of possible values in this node of the tree we need to solve the following equation:

$$D_{q_2^{10}}(x) = D_{q_1^1}(x) \int_{x(1-p)}^{x(1+p)} D_{x_2}(y)dy \tag{4.11}$$

The integral over the input signal distribution D_{x_2} within the limits from $x(1-p)$ to $x(1+p)$ yields the probability that a new input value lies within the deadband bound of a certain value x. This probability is multiplied by the probability of occurrence of a certain value x in the distribution of possible values in node 1 of the tree $D_{q_1^1}$. The equation evaluates this for all possible values of x and thereby yields the desired distribution of values in node 10 of the tree $D_{q_2^{10}}$ by weighting every possible value in the distribution in node 1 with the respective probability of the input signal staying within the deadband around it.

Note that in Equation 4.11 integration is done over the PDF of the input signal D_{x_2} whereas the output distribution $D_{q_1^1}$ is used directly. This is the case because the input signal does not violate the deadband bound and is therefore discarded and only the probability of it staying within the bound is of interest. In consequence we need the distribution of values which pertain through this step, namely the output values after step one, and modify it by the probabilities that no bound violation takes place.

Of course, the probability of no bound violation in step two of the signal is then

$$P_{q_2^{10}} = \int_{-\infty}^{+\infty} D_{q_2^{10}}(x)dx \tag{4.12}$$

making it easy to calculate the probability of bound violation:

$$P_{q_2^{11}} = 1 - P_{q_2^{10}} \tag{4.13}$$

Unfortunately this is not enough to be able to propagate further down the binary tree. We need the exact partial PDF at node 11 of the tree. Its calculation is a bit more complicated than in Equation 4.11:

$$D_{q_2^{11}}(x) = D_{x_2}(x)\left(P_{q_1^1} - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} D_{q_1^1}(y)dy\right) \tag{4.14}$$

The integral over the output distribution $D_{q_1^1}$ yields the probability that a given new input value x lies within any possible deadband around any last output value. The integral limits are determined as follows:

$$\begin{aligned} q - pq &< x < q + pq \\ q(1-p) &< x < q(1+p) \\ (1-p) &< \tfrac{x}{q} < (1+p) \\ \frac{1}{(1-p)} &> \tfrac{q}{x} > \frac{1}{(1+p)} \\ \frac{x}{(1-p)} &> q > \frac{x}{(1+p)} \end{aligned} \tag{4.15, 4.16}$$

So when determining the probability that some new input value lies within the bound around a fixed output signal we need Equation 4.15 as used in Equation 4.11. Vice versa, when determining the probability of all output values that generate a deadband bound where a fixed input value lies within, we need Equation 4.16, as applied in Equation 4.14.

Since we want to know the probability that a new input value x lies outside of the deadband bound of any possible preceding output value, we need the complementary probability which is represented by the term in large brackets. This term therefore represents the probability that a given new input value becomes the new output and reference value. Multiplying this term by the probability of occurrence of every input value yields the desired distribution.

Note that this time integration is done over the output distribution $D_{q_1^1}$ whereas the input distribution D_{x_2} is used directly. In contrast to Equation 4.11 every value in the input distribution is multiplied by the probability that this value violates the bound around any possible output value.

At first glance one would think that the straightforward way of calculating $D_{q_2^{11}}$ would be

$$D(x) = D_{q_1^1}(x)\left(1 - \int_{x(1-p)}^{x(1+p)} D_{x_2}(y)dy\right) \tag{4.17}$$

but this would yield the distribution of output values which generate a deadband bound which is violated by the new input value. Therefore these new incoming input values replace the old output values and become the new reference values. The distribution of replaced old output values is of no interest to us. Consequently, this does not yield the correct distribution of the respective tree node. The integral over this distribution still yields the correct probability simply because it is the complementary case to Equation 4.14.

The straightforward way to propagate further down the tree would now be to use Equations 4.11 and 4.14 on the respective output distribution of a tree node to calculate the distributions of its two successors down the tree. The generalized forms of these equations are:

$$D_{q_{i+1}^{t0}}(x) = D_{q_i^t}(x) \int_{x(1-p)}^{x(1+p)} D_{x_{i+1}}(y)dy \tag{4.18}$$

and

$$D_{q_{i+1}^{t1}}(x) = D_{x_{i+1}}(x) \left(P_{q_i^t} - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} D_{q_i^t}(y)dy \right) \tag{4.19}$$

Using these equations to generate the binary tree allows us to calculate the steady state probability for iid signals. Since the probability of threshold violation converges to the steady state probability after five to seven steps in the signal the according five to seven tree levels have to be generated to calculate P_{SS}.

The generation of the whole binary tree structure is necessary in general and is the solution for any signal model. Of course the above equations only work for the iid signal case but tree generation methods for other signal models are presented later on. However the iid signal case offers a means of simplification because of its independent signal behavior.

Since Equations 4.18 and 4.19 have to be used on every node of a tree level in order to get to the next it would be useful if we could use them on a sum of different nodes to keep the amount of computations low. By taking a look at Equation 4.18, we can see that the integral part is independent from the position in the tree because the distribution of the input signal D_{x_i} is always the same in the different tree nodes of one step and in this iid case even over all steps since the input signal is independent. Therefore all output distributions in one tree level can be combined into one distribution by summing them all up. By doing so already in the second tree level, we combine the outcome of the deadband violation

distribution and the deadband compliance distribution into one output distribution for the whole tree level. Two nodes can be combined like this

$$D_{q_1^0}(x) \int_{x(1-p)}^{x(1+p)} D_{x_2}(y)dy + D_{q_1^1}(x) \int_{x(1-p)}^{x(1+p)} D_{x_2}(y)dy = \left(D_{q_1^0}(x) + D_{q_1^1}(x)\right) \int_{x(1-p)}^{x(1+p)} D_{x_2}(y)dy \tag{4.20}$$

Therefore the equation to get to the next tree level would then be

$$D_{q_{i+1}^0}(x) = \left(D_{q_i^0}(x) + D_{q_i^1}(x)\right) \int_{x(1-p)}^{x(1+p)} D_x(y)dy \tag{4.21}$$

where q_i^t with t being only 0 or 1 means the sum of all node distributions at tree level i with 0 or 1 at the end of the superscript respectively.

Of course in order to fully incorporate this simplification it ought to be possible for Equation 4.19 also. Since when combining all output nodes of one tree level $P_{q_i^t}$ will become 1 and again the input distribution in one tree level is always the same, the calculation can be simplified as follows:

$$\begin{aligned} D_{x_2}(x) \left(P_{q_1^0} - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} D_{q_1^0}(y)dy \right) + D_{x_2}(x) \left(P_{q_1^1} - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} D_{q_1^1}(y)dy \right) = \\ = D_{x_2}(x) \left(1 - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} \left(D_{q_1^0}(y) + D_{q_1^1}(y)\right) dy \right) \end{aligned} \tag{4.22}$$

resulting in the following equation to get to the next step in the signal:

$$D_{q_{i+1}^1}(x) = D_x(x) \left(1 - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} \left(D_{q_i^0}(y) + D_{q_i^1}(y)\right) dy \right) \tag{4.23}$$

This of course makes it very easy to get down the tree, since we only have to evaluate Equations 4.21 and 4.23 once per tree level to generate the distributions for deadband violation and compliance and therefore the probabilities of violation and compliance as the integrals over those distributions.

Two special cases of iid input signals are presented in the following.

4.3.1 Gaussian Input Sequence

4.3.1.1 Signal characteristics

We consider a normally distributed input sequence with zero mean and variance 1, i.e.,

$$x_i \in \mathcal{N}(0, \sigma^2 = 1) \tag{4.24}$$

The PDF of the input signal is then:

$$D_{x_i}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{4.25}$$

4.3.1.2 First tree levels

The calculation of node probabilities down the tree looks as follows. Naturally, q_1^1 is normally distributed because it just assumes the input distribution:

$$D_{q_1^1}(x) = D_{x_1}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{4.26}$$

Following the principles from above, the partial PDF of q_2^{10} can be calculated as

$$\begin{aligned} D_{q_2^{10}}(x) &= D_{q_1^1}(x) \int_{x(1-p)}^{x(1+p)} D_{x_2}(y)dy = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \int_{x(1-p)}^{x(1+p)} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy = \\ &= \frac{1}{\sqrt{8\pi}} e^{-\frac{x^2}{2}} \left(erf(\frac{x(1+p)}{\sqrt{2}}) - erf(\frac{x(1-p)}{\sqrt{2}}) \right) \end{aligned} \tag{4.27}$$

This is the distribution of all q_1^1 which are not replaced as reference value by a new x_2 and therefore become q_2^{10}. Thus, to determine the probability that x_2 lies within the deadband bound of q_1^1 we have to integrate over the partial PDF. Because of its symmetrical distribution around 0, the equation can be written as:

$$\begin{aligned} P_{q_2^{10}} &= 2 \int_0^{+\infty} \frac{1}{\sqrt{8\pi}} e^{-\frac{x^2}{2}} \left(erf(\frac{x(1+p)}{\sqrt{2}}) - erf(\frac{x(1-p)}{\sqrt{2}}) \right) dx = \\ &= \frac{1}{\sqrt{\pi}} \left(\arctan(1+p) - \arctan(1-p) \right) \end{aligned} \tag{4.28}$$

The solution is obtained using the Taylor-series for the standard error function (erf). It can also be solved using the Cumulative Distribution Function (CDF) of the Cauchy-Distribution, as we basically look at the ratio of two standard normally distributed variables. Unfortunately, this is the only probability which can be given in closed form. Further down the tree the distributions are no longer standard Gaussian. Fortunately, the probability of q_2^{11} (when x_2 violates the deadband bound around q_1^1) can be easily calculated:

$$P_{q_2^{11}} = 1 - P_{q_2^{10}} \tag{4.29}$$

As already shown in Equation 4.14, the PDF of q_2^{10} can not be calculated that easily. Since x_2 replaces q_1^1 as the reference value we are not interested in the distribution of all q_1^1 which are replaced as reference value by a new x_2. Instead we need the distribution of all possible values of x_2 that violate the deadband bound to be able to go further down the tree. For positive values of x and q the distribution can be calculated as follows:

$$\begin{aligned} D_{q_2^{11}}(x) &= D_{x_2}(x)\left(1 - \int_{\frac{x}{(1+p)}}^{\frac{x}{(1-p)}} D_{q_1^1}(y)dy\right) = \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \left(1 - \frac{1}{2}\left(erf(\frac{x}{(1-p)\sqrt{2}}) - erf(\frac{x}{(1+p)\sqrt{2}})\right)\right) \end{aligned} \tag{4.30}$$

We now know the distributions of all the tree nodes after the first two steps in the signal sequence. Unfortunately, the partial PDFs become more complex the further we go down the tree. The partial PDFs of the output values in step three of the input sequence x_3 would be:

$$D_{q_3^{100}}(x) = D_{q_2^{10}}(x) \int_{x(1-p)}^{x(1+p)} D_{x_3}(y)dy \tag{4.31}$$

$$D_{q_3^{101}}(x) = D_{x_3}(x)\left(P_{q_2^{10}} - \int_{\frac{x}{(1+p)}}^{\frac{x}{(1-p)}} D_{q_2^{10}}(y)dy\right) \tag{4.32}$$

$$D_{q_3^{110}}(x) = D_{q_2^{11}}(x) \int_{x(1-p)}^{x(1+p)} D_{x_3}(y)dy \tag{4.33}$$

$$D_{q_3^{111}}(x) = D_{x_3}(x)\left(P_{q_2^{11}} - \int_{\frac{x}{(1+p)}}^{\frac{x}{(1-p)}} D_{q_2^{11}}(y)dy\right) \tag{4.34}$$

Since the simplification for iid signals shown in Equations 4.21 and 4.23 can be used here, the calculation of those four node distributions is not necessary. Still the problem remains that calculating the partial PDFs further down the tree is no longer possible in a closed form solution. We therefore numerically solve the integrals to calculate the partial PDFs of all nodes in the binary tree up to the desired tree level for both the general and the simplified case.

