COMMUNICATION AND TIME DISTORTION

BY

THOMAS RIEDL

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Professor Andrew Singer, Chair
Professor Bruce Hajek
Adjunct Professer Sean Meyn
Professor Pierre Moulin
Professor Rüdiger Urbanke, École Polytechnique Fédérale de Lausanne

ABSTRACT

Communication systems always suffer time distortion. At the physical layer asynchrony between clocks and motion-induced Doppler effects warp the time scale, while at higher layers there are packet delays.

Current wireless underwater modems suffer a significant performance degradation when communication platforms are mobile and Doppler effects corrupt the transmitted signals. They are advertised with data rates of a few kbps, but the oil and gas industry has found them useful only to around 100 bps. In our work, time-varying Doppler is explicitly modeled, tracked and compensated. Integrated into an iterative turbo equalization based receiver, this novel Doppler compensation technique has demonstrated unprecedented communication performance in US Navy sponsored field tests and simulations. We achieved a data rate of $39kbps$ at a distance of $2.7km$ and a data rate of $1.2Mbps$ at a distance of $12m$. The latter link is capable of streaming video in real-time, a first in wireless underwater communication.

Time distortion can also be intentional and be used for communication. We explore how much information can be conveyed by controlling the timing of packets when sent from their source towards their destination in a packet-switched network. By using Markov chain analysis, we prove a lower bound on the maximal channel coding rate achievable at a given blocklength and error probability.

Finally, we propose an easy-to-deploy censorship-resistant infrastructure, called FreeWave. FreeWave modulates a client's Internet traffic into acoustic signals that are carried over VoIP connections. The use of actual VoIP connections allows FreeWave to relay its VoIP connections through oblivious VoIP nodes, hence keeping the FreeWave server(s) unobservable and unblockable. When the VoIP channel suffers packet transfer delays, the transmitted acoustic signals are time distorted. We address this challenge and prototype FreeWave over Skype, the most popular VoIP system.

To my parents, for their love and support.

TABLE OF CONTENTS

CHAPTER 1

OVERVIEW

Communication systems always suffer time distortion. At the physical layer asynchrony between clocks and motion-induced Doppler effects warp the time scale, while at higher layers there are packet delays. Particularly in acoustic communication channels, Doppler can be catastrophic if not compensated dynamically. This is mainly due to the higher Mach numbers experienced in these channels. Our research in acoustic communications tries to understand the fundamental causes of this effect and uses the gained insight towards the implementation of more robust and faster acoustic communication systems. Acoustic communication is still in its infancy and the research community has yet to agree on a standard channel model. Chapter 2 gives a comprehensive overview of current acoustic research and methods and derives a new channel model from first principles. Perhaps a fatal flaw of previous works is that they borrow the channel model from the radio communication community. Our model builds upon the established physical principles of acoustic wave propagation. We unveil the close relationship between acoustic positioning and communications. This opens the door for inertial sensors to enhance signal detection. We derive an efficient receiver algorithm based upon this new channel model and show its superior performance in simulations, laboratory experiments and at-sea field-tests.

In Chapter 3 we explore how much information can be conveyed by controlling the timing of packets when sent from their source towards their destination in a packet-switched network. The aggregate effect of the involved forwarding nodes can be modeled as a queuing timing channel. The exponential server timing channel is known to be the simplest, and in some sense canonical, queuing timing channel. The capacity, C, of this infinite-memory channel is known. We discuss practical finite-length restrictions on the codewords and attempt to understand the maximal rate that can be achieved for a target error probability. By using Markov chain analysis, we prove a lower bound on the maximal channel coding rate achievable at blocklength n and error probability ϵ. The bound is approximated by $C - n^{-1/2}\sigma Q^{-1}(\epsilon)$, where Q denotes the Q-function and σ^2 is the asymptotic variance of the underlying Markov chain. A closed form expression for σ^2 is given.

Open communication over the Internet poses a serious threat to countries with repressive regimes, leading them to develop and deploy censorship mechanisms within their networks. Unfortunately, existing censorship circumvention systems face difficulties in providing *unobservable* communication with their clients; this highly limits their *availability* as censors can easily block access to circumvention systems that make observable communication patterns. Moreover, the lack of unobservability may pose serious threats to their users. Recent research takes various approaches to tackle this problem; however, they introduce new challenges, and the provided unobservability is breakable. In Chapter 4 we propose an easy-to-deploy and unobservable censorship-resistant infrastructure, called FreeWave. FreeWave works by modulating a client's Internet traffic into acoustic signals that are carried over VoIP connections. Such VoIP connections are targeted to a server, the FreeWave server, that extracts the tunneled traffic and proxies them to the uncensored Internet. The use of actual VoIP connections, as opposed to traffic morphing, allows FreeWave to relay its VoIP connections through oblivious VoIP nodes (e.g., Skype supernodes), hence keeping the FreeWave server(s) unobservable and unblockable. In addition, the use of end-to-end encryption, which is supported/mandated by most VoIP providers like Skype, prevents censors from distinguishing FreeWave's VoIP connections from regular VoIP connections. To utilize a VoIP connection's throughput efficiently we design communications encoders tailored specifically for VoIP's lossy channel. We prototype FreeWave over Skype, the most popular VoIP system. A major challenge is the time distortion that the acoustic signal experiences when sent over the Skype channel. This distortion is caused by packet transfer delays and its intensity depends on the level of network congestion and the number of routers along the way of transmission. We show that FreeWave is able to reliably achieve communication bandwidths that are sufficient for web browsing, even when clients and the FreeWave server are thousands of miles apart. We also validate FreeWave's communication unobservability against traffic analysis and standard censorship techniques.

CHAPTER 2

ACOUSTIC POSITIONING AND COMMUNICATION

2.1 Introduction

> Imagine that you are in the midst of events that are about to trigger the largest accidental marine oil spill in history, and you do not even know it is happening. What is worse, even if you know what is happening, miles beneath the ocean surface, you have no way to stop it.
>
> An explosion at the surface causes massive structural failure. Communication wires are cut. Control of the subsea infrastructure is lost.
>
> The ensuing collapse of the column destroys the blowout preventer, eliminating the last remaining safety mechanism that could have prevented an uncontrolled oil flow from the Deepwater Horizon site into the ocean in April 2010.
>
> During this process, according to the US Government Macondo Expert Report [1], a reliable underwater wireless communication backup link was unavailable, but could have prevented the resulting unimaginable environmental disaster.
>
> Days passed before the fire on the surface was sufficiently under control for a ship to be brought to the location safely. This enabled a remotely operated vehicle, tethered by cabled communications, to begin subsea repair. Immediate remote vehicle operation via wireless control, enabling operation at a safe distance, was (and still is) unavailable.

This is but one scenario illustrating the need for a dramatic improvement in the wireless communications and control capabilities within our world's oceans. Imagine life today without GPS, WIFI, or mobile phones. The transition from wired to wireless communication over the past 20 years has fundamentally changed how people interact and how industries operate. Unfortunately, this technological revolution has had little impact on communication undersea. Radio waves - used to carry information wirelessly above land - propagate poorly in seawater. As a result, revolutionizing wireless communication technologies such as GPS, WIFI or cellular communication do not work below the ocean surface: Industries and organizations that operate underwater are still in the digital dark ages. Communication underwater is still almost entirely done through wired links; literally a wire or a cable connects the sender to the receiver.