4.3.1.3 Validation and Steady State Probability

As shown previously, even for a simple signal model, the analytical solution for the update rate of the deadband sampler becomes rather involved. To show that this analysis holds, we determined the partial PDFs of the nodes in the binary tree for the first three levels by both simulation over 10000000 sample signals and by numerical solution of the aforementioned formulae. We show both the correctness of the general approach of calculating the whole tree (Equations 4.18 and 4.19) and the simplified approach (Equations 4.21 and 4.23). As you can see in Figure 4.5, the grey signal represents the simulation, and the black line is the numerical solution. Numerical solutions and the general numeric approach are in very good agreement (in this case for $p = 0.8$). In Figure 4.6 the simplified approach is shown where only two partial PDFs have to be calculated per tree level (Step 10 and Step 11). These two partial PDFs are then summed up to generate the new output PDF (Step 2). From this PDF, two new partial PDFs are generated (Step 20 and 21) which are again summed up (Step 3) and so on.

As stated above, we can calculate an estimate of the steady state probability P_{SS} by adding up all probabilities for a deadband bound violation at one level of the binary tree. Using the simplified approach enables us to go down the tree very quickly and calculate the violation probabilities very accurately. As we can see in Figure 4.7 level two in the tree (sample two of the input signal) is enough for $p = 0.1$ to $p = 0.4$. For $p = 0.5$ to $p = 0.7$ level three would be a good approximation and for $p = 0.8$ to $p = 0.9$ level four, five or even six would be necessary.

As mentioned earlier, the steady state probability P_{SS} directly represents the relative update rate at the output of the deadband approach. Figure 4.7 also shows the analytically derived P_{SS} for values of p from 0.1 to 0.9 for the first 10 steps in the input sequence using the simplified approach and the corresponding simulation results. We can see an almost perfect agreement for all probabilities in the Figure. The steady state probability is reached later in the sequence for higher p. For the case of $p = 0.3$ in Figure 4.7, an update rate of 90% or equivalently a rate reduction of 10% can be observed.

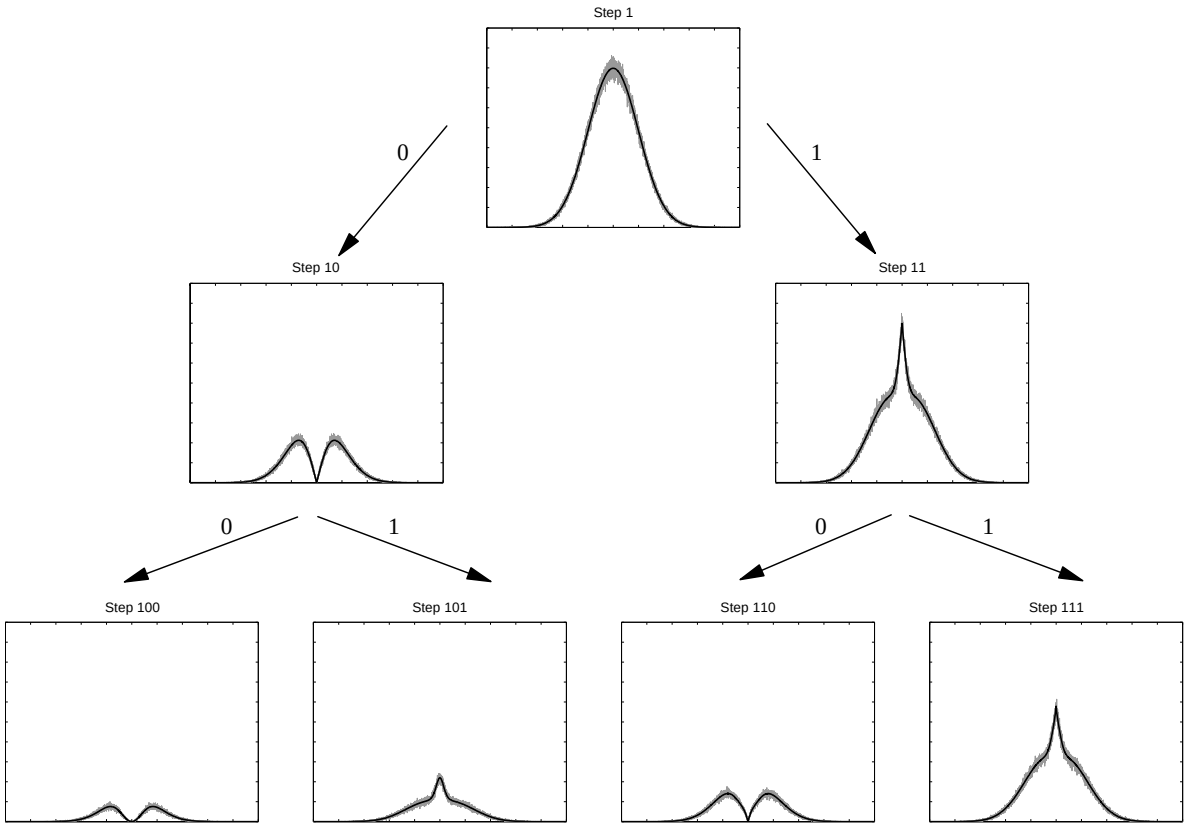

Figure 4.5: Simulation (grey signal) and analytical derivation (black line) of the PDFs of q for the first three tree levels of the input sequence for $p = 0.8$. All plots have the same scale.

4.3.2 Uniform Input Sequence

4.3.2.1 Signal characteristics

In this section an input signal with uniform distribution from -1 to 1 is analyzed.

$$x_i \in \mathcal{U}(-1, 1) \tag{4.35}$$

The PDF of the signal is then:

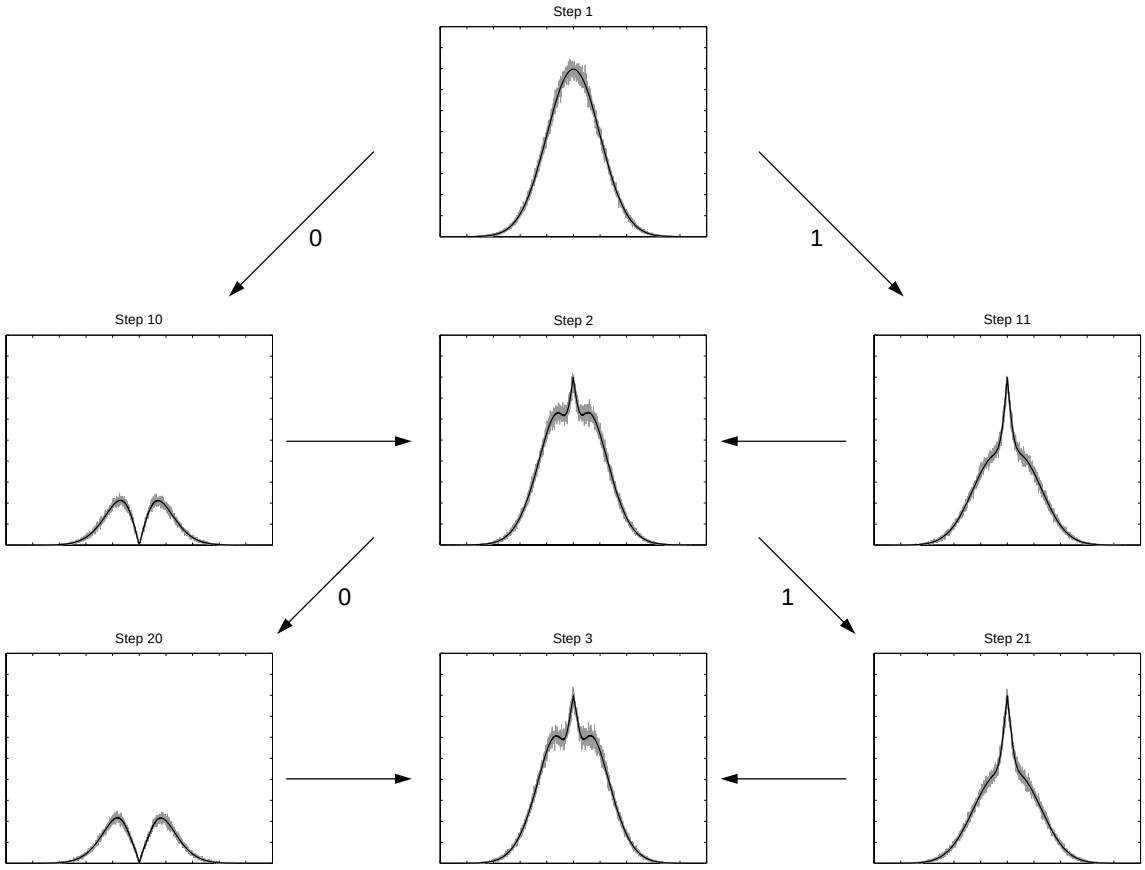

Figure 4.6: Simulation (grey signal) and analytical derivation (black line) of the PDFs of q for the first three tree levels using the simplified approach for iid signals and $p = 0.8$. All plots have the same scale.

$$D_{x_i}(x) = \begin{cases} 0 & : \quad x < -1 \\ \frac{1}{2} & : \quad -1 \leq x \leq 1 \\ 0 & : \quad 1 < x \end{cases} \tag{4.36}$$

4.3.2.2 First tree levels

In the following, the first three tree levels are calculated in closed form in order to prove that the theoretical tree generation formulae are correct. The uniform signal model allows for simple integration steps which led to problems in the uniform case. The downside are discontinuities in the partial PDFs because of the non-continuous nature of the input PDF. In order not to introduce additional discontinuities the general tree generation approach is

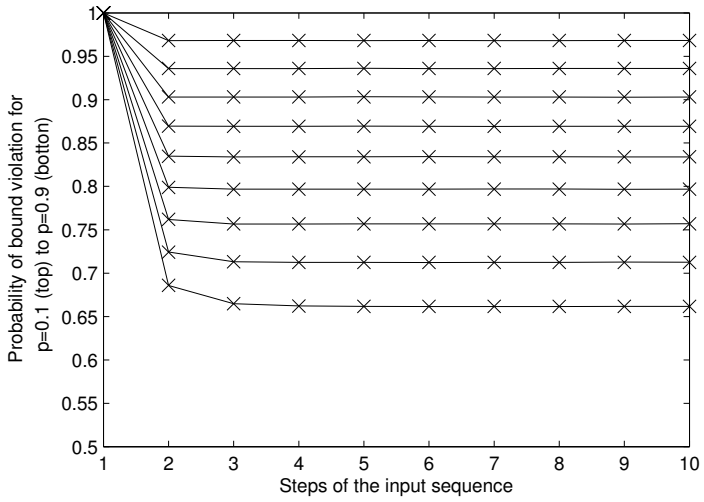

Figure 4.7: Simulation (black lines) and analytical derivation (black crosses) of the steady state probability (the average probability that a new input value violates the active deadband bound) for $p = 0.1$ (top) to $p = 0.9$ (bottom) for the first ten samples of the input signal calculated using the simplified approach for iid signals.

used here and not the simplified version. It is getting complicated enough in the general case and would be very hard to follow in the simplified approach.