Underwater operations that rely on divers are expensive, restricted to shallow waters, and put a human life at risk. The subsea industry relies on remotely operated vehicles (ROVs) for virtually all work performed in the deep ocean. An operator on the surface communicates with the machine through a bulky cable that usually is about $3.5km$ long [2]. A massive surface ship is required to safely deploy such a vehicle and handle its heavy cable to the sea floor. Even when winds are strong and waves are high, the surface ship needs to be capable to hold its position right above the vehicle. Mooring or anchoring is not practical in deep water or above dense infrastructure at the sea bottom. So instead these ROV support ships are outfitted with expensive dynamic positioning systems that use GPS, inertial sensor and gyro compass readings to automatically control position and heading exclusively by means of active thrust. Such ships cost about $\$120k$ per day [3, 4]. If, instead of a cable, a wireless carrier is used to communicate with the ROV, the heavy cables can be cut and these expensive surface ships are no longer needed. Subsea missions could be accomplished quicker, cheaper and with fewer personnel. The surface vessel is the main cost driver in underwater vehicle operations. In 2013, the subsea industry demanded more than $123k$ ROV days [5, 6] and these are expected to increase to at least $140k$ days in 2017. Since each ROV support ship only carries $1-2$ ROVs, the total expenditure on these ships is over $\$7B$. Wireless links could eliminate the surface vessel and associated cost.

There is a clear need for reliable, high-speed wireless underwater communications for remote-control of subsea machinery. A data rate of $1Mbps$ and a range of $100m$ are the minimum communication requirements for this application [3]. Existing wireless solutions are far from satisfying these requirements. They are based on acoustic modem technology developed in the late 1980s. Their vendors advertise them with data rates of a few kbps, but the oil and gas industry has found them useful only to around $100bps$ [3], relegating this technology to only the most rudimentary of low data rate applications if not completely unusable. Uploading a simple 100 kilobyte image takes hours. Video and real-time control is impossible.

The ocean covers 71% of the Earth's surface. It holds the vast majority of its mineral and fossil resources, it is home to over 95% of the world's living biomass and it carries 90% of international trade. The exploration, utilization, and protection of this space is of utmost importance to society, but they require deployment of subsea machinery and an effective way of communicating with it. Leading energy firm Douglas-Westwood [7] and the US Navy underline the importance of underwater communications. The Navy Unmanned Undersea Vehicle (UUV) Master Plan [8], for example, repeatedly highlights communications as a severe limitation in today's undersea missions, and "particularly in the area of acoustic communications, advancements are desirable in bandwidth, data rates, range, security, and reliability." The acoustic communication technology described in this thesis is capable of meeting these needs. It has the potential to completely revolutionize underwater environmental monitoring, scientific exploration, resource discovery and harvesting, and national defense. For example, this technology would allow the collection of data from underwater sensors in real-time. Comprehensive environmental monitoring is essential to effective climate modeling and the assessment of climate change.

2.2 Alternative Wireless Communication Technologies

There are two types of waves that can be used to carry information wirelessly subsea: Electromagnetic (EM) waves and acoustic waves. We argue that acoustic waves are the superior carrier and have the potential to meet the wireless communication needs of the subsea industry. We start by reviewing some of the properties of EM wave propagation underwater.

Salt water has a significantly higher electrical conductivity than air and attenuates EM waves substantially as they propagate. The level of attenu-