The first level of the tree with only one node (q_1^1) is very easy to calculate, because the first sample in the signal always violates the current deadband bound. Therefore the distribution of q_1^1 equals the distribution of the input signal.

$$D_{q_1^1}(x) = D_{x_1}(x) = \begin{cases} 0 & : \quad x < -1 \\ \frac{1}{2} & : \quad -1 \leq x \leq 1 \\ 0 & : \quad 1 < x \end{cases} \tag{4.37}$$

The second level of the tree consists of the nodes q_2^{10} and q_2^{11}. First we explain the calculation of $D_{q_2^{10}}$ which can be done according to Equation 4.11. Because all distributions in this signal model case are symmetric around zero we only calculate the positive part of the distributions. The non-continuous nature of the input signal distribution requires different intervals in which the integrals are evaluated. In this first case, the integration limits are $x(1-p)$ and $x(1+p)$ in which we need to evaluate the integral over the input signal. Since we are only interested in values of p from 0 to 1, we must evaluate if the interval limits conflict with the borders of the input and output distributions. The lower limit $x(1-p)$ becomes x in the worst case that $p = 1$ and stays below x for $0 < p < 1$, therefore this is no problem in any case. The upper limit $x(1+p)$ becomes greater than 1 and therefore

exceeds past the discontinuity of the distribution if the following is true:

$$\begin{aligned} x(1+p) &> 1 \\ x &> \frac{1}{1+p} \end{aligned}$$

So we need to evaluate the integral for two different cases:

For $x < \frac{1}{1+p}$:

$$D^1_{q_2^{10}}(x) = D_{q_1^1}(x) \int_{x(1-p)}^{x(1+p)} D_{x_2}(y)dy = \frac{1}{2} \int_{x(1-p)}^{x(1+p)} \frac{1}{2} dy = \frac{1}{2}xp \tag{4.38}$$

and for $x > \frac{1}{1+p}$:

$$D^2_{q_2^{10}}(x) = D_{q_1^1}(x) \int_{x(1-p)}^{1} D_{x_2}(y)dy = \frac{1}{2} \int_{x(1-p)}^{1} \frac{1}{2} dy = \frac{1}{4} + x\left(\frac{p-1}{4}\right) \tag{4.39}$$

The calculation of $D_{q_2^{11}}$ is similar according to Equation 4.14. The only difference is the integral limits to be checked for violations. The lower integral limit $\frac{x}{1+p}$ is also no problem, because it never exceeds x for our given range of p. The upper integral limit $\frac{x}{1-p}$ becomes greater than 1 in the case:

$$\begin{aligned} \frac{x}{1-p} &> 1 \\ x &> 1-p \end{aligned}$$

So the two interval cases are:

For $x < 1 - p$:

$$D^1_{q_2^{11}}(x) = D_{x_2}(x)\left(1 - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} D_{q_1^1}(y)dy\right) = \frac{1}{2}\left(1 - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} \frac{1}{2} dy\right) = \frac{1}{2} - x\left(\frac{p}{2(1-p^2)}\right) \tag{4.40}$$

And for $x > 1 - p$:

$$D^2_{q_2^{11}}(x) = D_{x_2}(x)\left(1 - \int\limits_{\frac{x}{1+p}}^{1} D_{q_1^1}(y)dy\right) = \frac{1}{2}\left(1 - \int\limits_{\frac{x}{1+p}}^{1} \frac{1}{2}dy\right) = \frac{1}{4} + x\left(\frac{1}{4(p+1)}\right) \quad (4.41)$$

Since we can analytically derive the closed form solution of these integrals there is no need to use numerical approaches. The simulations shown in Figure 4.8 were done to immediately confirm the correctness of the calculations. We can see a very good match between simulation and analytical derivation. The intersection of the black lines marks the spot where the two integrals meet.

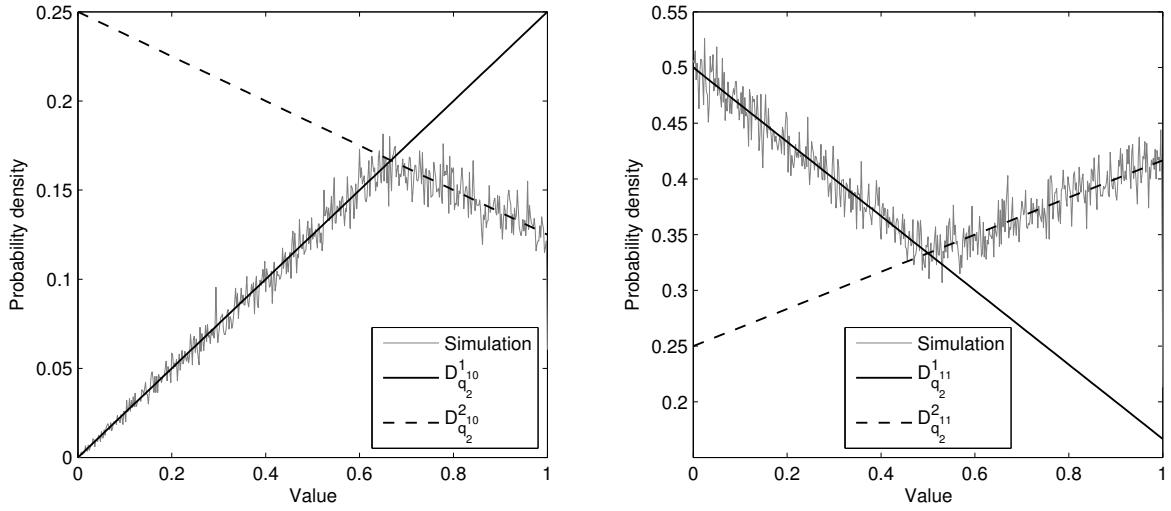

Figure 4.8: Simulation (grey) and analytical derivation (black lines) of the PDFs of q for the second tree level of a uniform input sequence for $p = 0.5$.

Since we need the probabilities of the tree nodes in the second level to propagate to the third level of the tree, we have to calculate:

$$\begin{aligned} P_{q_2^{10}} &= 2\int\limits_0^1 D_{q_2^{10}}(y)dy = 2\left(\int\limits_0^{\frac{1}{1+p}} D^1_{q_2^{10}}(y)dy + \int\limits_{\frac{1}{1+p}}^{1} D^2_{q_2^{10}}(y)dy\right) = \\ &= 2\left(\int\limits_0^{\frac{1}{1+p}} \frac{1}{2}ypdy + \int\limits_{\frac{1}{1+p}}^{1} \frac{1}{4} + y\left(\frac{p-1}{4}\right)dy\right) = \frac{p^2+2p}{4(1+p)} \end{aligned} \quad (4.42)$$

$$P_{q_2^{11}} = 2\int_0^1 D_{q_2^{11}}(y)dy = 2\left(\int_0^{1-p} D^1_{q_2^{11}}(y)dy + \int_{1-p}^1 D^2_{q_2^{11}}(y)dy\right) = \tag{4.43}$$
$$= 2\left(\int_0^{1-p}\frac{1}{2} - y\left(\frac{p}{2(1-p^2)}\right)dy + \int_{1-p}^1 \frac{1}{4} + x\left(\frac{1}{4(p+1)}\right)dy\right) = \frac{-p^2+2p+4}{4(1+p)}$$

Of course it is also possible to evaluate

$$P_{q_2^{11}} = 1 - P_{q_2^{10}} \tag{4.44}$$

as an easier way to get $P_{q_2^{11}}$.

In the third level of the tree we have to determine four different distribution functions. We begin with $D_{q_3^{100}}(x)$, which is determined as follows:

$$D_{q_3^{100}}(x) = D_{q_2^{10}}(x)\int_{x(1-p)}^{x(1+p)} D_{x_3}(y)dy \tag{4.45}$$

Since $D_{q_2^{10}}(x)$ has a discontinuity in $\frac{1}{1+p}$ the resulting distribution will also consist of at least two parts. The discontinuity of $D_{x_3}(y)$ is at 1, but if $x = \frac{1}{1+p}$ the upper limit of the integral will be 1 as well. So the whole integral consists of only two continuous parts:

For $0 < x < \frac{1}{1+p}$:

$$D^1_{q_3^{100}}(x) = D^1_{q_2^{10}}(x)\int_{x(1-p)}^{x(1+p)} D_{x_3}(y)dy = \frac{1}{2}xp\int_{x(1-p)}^{x(1+p)}\frac{1}{2}dy = \frac{1}{2}x^2p^2 \tag{4.46}$$

and for $\frac{1}{1+p} < x < 1$:

$$D^2_{q_3^{100}}(x) = D^2_{q_2^{10}}(x)\int_{x(1-p)}^{1} D_{x_3}(y)dy = \left(\frac{1}{4} + x\left(\frac{p-1}{4}\right)\right)\int_{x(1-p)}^{1}\frac{1}{2}dy = x^2\frac{(1-p)^2}{8} - x\frac{1-p}{4} + \frac{1}{8} \tag{4.47}$$

Unfortunately, the distribution of $D_{q_3^{101}}(x)$ is more complicated. It is determined as:

$$D_{q_3^{101}}(x) = D_{x_3}(x)\left(P_{q_3^{10}} - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} D_{q_2^{10}}(y)dy\right) \tag{4.48}$$

Because the integrand $D_{q_2^{10}}(y)$ has discontinuities at $\frac{1}{1+p}$ and 1 we again need to split the integral into different parts. First, we need to know for which x the upper limit of the integral reaches the first discontinuity:

$$\frac{x}{1-p} = \frac{1}{1+p} \quad \Longrightarrow \quad x = \frac{1-p}{1+p} \tag{4.49}$$

Second, we need to do the same for the second discontinuity:

$$\frac{x}{1-p} = 1 \quad \Longrightarrow \quad x = 1-p \tag{4.50}$$

And third, we need to check if the lower limit of the integral violates the first discontinuity:

$$\frac{x}{1+p} = \frac{1}{1+p} \quad \Longrightarrow \quad x = 1 \tag{4.51}$$

This means that it reaches the first discontinuity for $x = 1$, but since x doesn't get any higher this doesn't matter.

Therefore the integral consists of three parts which are the following:

For $0 < x < \frac{1-p}{1+p}$:

$$D^1_{q_3^{101}}(x) = D_{x_3}(x)\left(P_{q_2^{10}} - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} D^1_{q_2^{10}}(y)dy\right) = -x^2\frac{p^2}{2(1-p^2)^2} + \frac{p^2+2p}{8(1+p)} \tag{4.52}$$

For $\frac{1-p}{1+p} < x < 1-p$:

$$\begin{aligned} D^2_{q_3^{101}}(x) &= D_{x_3}(x)\left(P_{q_2^{10}} - \left(\int_{\frac{x}{1+p}}^{\frac{1}{1+p}} D^1_{q_2^{10}}(y)dy + \int_{\frac{1}{1+p}}^{\frac{x}{1-p}} D^2_{q_2^{10}}(y)dy\right)\right) = \\ &= -x^2\frac{p^2-4p-1}{16(1-p)(1+p)^2} - x\frac{1}{8(1-p)} + \frac{2p^3+6p^2+5p+1}{16(1+p)^2} \end{aligned} \tag{4.53}$$

And for $1-p < x < 1$:

$$D^3_{q_3^{101}}(x) = D_{x_3}(x)\left(P_{q_2^{10}} - \left(\int_{\frac{x}{1+p}}^{\frac{1}{1+p}} D^1_{q_2^{10}}(y)dy + \int_{\frac{1}{1+p}}^{1} D^2_{q_2^{10}}(y)dy\right)\right) =$$
$$= x^2\frac{p}{8(1+p)^2} + \frac{p^3+3p^2+2p}{16(1+p)^2} \tag{4.54}$$

To proof the correctness of these results so far a simulation run was conducted. The results are shown in Figure 4.9. It is clearly shown how the different functions match the simulation in their respective interval of validity.

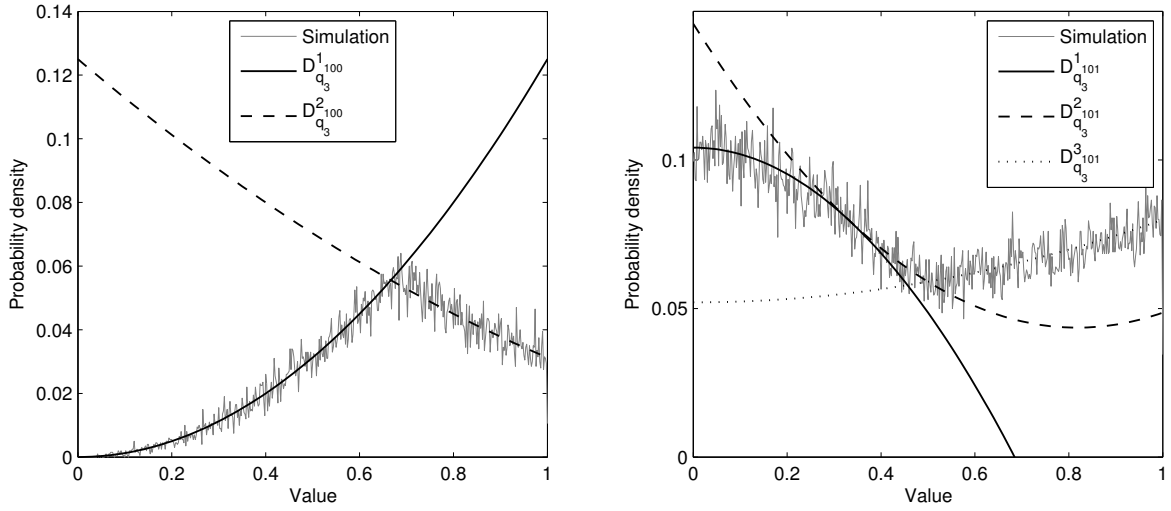

Figure 4.9: Simulation (grey) and analytical derivation (black lines) of the PDFs of q_3^{100} (left) and q_3^{101} (right) in the third tree level of a uniform input sequence for $p = 0.5$.