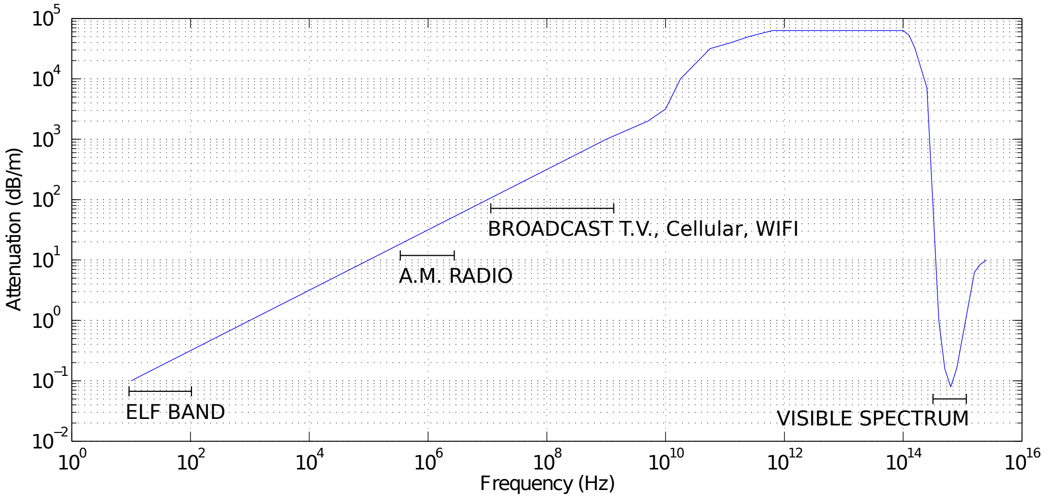

Figure 2.1: Attenuation of a plane electromagnetic wave in sea water as a function of frequency.

ation depends on frequency. Figure 2.1 sketches the attenuation of an EM plane wave in seawater for frequencies up to about $10^{16}Hz$ [9–12]. Only at frequencies below about $100Hz$ and in the visible spectrum is the attenuation low enough to allow useful penetration into the water column [9]. Note that the attenuation is greater than $30dB/m$ for all radio frequencies above $1MHz$. Inside the visible spectrum, blue-green light, around $480nm$ in wavelength, propagates with the least attenuation [13]. So-called extremely low frequency (ELF) waves are EM waves with frequencies below $100Hz$. These waves are still the only practical means to communicate wirelessly with a submerged vessel from land. The main drawback of ELF communication is the low bandwidth available and hence low achievable data rate of less than $1bps$ [14]. In typical seawater, a $100Hz$ EM wave is attenuated by $100dB$ after $323m$ and a $100kHz$ EM wave is attenuated by $100dB$ after only $8.8m$. At a range of $50m$, data rates of only about $300bps$ have been reached [14]. The company WFS sells RF underwater modems with an advertised data rate of $156kbps$ at a $3m$ range using 27 watts of power.

Free-space optical communication underwater has received renewed interest from researchers due to recent improvement in laser and LED technology [15, 16]. LEDs are low-cost and power-efficient light sources and their light intensity and switching speed have been shown to accommodate wireless underwater communication at $1Mbps$ over $100m$ [15]. The authors of this work report that transmissions were error free for ranges up to $100m$, but their data also shows that the error rate increases sharply at ranges beyond $100m$. The error rate reaches 0.5 at about $140m$ making reliable communication impossible. This is still a significant step up from RF communication. Several serious issues, however, limit the applicability of free-space optical communication in practice: First and perhaps most importantly, communication range is highly dependent upon water turbidity. The above values for light attenuation in water only hold for operation in pristine and transparent water. But near-shore and estuarine waters are typically highly turbid because of inorganic particles or dissolved organic matter from land drainage [17]. Light attenuation is exponential in distance. If, for a given wavelength λ, $I_0(\lambda)$ is the light intensity at the source, the light intensity $I(\lambda, z)$ at distance z from the source is described by the Beer-Lambert law [18]

$$I(\lambda, z) = I_0 e^{-c(\lambda)z} \tag{2.1}$$

The wavelength-dependent factor $c(\lambda)$ is the extinction coefficient of the water through which the optical system operates. For the type of light best suited for optical communication, blue-green light with a wavelength of $480nm$, the extinction factor is about $0.16m^{-1}$ for pristine ocean water and about $2.8m^{-1}$ for typical coastal waters [17]. The above mentioned experiment that proved the feasibility of error free optical underwater communication at $1Mbps$ over $100m$ was conducted in the clearest water - near the seafloor in the deep ocean [15], for which the authors measured the extinction coefficient to be $0.05m^{-1}$. According to Equation 2.1, the attenuation would have been $21.7dB$ at $100m$ distance in this clear-water environment. This suggests that in typical coastal water with an extinction coefficient of about $2.8m^{-1}$, this system would likely only manage a range of about $1.8m$. Note that the waters of most commercial interest, such as in the Gulf of Mexico or in the Irish sea, are highly turbid. Measurements in the Gulf of Mexico indicate that the extinction factor exceeds $3m^{-1}$ at many sites and can be as high as $5.1m^{-1}$ [19]. Turbidity is also high near underwater work and construction sites because sand and other particles are stirred up by operations. These are the spaces in which most underwater vehicles operate, and in which the need for wireless communication is greatest. Another issue of underwater optical communication is that different hardware