As the next step, we calculate the distributions which originate from node q_2^{11} in the second level of the tree. We begin with $D_{q_3^{110}}(x)$, which is determined as follows:

$$D_{q_3^{110}}(x) = D_{q_2^{11}}(x)\int_{x(1-p)}^{x(1+p)} D_{x_3}(y)dy \tag{4.55}$$

Again, we need to take a look at the discontinuities. The integrand has its only discontinuity at 1. $D_{q_2^{11}}(x)$ has a discontinuity at $1-p$. For $x < 1-p$ the integral stays within continuous limits. Therefore the first part of $D_{q_3^{110}}(x)$ lies in the interval $0 < x < 1-p$. The second discontinuity of $D_{q_3^{110}}(x)$ takes place where the integrand reaches its discontinuity,

which is at $\frac{1}{1+p}$ as already determined earlier. This leaves us with two discontinuities in the resulting distribution and therefore three continuous parts, which are:

For $0 < x < 1 - p$:

$$D^1_{q_3^{110}}(x) = D^1_{q_2^{11}}(x) \int_{x(1-p)}^{x(1+p)} D_{x_3}(y)dy = -x^2 \frac{p^2}{2(1-p^2)} + x\frac{p}{2} \tag{4.56}$$

For $1 - p < x < \frac{1}{1+p}$:

$$D^2_{q_3^{110}}(x) = D^2_{q_2^{11}}(x) \int_{x(1-p)}^{x(1+p)} D_{x_3}(y)dy = x^2 \frac{p}{4(1+p)} + x\frac{p}{4} \tag{4.57}$$

And for $\frac{1}{1+p} < x < 1$:

$$D^3_{q_3^{110}}(x) = D^2_{q_2^{11}}(x) \int_{x(1-p)}^{1} D_{x_3}(y)dy = -x^2 \frac{1-p}{8(1+p)} + x\frac{p^2}{8(1+p)} + \frac{1}{8} \tag{4.58}$$

Finally we take a look at the distribution of q_3^{111} which can be determined as:

$$D_{q_3^{111}}(x) = D_{x_3}(x) \left(P_{q_3^{11}} - \int_{\frac{x}{1+p}}^{\frac{x}{1-p}} D_{q_2^{11}}(y)dy \right) \tag{4.59}$$

And again we need to determine the discontinuities. $D_{x_3}(x)$ has no discontinuities in the evaluated range from 0 to 1 but $D_{q_2^{11}}(y)$ has two at $1 - p$ and 1. The upper integral limit reaches $1 - p$ when

$$\frac{x}{1-p} = 1 - p \quad \Longrightarrow \quad x = (1-p)^2 \tag{4.60}$$

The second discontinuity takes place when the upper integral limit reaches 1:

$$\frac{x}{1-p} = 1 \quad \Longrightarrow \quad x = 1 - p \tag{4.61}$$

And again, we need to check if the lower limit of the integral violates the first discontinuity:

$$\frac{x}{1+p} = 1-p \quad \Longrightarrow \quad x = 1-p^2 \tag{4.62}$$

This means that the lower limit does violate the discontinuity and therefore there is a third discontinuity of the resulting distribution.

The four parts of the distribution are:

For $0 < x < (1-p)^2$:

$$\begin{aligned} D^1_{q_3^{111}}(x) &= D_{x_3}(x)\left(P_{q_2^{11}} - \int\limits_{\frac{x}{1+p}}^{\frac{x}{1-p}} D^1_{q_2^{11}}(y)dy\right) = \\ &= x^2\frac{2p^2}{(1-p^2)^3} - x\frac{2p}{(1-p^2)} - \frac{p^2-2p-4}{8(1+p)} \end{aligned} \tag{4.63}$$

For $(1-p)^2 < x < 1-p$:

$$\begin{aligned} D^2_{q_3^{111}}(x) &= D_{x_3}(x)\left(P_{q_2^{11}} - \left(\int\limits_{\frac{x}{1+p}}^{1-p} D^1_{q_2^{11}}(y)dy + \int\limits_{1-p}^{\frac{x}{1-p}} D^2_{q_2^{11}}(y)dy\right)\right) = \\ &= x^2\frac{p^2-4p-1}{16(1-p^2)^2(1+p)} + x\frac{1-3p}{8(1-p^2)} - \frac{p^2-4p-7}{16(1+p)} \end{aligned} \tag{4.64}$$

For $1-p < x < 1-p^2$:

$$\begin{aligned} D^3_{q_3^{111}}(x) &= D_{x_3}(x)\left(P_{q_2^{11}} - \left(\int\limits_{\frac{x}{1+p}}^{1-p} D^1_{q_2^{11}}(y)dy + \int\limits_{1-p}^{1} D^2_{q_2^{11}}(y)dy\right)\right) = \\ &= -x^2\frac{p}{8(1+p)^2(1-p^2)} + x\frac{1}{4(1+p)} - \frac{p^2-2p-4}{16(1+p)} \end{aligned} \tag{4.65}$$

And for $1-p^2 < x < 1$:

$$D^4_{q_3^{111}}(x) = D_{x_3}(x)\left(P_{q_2^{11}} - \int\limits_{\frac{x}{1+p}}^{1} D^2_{q_2^{11}}(y)dy\right) =$$

$$= x^2 \frac{1}{16(1+p)^3} + x\frac{1}{8(1+p)} - \frac{2p^2-2p-5}{16(1+p)} \tag{4.66}$$

In order to confirm the results for the last two distributions another simulation run was conducted. The results are shown in Figure 4.10. We again see a very good agreement of the simulated distribution and the analytical derivations in their respective intervals.

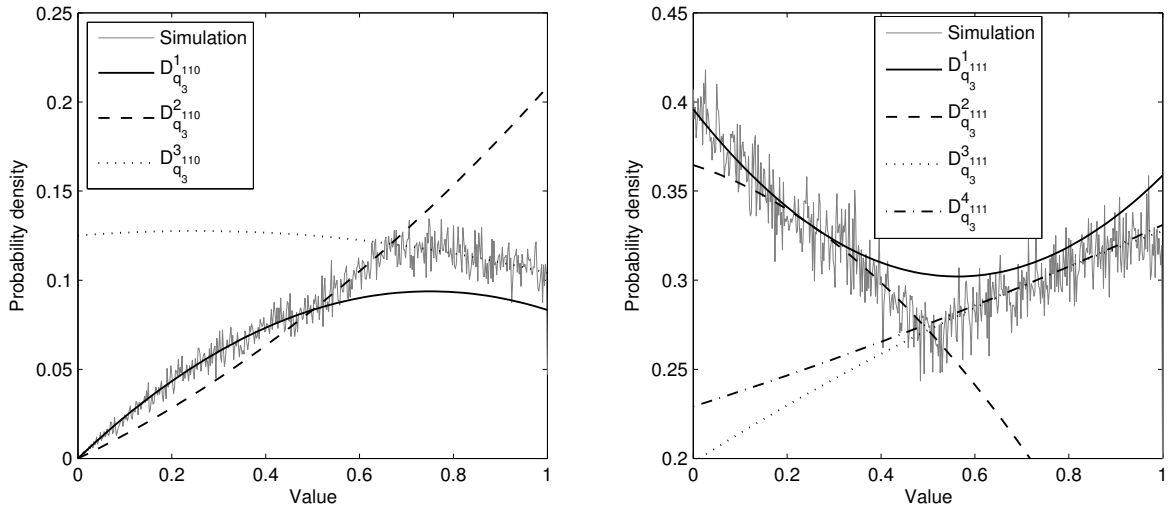

Figure 4.10: Simulation (grey) and analytical derivation (black lines) of the PDFs of q_3^{110} (left) and q_3^{111} (right) in the third tree level of a uniform input sequence for $p = 0.5$.

4.3.2.3 Steady state probability

Since we analytically derived the tree node probabilities for the first three levels of the binary tree we are now able to calculate the approximate steady state probability P_{SS} for uniform input signals in these levels. We already calculated $P_{q_2^{10}}$ and $P_{q_2^{11}}$ earlier which are the approximated steady state probabilities for bound violation ($P_{q_2^{11}}$) and no bound violation ($P_{q_2^{10}}$) in the second level of the tree. In order to get the approximated steady state probability for bound violation in the third tree level, we need to sum up all probabilities in which a bound violation took place, in this case $P_{q_3^{101}}$ and $P_{q_3^{111}}$. To calculate those two probabilities, we have to integrate over the respective partly continuous distributions. As we have seen above $D_{q_3^{101}}$ consists of 3 continuous parts and $P_{q_3^{111}}$ of even four. On the other hand, $D_{q_3^{100}}$ consists of only two parts and $D_{q_3^{110}}$ of three parts. Therefore the easier way to calculate P_{SS_3} would be:

$$P_{SS_3} = 1 - \left(P_{q_3^{100}} + P_{q_3^{110}}\right) \tag{4.67}$$

So we calculate the probabilities $P_{q_3^{100}}$ and $P_{q_3^{110}}$:

$$\begin{aligned} P_{q_3^{100}} &= 2\int_0^1 D_{q_3^{100}}(y)dy = 2\left(\int_0^{\frac{1}{1+p}} D^1_{q_3^{100}}(y)dy + \int_{\frac{1}{1+p}}^1 D^2_{q_3^{100}}(y)dy\right) = \\ &= \frac{p^4+3p^3+4p^2}{12(1+p)^2} \end{aligned} \tag{4.68}$$

$$\begin{aligned} P_{q_3^{110}} &= 2\int_0^1 D_{q_3^{110}}(y)dy = 2\left(\int_0^{1-p} D^1_{q_3^{110}}(y)dy + \int_{1-p}^{\frac{1}{1+p}} D^2_{q_3^{110}}(y)dy + \int_{\frac{1}{1+p}}^1 D^3_{q_3^{110}}(y)dy\right) = \\ &= \frac{2p^6+2p^5-p^4+10p^3+22p^2+12p}{24(1+p)^3} \end{aligned} \tag{4.69}$$

Finally, the approximate steady state probability in tree level three is:

$$\begin{aligned} P_{SS_3} &= 1-\left(P_{q_3^{100}}+P_{q_3^{110}}\right) = 1-\frac{2p^6+4p^5+7p^4+24p^3+30p^2+12p}{24(1+p)^3} \\ &= \frac{24+60p+42p^2-7p^4-4p^5-2p^6}{24(1+p)^3} \end{aligned} \tag{4.70}$$

In Figure 4.11 the simulation run for the steady state probability is shown. The crosses mark the probabilities obtained using above formulas. We see an excellent match of simulation and analytical derivation in the first three tree levels.

Since every new step in the signal leads to additional discontinuities in the signal, we stop our analytical derivation at step 3 of the signal. While the methodology stays the same for later steps of the signal, the actual calculation becomes rather involved very quickly.

4.4 Autoregressive Input Sequence

4.4.1 Signal Characteristics

In this section an autoregressive input signal with normally distributed increment is analyzed. The signal model can be described as:

$$x_{i+1} = ax_i + \eta \tag{4.71}$$

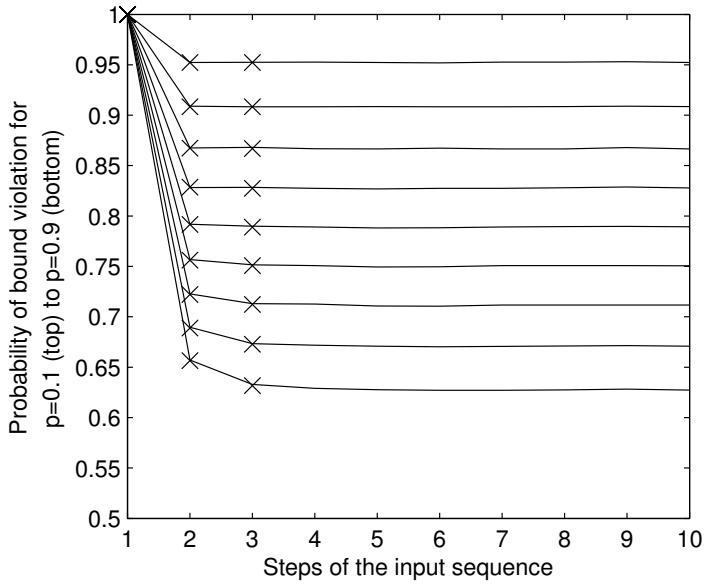

Figure 4.11: Simulation (black lines) and analytical derivation (black crosses) of the steady state probability (the average probability that a new input value violates the active deadband bound) for $p = 0.1$ (top) to $p = 0.9$ (bottom) for the first ten steps of a uniformly distributed input signal.

where η ist normally distributed with zero mean and variance 1 or in short:

$$\eta \in \mathcal{N}(0, 1) \tag{4.72}$$

with the distribution

$$D_\eta(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{4.73}$$

a is the autoregression factor which in our case is varied from 0 to 1. With $a = 0$ this signal model becomes equivalent to a normally distributed input sequence as already discussed in Section 4.3.1. For $a = 1$ the input sequence becomes the so called random walk model, also called the Wiener Process.

In this signal model the values in the input sequence are no longer independent from each other as in the iid cases. Also the input distributions change over time in the input sequence. Since this represents a far more realistic behavior for haptic signals, the autoregressive input sequence is certainly a step in the direction of more realistic signal models. Of course, this also makes the analytical derivation far more complex as well.

4.4.2 Tree Generation Algorithm

As already observed before for other signal models, the first tree node or root node q_1^1 simply assumes the input distribution of x_1, if we assume $x_0 = 0$ as in all previous cases.

$$D_{q_1^1}(x) = D_{x_1}(x) = D_\eta(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{4.74}$$

In the second tree level, the two node distributions can be calculated similarly to the iid-gauss sequence, but with incorporation of the autoregression factor a. The equation for deadband compliance is:

$$D_{q_2^{10}}(x) = D_{q_1^1}(x) \int_{x(1-p-a)}^{x(1+p-a)} D_\eta(y)dy \tag{4.75}$$

We can see that the integral borders are both shifted by $-a$ in comparison to Equation 4.11 which represents the same tree node without autoregression. For $a = 0$ the equation becomes the same as Equation 4.11 that means integral borders are symmetric around x. For $a = 1$ the borders become $-xp$ and xp and are symmetric around 0. This can be explained as follows: When $a = 1$ the value from the first step in the input sequence is kept without regression and the second η is simply added to this value. Since the first value defines the deadband, the increment in the second step has to be within the deadband borders in order to comply. Therefore the integral borders are symmetric around 0 for $a = 1$.

The equation for deadband violation is:

$$D_{q_2^{11}}(x) = \int_{-\infty}^{\frac{x}{1+p}} D_\eta(x - ay) D_{q_1^1}(y)dy + \int_{\frac{x}{1-p}}^{+\infty} D_\eta(x - ay) D_{q_1^1}(y)dy \tag{4.76}$$

For $a = 0$ this equation becomes the same as Equation 4.14. The integrals cover all values y which do not generate a deadband bound where a given x lies within. Therefore x would certainly violate the bound around those y and become the new reference value. The integrals calculate the probability that this x becomes the new reference value. This is done by integrating over the probability that y occurs in the output distribution q_1^1 multiplied with the probability that an η will occur which leads to the desired x when added to y. This yields the correct output distribution.

Unfortunately, the strong dependency in the input signal for $a > 0$ leads to further complications beginning with the third tree level. Distribution $D_{q_2^{10}}(x)$ shows the probabilities at the deadband sampler's output in the case no violation took place at the second step. But although the output value didn't change since no violation took place, the second value in the input sequence is the one used to generate the third value according to the signal model. The output distribution at this node gives no information about the current state of the signal and can therefore not be used as the reference for nodes q_3^{100} and q_3^{101}. Consequently, only the nodes where a deadband violation took place (the ones with a 1 at the end of the superscript) can be used as reference nodes for further propagation down the tree because only the output distributions at these nodes represent the exact state of the signal at these points.

Accordingly, all node distributions have to be calculated based on the distribution of the last deadband violation on the way back to the root node. It is very easy to see which node has to be used as reference by looking at the binary superscript. So for example for the calculation of the distribution of node q_6^{101001} node q_3^{101} would be the reference node. The tree generation equations have to be extended of course. Equations 4.75 and 4.76 are only applicable if the node to calculate directly follows a deadband violating node.

We first present the equations for nodes q_3^{100} and q_3^{101} where the reference lies 2 steps back before giving the generalized solution.

For q_3^{100} the solution is:

$$D_{q_3^{100}}(x) = D_{q_1^1}(x) \int_{(x-xa)-xp}^{(x-xa)+xp} D_\eta(\eta_1) \int_{(x-(xa+\eta_1)a)-xp}^{(x-(xa+\eta_1)a)+xp} D_\eta(y)dyd\eta_1 \tag{4.77}$$

The outer integral has the same borders as in Equation 4.75 but slightly transformed into a notation, where the single sided deadband size is added to the middle for the upper border and subtracted from the middle for the lower border. This integral covers the inside of the deadband. Because the reference lies two steps back, not only one random increment is added but two. Additionally an autoregression step takes place between the two. The inner integral covers all values of the second increment which lead to deadband compliance for a first increment given in the outer integral. Figure 4.12 shows how the algorithm works. In step i the deadband is defined by the current reference value x. In the next step it is multiplied by the autoregression factor a. Now the first increment η_1 is added. We need to look at all possible η_1 which lead to deadband compliance in the next step. For every one of those possible η_1 we need to make another autoregression step, which means that the current value has to be multiplied by a again. From this value we again need to find

all possible second increments η_2 which again lead to deadband compliance. Those steps have to be repeated for every step the reference signal lies back in the past.

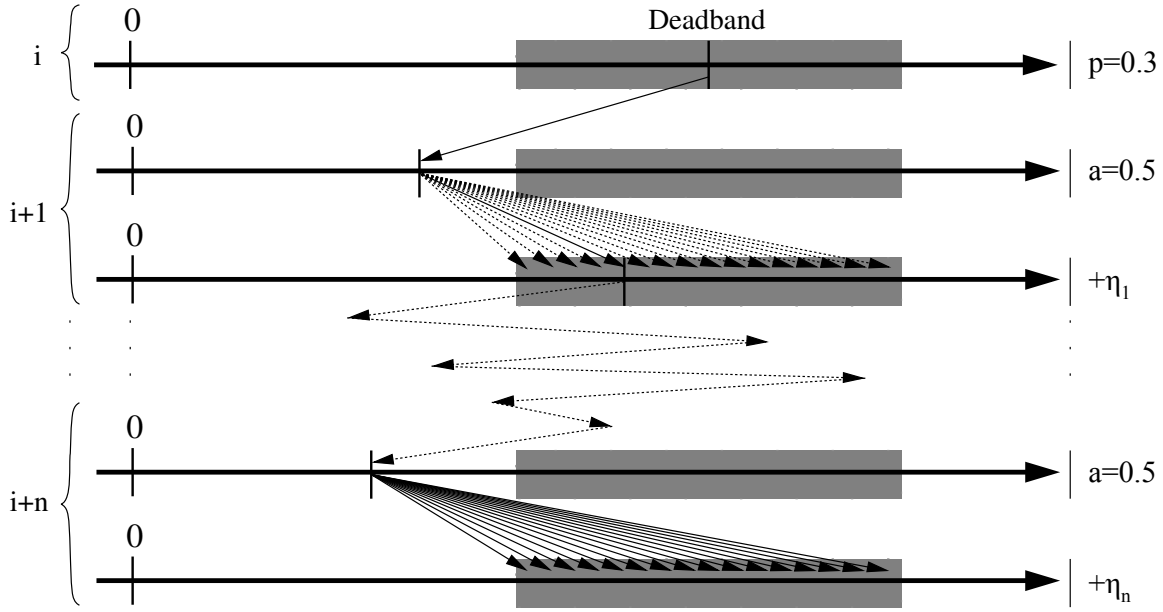

Figure 4.12: Deadband compliance over multiple steps when the reference signal lies back n steps in the case of an autoregressive input signal.

In order to write this is a general form, it is useful to introduce

$$\begin{aligned} c_1 &= xa \\ c_2 &= (c_1 + \eta_1)a \\ c_3 &= (c_2 + \eta_2)a \\ c_{n+1} &= (c_n + \eta_n)a \end{aligned} \tag{4.78}$$

because then we can write

$$D_{q_i^{t10\cdots0}}(x) = D_{q_{i-n}^{t1}}(x) \int_{(x-c_1)-xp}^{(x-c_1)+xp} D_\eta(\eta_1) \int_{(x-c_2)-xp}^{(x-c_2)+xp} D_\eta(\eta_2) \quad \cdots \int_{(x-c_n)-xp}^{(x-c_n)+xp} D_\eta(y) \quad dy d\eta_{n-1} \cdots d\eta_2 d\eta_1 \tag{4.79}$$

For every step the reference signal lies back in the past an additional integral is necessary. Unfortunately the computational complexity for numerically solving this equation increases exponentially with every additional integral.

Deadband violation after 2 steps looks as follows:

$$
\begin{aligned}
D_{q_3^{101}}(x) \;=\; & \int_{-\infty}^{\frac{x}{1+p}} D_{q_1^1}(y) \int_{(y-ya)-yp}^{(y-ya)+yp} D_\eta(\eta_1) D_\eta(y-(ya+\eta_1)a) \quad d\eta_1 dy + \\
& + \int_{\frac{x}{1-p}}^{+\infty} D_{q_1^1}(y) \int_{(y-ya)-yp}^{(y-ya)+yp} D_\eta(\eta_1) D_\eta(y-(ya+\eta_1)a) \quad d\eta_1 dy
\end{aligned} \tag{4.80}
$$

Figure 4.13 helps to explain what happens here. For the calculation of the probability density at value x in the output distribution we have to consider all values y which do not cause a deadband where x lies within. The arrows at the top of the figure symbolize those values y. For each of those y we have to find the probability that it leads to output value x over $n-1$ steps of deadband compliance and a violation in the last step. In order to do that, we calculate the deadband around y because the $n-1$ following steps need to comply to it. Those $n-1$ steps of compliance are treated similarly to Equations 4.77 and 4.79. The final step is different again. With this step n we need to reach the desired output value x and therefore need to calculate the probability that the final η_n is exactly the right number to reach x from the value reached after the $n-1$ complying steps before.

By using adapted Equation 4.78, replacing x with y in c_1, the generalized form of the equation can be written as

$$
\begin{aligned}
& D_{q_i^{t10\cdots1}}(x) = \\
& = \int_{-\infty}^{\frac{x}{1+p}} D_{q_{i-n}^{t1}}(y) \int_{(y-c_1)-yp}^{(y-c_1)+yp} D_\eta(\eta_1) \quad \cdots \int_{(y-c_{n-1})-yp}^{(y-c_{n-1})+yp} D_\eta(\eta_{n-1}) D_\eta(y-c_n) \quad d\eta_{n-1}\cdots d\eta_1 dy + \\
& + \int_{\frac{x}{1+p}}^{+\infty} D_{q_{i-n}^{t1}}(y) \int_{(y-c_1)-yp}^{(y-c_1)+yp} D_\eta(\eta_1) \quad \cdots \int_{(y-c_{n-1})-yp}^{(y-c_{n-1})+yp} D_\eta(\eta_{n-1}) D_\eta(y-c_n) \quad d\eta_{n-1}\cdots d\eta_1 dy
\end{aligned} \tag{4.81}
$$

Also here, the computational complexity for a numerical solution increases exponentially with n.