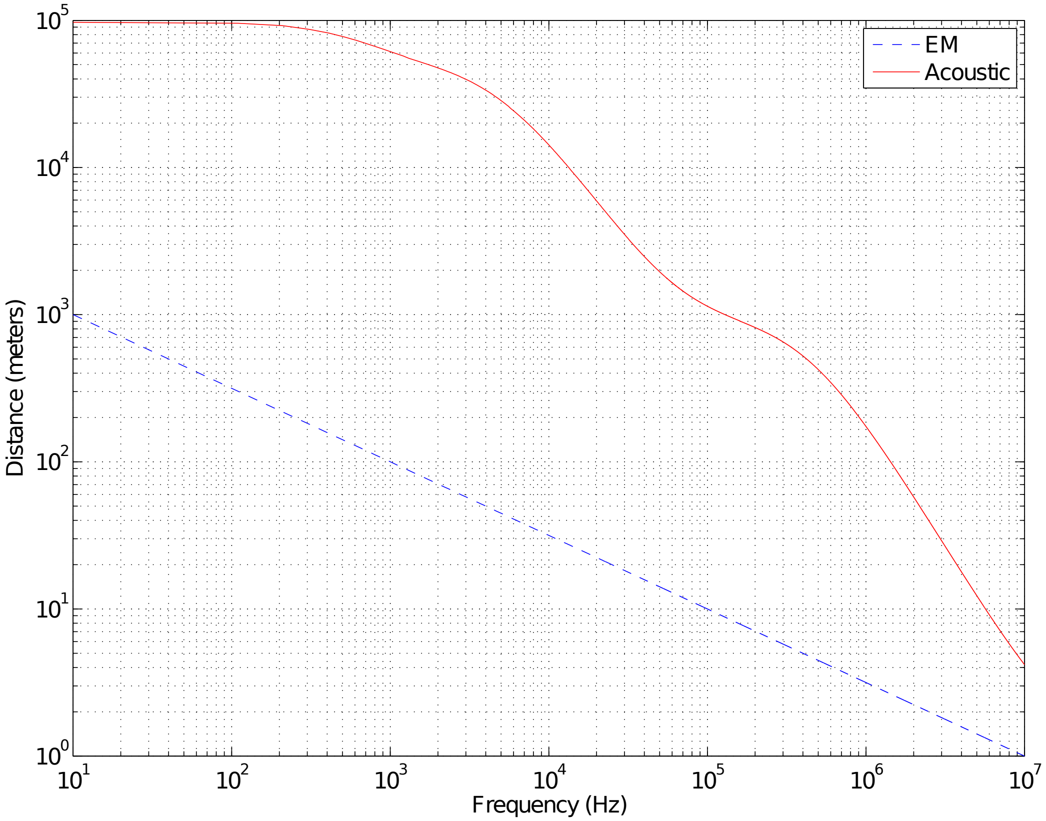

Figure 2.2: 100dB attenuation of an electromagnetic plane wave vs. an acoustic plane wave in sea water.

is needed for the emission and reception of light - LEDs for emission and a photo-multiplier tube for reception, for example. This roughly doubles the footprint of the complete system. Further, available light emission hardware such as LEDs and lasers are highly directional and require the transmitter and receiver to be aligned with each other. This is a major issue in mobile applications where the emitter would need to be constantly reaimed as the mobile platform moves through the water. In summary, high sensitivity to water turbidity, bulkiness and tight alignment requirements are major issues in free-space underwater optical communication and limit its applicability to cases where a clear line-of-sight path is available and alignment of transmitter and receiver is simple.