4.4.3 Validation and Steady State Probability

To validate the correctness of the calculations above the equations were solved numerically for a number of parameter sets of p and a. Due to the high complexity of the problem

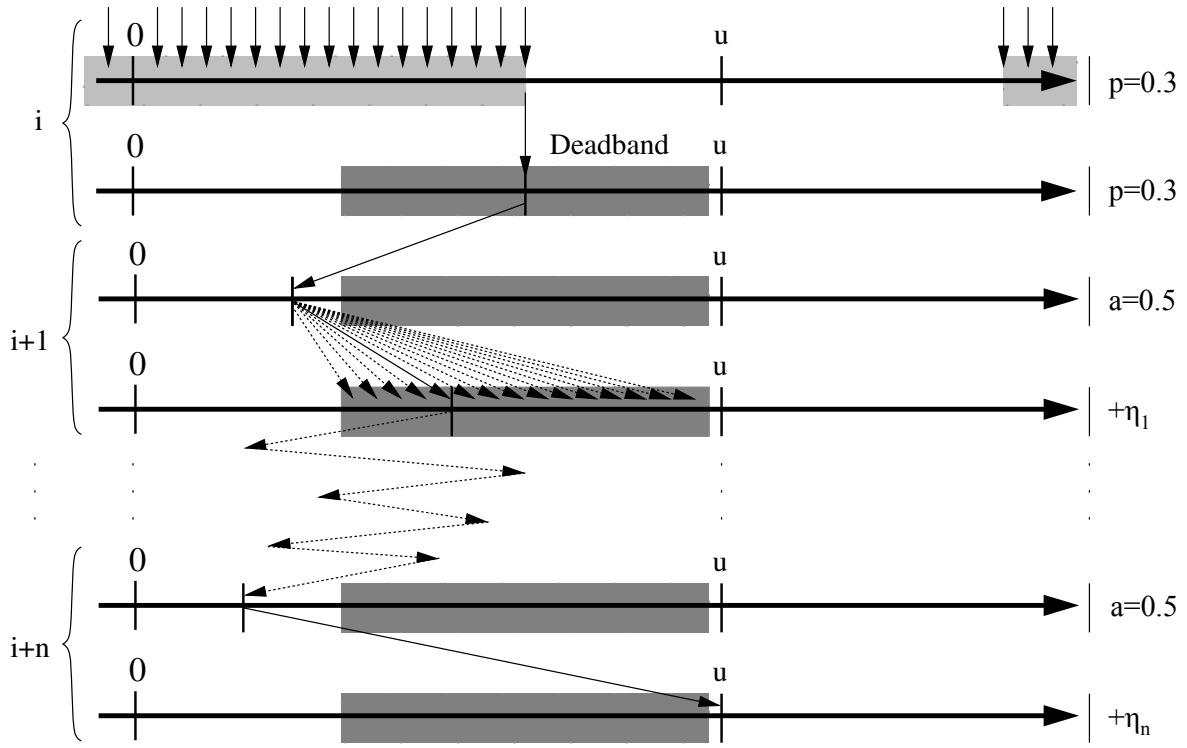

Figure 4.13: Deadband violation after multiple steps of compliance when the reference signal lies back n steps in the case of an autoregressive input signal.

only the first 5 tree levels were calculated in high resolution where the distributions were sampled at intervals of 0.02 from -6 to 6. Three examples for different parameter sets are shown in Figure 4.14.

Of course the estimation of the steady state probability P_{SS} is of high interest to us since it represents the update rate behavior of the deadband sampler for the given signal model. In Figure 4.15 the comparison between the simulation of deadband violation probabilities over 1000000 sample signals is compared to the numerical calculation of those probabilities for four different autoregression parameters ($a = 0.2$, $a = 0.5$, $a = 0.8$, $a = 1.0$) and for deadband parameters from $p = 0.1$ to $p = 0.9$. We can see a very good agreement between simulation and numerical calculation. The small errors for smaller p result from the limited resolution of the numerical calculations which leads to increased errors in the low-p cases. Higher resolutions were unfeasible due to the exponential increase of computational complexity. Even for the presented results four parallel 3GHz-cores were used over a whole week of computation time.

We can also see, that P_{SS} is not always reached after 5 steps. For $a = 0.2$ and $a = 0.5$ the results in level 5 seem to be good estimates for it but for $a = 0.8$ a few more steps would

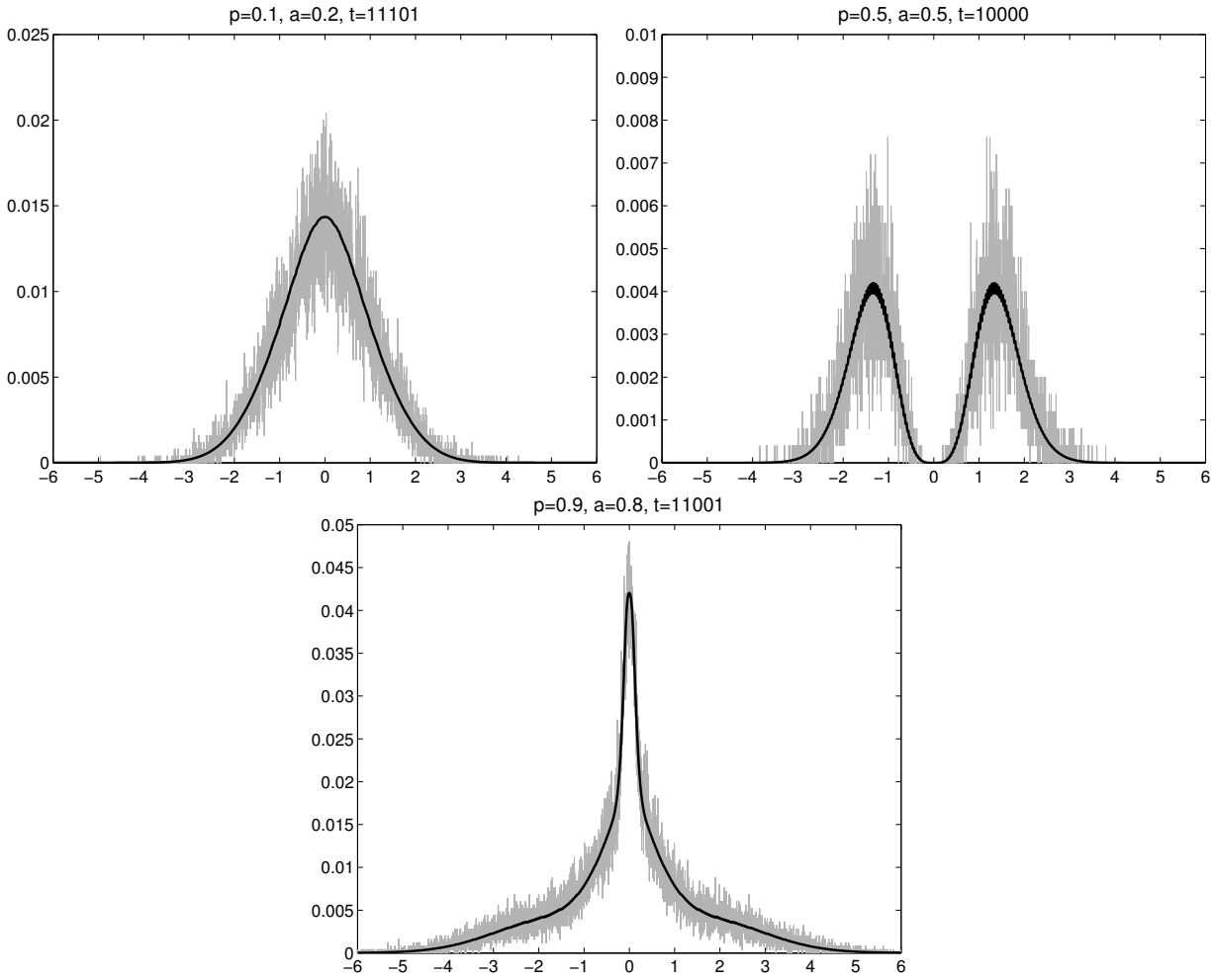

Figure 4.14: Three examples of signal distributions (simulation is grey, numerical calculation black) in tree level 5 for different parameter sets of p and a in the case of an autoregressive input signal.

be necessary for an exact result. For $a = 1.0$ the unbounded nature of the signal in this case does not allow for computation of a P_{SS} but we can say that it finally converges to $P_{SS} = 0$ for $n \to +\infty$.

4.5 Comparison to Real Haptic Signals

Despite the fact that the presented autoregressive signal model is a step into the right direction of more realistic signal models, it is still a long way to go from here. If we take a look at the update rate reductions encountered in the previous chapters we can roughly

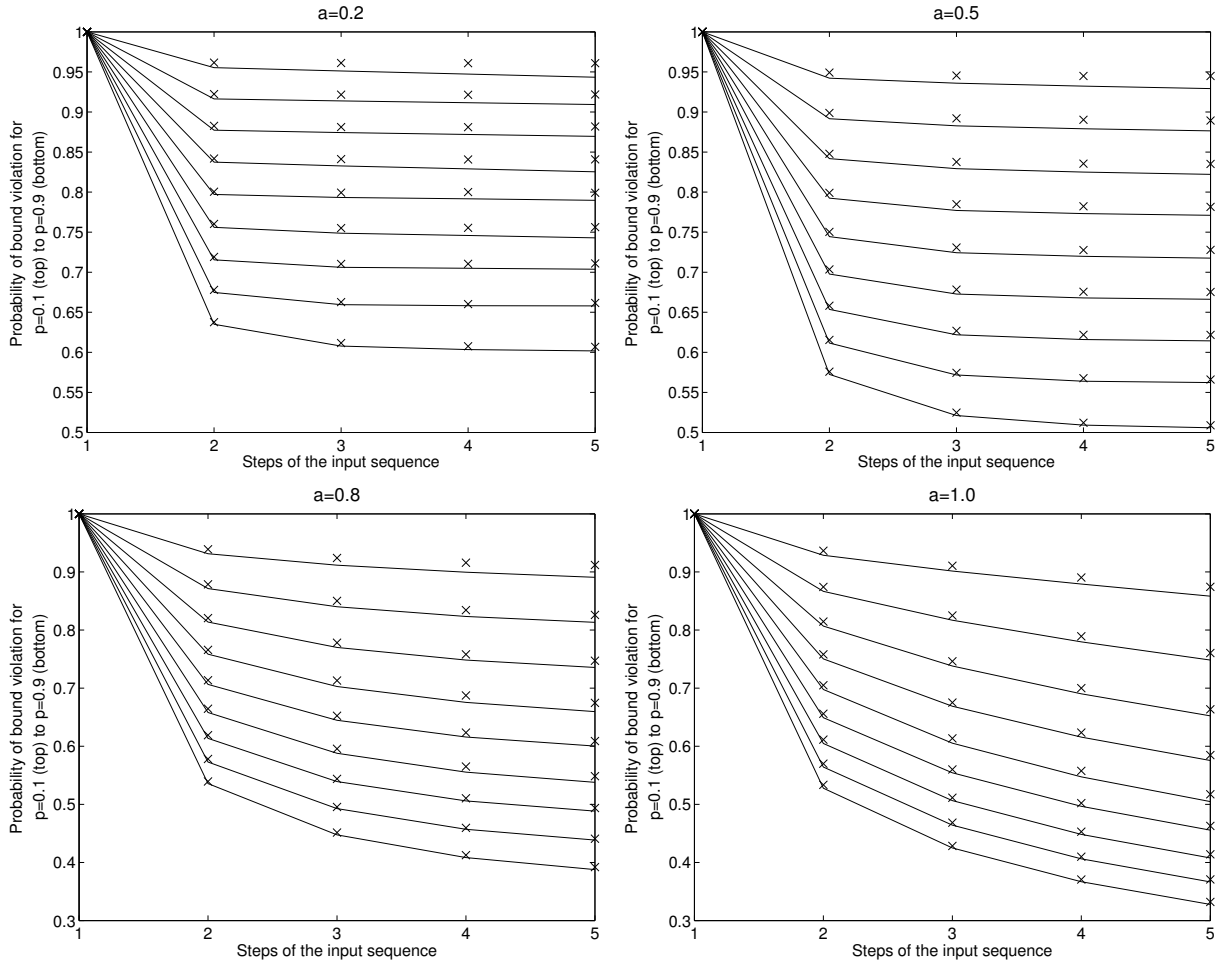

Figure 4.15: Simulation (lines) and numerical calculation (crosses) of deadband bound violation probabilities for different deadband parameters p and different autoregression parameters a for an autoregressive input sequence.

say that with a deadband parameter of $p = 20\%$ we encounter update rate reductions in the region of 90%. By looking at Figure 4.15 we can see that even for an AR-coefficient of $a = 0.8$ the update rate converges somewhere around 0.8 leaving us with an update rate reduction of 20%.

The main reason for this discrepancy lies in the fact that due to the bounded nature and the signal statistics of the AR-process for $0 < a < 1$ the signal hardly reaches regions in which the variance of the random increment η is small enough in comparison to the absolute amplitude of the signal so that it barely triggers the deadband as it is the case in real haptic signals where the signal change is mostly small compared to the absolute amplitude.

A first possibility to further enhance the signal model would be to introduce a constant offset c into the signal model. The equations would then look like

$$\begin{aligned} x_{i+1} &= ax_i + \eta \\ y_i &= c + x_i \qquad \text{for} \quad i = 1 \dots N \end{aligned} \tag{4.82}$$

to generate a new signal model y_i. c should then be adequately chosen according to the behavior of real haptic signals. One way to parameterize would be to measure the ratio of the variance of the change per step and the average amplitude of the real signal and adapt c accordingly, so that the ratio of c and the variance of η matches this measurement. A quite realistic signal model for the velocity during free space motion or the force signal during continuous contact could be the result.

Another extension of the signal model would be necessary in order to correctly represent the transient behavior of haptic signals when encountering contact. The velocity suddenly becomes very low and on the other hand force values increase instantly. For the modeling of this behavior a two-state Markov model, one state representing free space motion and the other representing contact, would be suitable. The transition probabilities could be parameterized according to measurements of average contact duration and contact probability in real haptic signals. Depending on the state of the Markov model two appropriately dimensioned signal models could be used to generate the next output samples.

4.6 Chapter Summary

This chapter presents theoretical work on the update rate of the deadband sampler for different signal models. Two general types of signal models are examined: iid signals and autoregressive signals with iid increment. For all presented signal models theoretical results for the calculation of output distributions and probabilities are given which are validated by numerical and simulative results.

The results of this chapter show the surprising complexity of the theoretical deduction of the signal behavior of an, at first glance, very simple algorithm. It becomes clear that the nonlinear nature of the deadband algorithm leads to challenges in the theoretical analysis. Nevertheless this chapter presents solutions for some interesting signal models and successfully verifies them.

Although even the most realistic signal model presented here leads to a significant difference in terms of the update rate when compared to experiments with real haptic signals, its

theoretical analysis forms a solid base for more realistic signal models. Discovering and understanding how the theoretical analysis for the deadband sampler can be done for different signal models greatly helps with its practical applications.

The next steps should be to identify how the presented signal models can be changed or extended in order to better reflect haptic signals encountered in TPTA systems and so further extend and simplify the theoretical analysis for these models. One possible extension would be the inclusion of multiple degrees of freedom since the presented approaches only cover 1-DoF signals. The final goal would be the exact prediction of the update rate behavior of a TPTA system which uses the deadband approach for data reduction.

Chapter 5

Conclusion and Outlook

Telepresence and teleaction systems will greatly gain in public, scientific, and economical interest in the near future. The rapid advances in technology of sensor and actuator hardware, control systems, network capacities, and data processing and coding will allow for efficient, reliable and cost-effective usage of such systems in many aspects of human society.

Hardware
Despite ongoing research over many years, current hardware for TPTA systems is still not as functional as desired. Sensors for force and torque already reach high precision and reliability levels and reach desirable sizes for their integration in lightweight high-fidelity systems. In contrast, actuators for tactile modalities like surface structure, vibrations and temperature are still in early development stages. Robotic systems that are able to cover the whole workspace of a human being and which are able to produce forces and torques in a range of human interaction forces/torques are still very large, heavy, and extremely expensive. Humanoid robots certainly lead in the right direction of how teleoperator systems should look like, but their dexterity and versatility is also still very limited. Still, recent advances in these areas point out that there will be decent hardware systems for the use in TPTA in the near future.

Control Systems
It was already pointed out throughout this dissertation that the underlying control structure of the TPTA system plays a fundamental role for its safe and reliable operation. Both the local control loops at the HSI and the TOP and the global control loop over the communication systems need close consideration concerning their control structure and control

algorithms. The desirable behavior of the HSI and the TOP for example would be very low mechanical resistance during free-space motions and very rigid and stiff behavior during hard contacts. Since it is difficult to find a single control approach which equally satisfies both of these desired properties, hybrid approaches become more and more popular. For the control of the global control loop across the communication system it is very important to be able to compensate for the imperfections of the communications link. One main problem is the inherent unavoidable delay which long distance communication introduces. This delay, which would lead to system instabilities if not properly dealt with, is adressed by the use of wave variables (Scattering Transformation) on the transmitted data. Still higher delays lead to worse transparency of the system. Therefore keeping delays to the absolute minimum is and will be the primary design objective of TPTA communication systems.

Network Systems

The underlying transport network also plays an important role for the operation of TPTA systems. For crucial tasks a dedicated line is still the favorable solution today, mainly due to the missing quality of service support of the Internet in its current state. Recent advances in networking technology and especially network resilience will help to make the Internet much more reliable and also faster and more versatile. Therefore its usage as the primary infrastructure for crucial and time-critical data will certainly increase in the near future.

Coding of Multimedia Data

The efficient coding of audio and video data is subject of intensive ongoing research. Mainly the great demand of powerful compression algorithms in the home entertainment industry supports those efforts and leads to quick progress in compression ratio and quality. TPTA systems only partly profit from this intensive research, since the requirements on the coding schemes in such a system are different from those in the entertainment sector. In TPTA systems, very low delay solutions are desirable for both video and audio whereas quality and high compression gain are the prime objectives in the entertainment sector. And of course since the demand for the compression of haptic data has not even emerged in this sector very little research has been done so far.

Outlook

As the usage of TPTA systems will always coincide with the need of efficient data transmission using specialized approaches for the different modalities needed for good immersion of the human operator in the remote environment, the work presented in this dissertation will

certainly be continued and extended in the upcoming years. Some possibilities to further enhance the described approaches as well as new research directions originating from this work can be the following.

The exact form of the deadzone in the multi-DoF deadband approaches would be one interesting point for further research. The assumption of an isotropic deadzone seems straight forward and could certainly be adapted more precisely to human haptic perception by respective experiments. First steps were already made but both methodology and experimentation need more refinement to reach conclusive results.

In its current state, only three degrees of freedom of haptic data can be combined in a reasonable way to form a conclusive vector representation on which the DBDR approach can reliably work on. In systems with many more DoFs the same problem as in the transition from 1-DoF to 3-DoF would arise. It would be very interesting to find reasonable representations, possibly in higher dimensions, to cope with this problem. A first idea could be the combination of three translational and three rotational DoFs in a combined pose representation.

In the various experiments, especially with real TPTA systems, an optimized solution for the functional transmission of position data would be desirable. The DBDR approach only works on velocity and does not make sense on pure position data. Therefore the omission of sample values while using the DBDR approach leads to position drifts at the TOP which tend to increase over time. The use of position control at the TOP side with predicted positions while no new samples are transmitted works well for virtual TOPs but leads to sudden hardware movements in real TPTA systems when the position prediction error becomes too large. An approach to deal with this complication would be very desirable and should be investigated thoroughly because of its high practical relevance. Possible solutions like regular updates at lower frequencies to keep the prediction error small, additional position error tracking at the OP side, or position error concealment at the TOP side would be only a few possible approaches.

With respect to the influence of compression schemes on task performance, numerous interesting experiments could be conducted. Primarily, a more sophisticated TPTA system with higher transparency and optimized control schemes would be a very interesting platform for experimental testing. One could hope for more detailed results on parameter ranges to use and on the severity of certain distortions to task performance.

From a theoretical point of view it would be of considerable interest to find a suitable signal model to describe the behavior of haptic data in detail with a number of parameters which can be related to the task or the environment at hand. Possible parameters could be the ratio of free-space motion and contact, average force exertion, stiffness of the environment

etc. After the derivations done in this dissertation it would be very hard to find a theoretical solution for such a signal model, but at least simulations would be much more accurate.

List of Figures

List of Tables

References

Publications of the author:

[1] 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*, 56(2):588–597, February 2008.

[2] P. Hinterseer, J. Kammerl, E. Steinbach, and S. Chaudhuri. The weber quantizer: Perceptual coding for networked telepresence and teleaction. In *Proc. of the First International Conference on Robot Communication and Coordination (Robocomm)*, page 29, Athens, Greece, October 2007.

[3] P. Hinterseer and E. Steinbach. A psychophysically motivated compression approach for 3d haptic data. In *Proc. of the IEEE Haptics Symposium*, pages 35–41, Alexandria, VA, USA, March 2006.

[4] P. Hinterseer, E. Steinbach, and S. Chaudhuri. Model based data compression for 3d virtual haptic teleinteraction. In *Proc. of the IEEE Int. Conf. on Consumer Electronics*, pages 23–24, Las Vegas, NV, USA, January 2006.

[5] P. Hinterseer, E. Steinbach, and S. Chaudhuri. Perception-based compression of haptic data streams using kalman filters. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages V 473–476, Toulouse, France, May 2006.

[6] P. Hinterseer, E. Steinbach, S. Hirche, and M. Buss. A novel, psychophysically motivated transmission approach for haptic data streams in telepresence and teleaction systems. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 1097–1100, Philadelphia, PA, USA, March 2005.

[7] P. Hinterseer, E. Steinbach, S. Hirche, and M. Buss. Parsimoneous data transmission in haptic telepresence systems. In *Proc. of the Workshop on Human Centered Robotic Systems (HCRS)*, pages 19–24, Munich, Germany, October 2006.

[8] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss. Network traffic reduction in haptic

telepresence systems by deadband control. In *Proc. of the IFAC World Congress, International Federation of Automatic Control*, Prague, Czech Republic, 2005.

[9] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss. Towards deadband control in networked teleoperation systems. In *Proc. of the IFAC World Congress, International Federation of Automatic Control*, Prague, Czech Republic, 2005.

[10] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss. Transparent data reduction in networked telepresence and teleaction systems. part i: Communication without time delay. *Presence, Teleoperators and Virtual Environments*, 16(5):523–531, October 2007.

[11] J. Kammerl, P. Hinterseer, S. Chaudhuri, and E. Steinbach. A theoretical analysis of data reduction using the weber quantizer. In *Proc. of the Data Compression Conference*, pages 524–524, Snowbird, Utah, USA, March 2008.

[12] J. Kammerl, P. Hinterseer, and E. Steinbach. A novel signal reconstruction algorithm for perception based data reduction in haptic signal communication. In *Proc. of the Int. Conference on Computer Communications and Networks (ICCCN)*, pages 1309–1314, Honolulu, Hawaii, USA, August 2007.

[13] A. Kron, G. Schmidt, B. Petzold, M. F. Zäh, P. Hinterseer, and E. Steinbach. Disposal of explosive ordnances by use of a bimanual haptic telepresence system. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1968–1973, New Orleans, USA, April 2004.

[14] V. Nitsch, P. Hinterseer, E. Steinbach, B. Färber, and L. Geiger. An experimental study of lossy compression in a real telepresence and teleaction system. In *Proc. of the Workshop of Haptic Audio Visual Environments and their Applications (HAVE)*, pages 75–80, Ottawa, Ontario, Canada, October 2008.

[15] H. Pongrac, B. Färber, P. Hinterseer, J. Kammerl, and E. Steinbach. Limitations of human 3d force discrimination. In *Proc. of the Workshop on Human Centered Robotic Systems (HCRS)*, pages 109–114, Munich, Germany, October 2006.

[16] M.F. Zaeh, S. Clarke, P. Hinterseer, and E. Steinbach. Telepresence across networks: A combined deadband and prediction approach. In *Proc. of the Int. Conference on Information Visualisation (IV)*, pages 597–604, London, UK, July 2006.

Other Publications and Sources:

[17] R. Anderson and M. Spong. Bilateral control of teleoperators with time delay. In *IEEE Transactions on Automatic Control*, volume 34, pages 494–501, 1989.

[18] C. Basdogan, S. De, J. Kim, M. Muniyandi, H. Kim, and M.A. Srinivasan. Haptics in minimally invasive surgical simulation and training. *IEEE Computer Graphics and Applications*, 24(2):56–64, March-April 2004.

[19] S.J. Biggs and M. A. Srinivasan. *Handbook of Virtual Environments*, chapter 5: Haptic Interfaces, pages 93–116. Lawrence Earlbaum, Inc., London, 2002.

[20] E. Bodden, M. Clasen, and J. Kneis. Arithmetic coding revealed. *Seminar Data Compression*, May 2004.

[21] C. W. Borst. Predictive coding for efficient host-device communication in a pneumatic force-feedback display. In *Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 596–599, Pisa, Italy, March 2005.

[22] Grigore C. Burdea. *Force and Touch Feedback for Virtual Reality*. Wiley, 1996.

[23] H. R. Choi, B. H. Choi, and S. M. Ryew. Haptic display in the virtual collaborative workspace shared by multiple users. In *Proc. of the IEEE Int. Workshop on Robot and Human Communication*, pages 478–483, Sendai, Japan, September 1977.

[24] L.D. Davisson. The theoretical analysis of data compression systems. *Proceedings of the IEEE*, 56(2):176–186, February 1968.

[25] J. Drösler. An n-dimensional weber law and the corresponding fechner law. *Journal of Mathematical Psychology*, 44:330–335, 2000.

[26] I. Elhajj, N. Xi, W. K. Fung, Y. H. Liu, W. J. Li, T. Kaga, and T. Fukuda. Haptic information in internet-based teleoperation. *IEEE/ASME Transactions on Mechatronics*, 6(3):295–304, September 2001.

[27] A. Field. *Discovering Statistics using SPSS*. Second Edition. Sage Publications, London, GB, 2005.

[28] A. P. Field. A bluffer's guide to sphericity. *Newsletter of the Mathematical, Statistical and Computing section of the British Psychological Society*, 6(1):13–22, 1998.

[29] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, December 1937.

[30] M. Friedman. A correction: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 34(205):109, March 1939.

[31] E.B. Goldstein. *Sensation and Perception.* Sixth Edition. WADSWORTH, Pacific Grove, CA, USA, 2002.

[32] J. A. Greefkes and K. Riemens. Code modulation with digitally controlled companding for speech transmission. *Philips Tech. Rev.*, 31:335–353, 1970.

[33] S. W. Greenhouse and S. Geisser. On methods in the analysis of profile data. *Psychometrika*, 24(2):95–112, June 1959.

[34] M. Guazzo. A general minimum-redundancy source-coding algorithm. *IEEE Transactions on Information Theory*, IT-26(1):15–25, January 1980.

[35] C. Gunn, M. Hutchins, D. Stevenson, M. Adcock, and P. Youngblood. Using collaborative haptics in remote surgical training. In *Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC)*, pages 481–482, Pisa, Italy, March 2005.

[36] J. P. Hespanha, M. McLaughlin, and G. S. Sukhatme. Haptic collaboration over the internet. In *Touch in Virtual Environments: Haptics and the Design of Interactive Systems*, chapter 8, pages 158–168. Prentice Hall, 2002.

[37] K. Hikichi, H. Morino, I. Fukuda, S. Matsumoto, Y. Yasuda, I. Arimoto, M. Iijima, and K. Sezaki. Architecture of haptics communication system for adaptation to network environments. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo*, pages 744–747, Tokyo, Japan, August 2001.

[38] S. Hirche. *Haptic Telepresence in Packet Switched Communication Networks.* Nr.1082 in Fortschrittsberichte VDI, series 8: Mess-, Steuerungs- und Regelungstechnik. VDI-Verlag, PhD thesis, Düsseldorf, 2005.

[39] S. Hirche, A. Bauer, and M. Buss. Transparency of haptic telepresence systems with constant time delay. In *Proc. of the IEEE Int. Conf. on Control Applications*, pages 328–333, Toronto, Canada, 2005.

[40] D. C. Howell. *Statistical methods for psychology.* Fifth Edition. Duxbury, 2002.

[41] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[42] G. Hwang and H. Hashimoto. Developing dexterous bilateral nanomanipula-

tion system using haptic interface. In *Proc. of the Int. Symposium on Micro-NanoMechatronics and Human Science*, pages 1–6, Nagoya, Japan, November 2006.

[43] Y. Ishibashi, T. Hasegawa, and S. Tasaka. Group synchronization control for haptic media in networked virtual environments. In *Proc. of the 12th Int. Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 106–113, Chicago, Illinois, USA, March 2004.

[44] L. A. Jones and I. W. Hunter. Human operator perception of mechanical variables and their effects on tracking performance. *Advances in Robotics*, 42:49–53, 1992.

[45] L.A. Jones and M. Berris. The psychophysics of temperature perception and thermal-interface design. In *Proc. of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 137–142, Orlando, FL, USA, March 2002.

[46] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[47] A. Kron and G. Schmidt. A bimanual haptic telepresence system - design issues and experimental results. In *In Proc. of the Int. Workshop on High-Fidelity Telepresence and Teleaction jointly with IEEE Humanoids 2003*, Munich, Germany, 2003.

[48] D. Lee, O. Martinez-Palafox, and M. Spong. Bilateral teleoperation of a wheeled mobile robot over delayed communication network. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3298–3303, Orlando, Florida, USA, May 2006.

[49] H. R. Lindman. *Analysis of variance in complex experimental designs*. W. H. Freeman & Co, San Francisco, 1974.

[50] C. Mahlo, C. Hoene, A. Rosami, and A. Wolisz. Adaptive coding and packet rates for tcp-friendly voip flows. In *Proc. of the Int. Symposium on Telecommunications*, Shiraz, Iran, September 2005.

[51] J. W. Mauchly. Significance test for sphericity of a normal n-variate distribution. *The Annals of Mathematical Statistics*, 11(2):204–209, June 1940.

[52] A. Ortega and Y. Liu. Lossy compression of haptic data. In *Touch in Virtual Environments: Haptics and the Design of Interactive Systems*, chapter 6, pages 119–136. Prentice Hall, 2002.

[53] P. G. Otanez, J. R. Moyne, and D. M. Tilbury. Using deadbands to reduce communication in networked control systems. In *Proc. of the American Control Conference*, Anchorage, Alaska, May 2002.

[54] E. Ou and C. Basdogan. Network considerations for a dynamic shared haptic environ-

ment. In *Proc. of The Nat. Conf. on Undergraduate Research*, Whitewater, Wisconsin, USA, April 2002.

[55] K. S. Park and V. Kenyon. Effects of network characteristics on human performance in a collaborative virtual environment. In *Proc. of the IEEE Virtual Reality Conference*, pages 104–111, Houston, TX, USA, August 1999.

[56] A. Peer, Y. Komoguchi, and M. Buss. Towards a mobile haptic interface for bimanual manipulations. In *Proc. of the Int. Conf. on Intelligent Robots and Systems*, pages 384–391, San Diego, USA, October/November 2007.

[57] A. Peer, U. Unterhinninghofen, and M. Buss. Tele-assembly in wide remote environments. In *Proc. of the 2nd Int. Workshop on Human-Centered Robotic Systems*, pages 75–80, Munich, Germany, October 2006.

[58] B. Petzold, M.F. Zaeh, B. Faerber, B. Deml, H. Egermeier, J. Schilp, and S. Clarke. A study on visual, auditory and haptic feedback for assembly tasks. *PRESENCE: Teleoperators and Virtual Environments, MIT Press*, 13:16–21, 2004.

[59] G. Rosati, A. Rossi, G. Boschetti, and A. Trevisani. First experimental results of an integrated robotic system for haptic teleoperation. In *Proc. of the Symposium on Industrial Electronics*, pages 3138–3143, Vigo, Spain, June 2007.

[60] D. Roy. Design and implementation of remote robotic control system for nuclear power plant application achieved through kinesthetic force feedback model. In *Proc. of the Int. Conf. on Power Electronics, Drives and Energy Systems for Industrial Growth*, pages 1018–1023, New Delhi, India, January 1996.

[61] F. Rubin. Arithmetic stream coding using fixed precision registers. *IEEE Transactions on Information Theory*, IT-25(6):672–675, November 1979.

[62] N. Sakr, N.D. Georganas, J. Zhao, and X. Shen. Towards an architecture for the compression of haptic media. In *IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VECIMS)*, pages 13–18, Ostuni, Italy, June 2007.

[63] R. Scheuchenpflug. Measuring presence in virtual environments. In *Proc. of the Int. Conference on Human-Computer Interaction*, pages 56–58, New Orleans, LA, USA, August 2001.

[64] K.J. Schilling and H. Roth. Control interfaces for teleoperated mobile robots. In *Proc. of the IEEE Int. Conf. on Emerging Technologies and Factory Automation*, pages 1399–1403, Barcelona, Spain, October 1999.

[65] H. R. Schindler. Delta modulation. *IEEE Spectrum*, 7:69–78, 1970.

[66] C. Shahabi, M.R. Kolahdouzan, G. Barish, R. Zimmermann, D. Yao, K. Fu, and L. Zhang. Alternative techniques for the efficient acquisition of haptic data. *ACM SIGMETRICS Performance Evaluation Review*, 29(1):334–335, 2001.

[67] C. Shahabi, A. Ortega, and M. R. Kolahdouzan. A comparison of different haptic compression techniques. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo*, pages 657–660, Lausanne, Switzerland, August 2002.

[68] K. Shimoga. Finger force and touch feedback issues in dexterous telemanipulation. In *Proc. of the NASA-CIRSSE Int. Conf. on Intelligent Robotic Systems for Space Exploration*, pages 159–178, Greenbelt, MD, USA, Sptember 1992.

[69] K. Shimoga. Finger force and touch feedback issues in dexterous telemanipulation: A survey. In *Proc. of the NASA-CIRSSE Int. Conf. on Intelligent Robotic Systems for Space Exploration*, pages 159–178, Greenbelt, MD, USA, September 1992.

[70] R. T. Souayed, D. Gaiti, G. Pujolle, W. Yu, Q. Gu, and A. Marshall. Haptic virtual environment performance over ip networks: A case study. In *IEEE Symposium on Distributed Simulation and Real-Time Applications*, pages 181–189, Delft, Netherlands, October 2003.

[71] S. Tomoyuki, K. Seiichiro, K. Ryogo, and O. Kouhei. Multilateral control for skill education based on haptic data storage. In *Proc. of the IEEE Int. Conf. on Industrial Technology (ICIT)*, pages 334–339, Mumbai, India, December 2006.

[72] C.J. van Strijp, H.H. Langen, and M. Onosato. The application of a haptic interface on microassembly. In *Proc. of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 289–293, Arlington, Virginia, USA, March 2006.

[73] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

[74] J. Zhou, X. Shen, and N.D. Georganas. Haptic tele-surgery simulation. In *Proc. of the IEEE Int. Workshop on Haptic, Audio and Visual Environments and Their Applications (HAVE)*, pages 99–104, Ottawa, Ontario, Canada, October 2004.