The only practical method of carrying information wirelessly undersea over distances greater than a couple of meters is through acoustic wave propagation. In seawater, acoustic waves are significantly less attenuated than radio waves. Figure 2.2 compares how far acoustic and radio waves can propagate

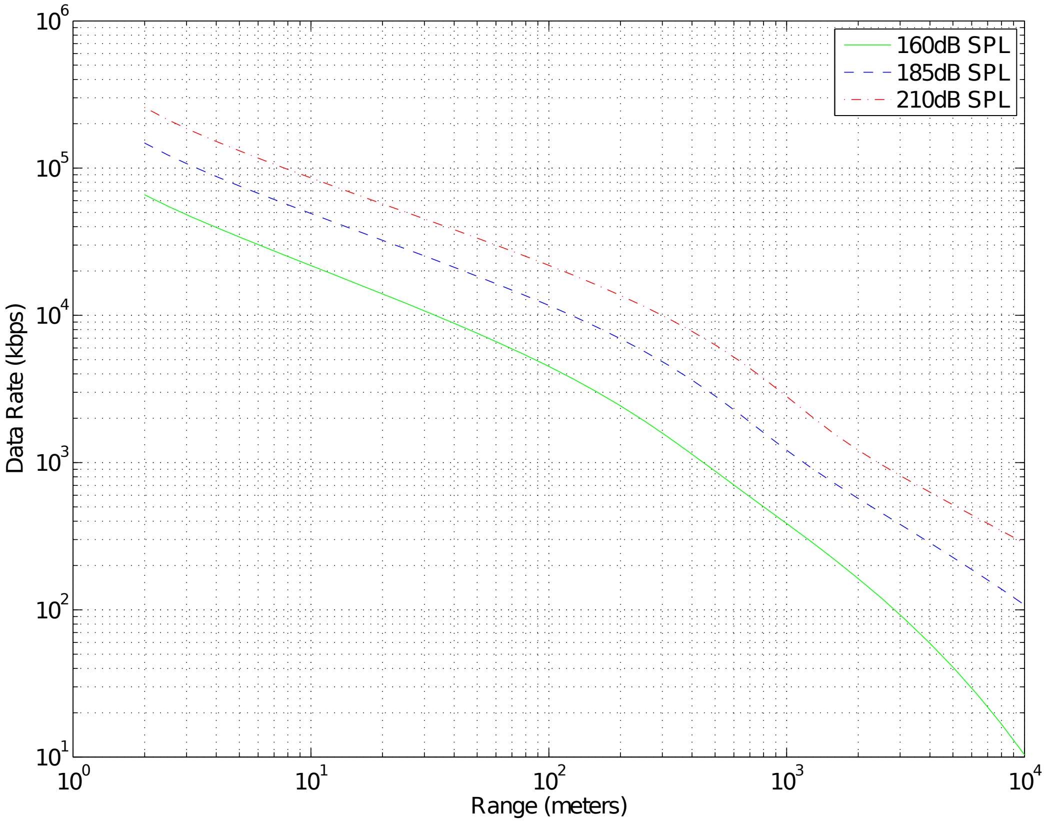

Figure 2.3: Information theoretic channel capacity of the underwater acoustic channel as a function of distance and source power.

through seawater until total attenuation reaches $100dB$. At a frequency of $1MHz$, radio waves are attenuated by $100dB$ at only about $3m$ of distance. At the same frequency, acoustic waves propagate for $200m$ until this level of attenuation is observed. These lower levels of attenuation allow acoustic communication systems to achieve much higher data rates than would be possible with underwater radio communication. Figure 2.3 shows the information theoretic capacity of the underwater acoustic channel for different levels of transmit power. The transmit power is given as sound pressure level (SPL) at one meter distance from the sound source. This plot does not include the frequency limitations imposed by commercially available acoustic sources but shows the potential of acoustic communication in general without the restrictions imposed by the limitations of today's hardware. At an SPL of $160dB$, a data rate greater than $4Mbps$ can be achieved at a range of $100m$. At an SPL of $210dB$ the data rate increases to more than $20Mbps$ for the same range. If the characteristics of available acoustic sources and sensors

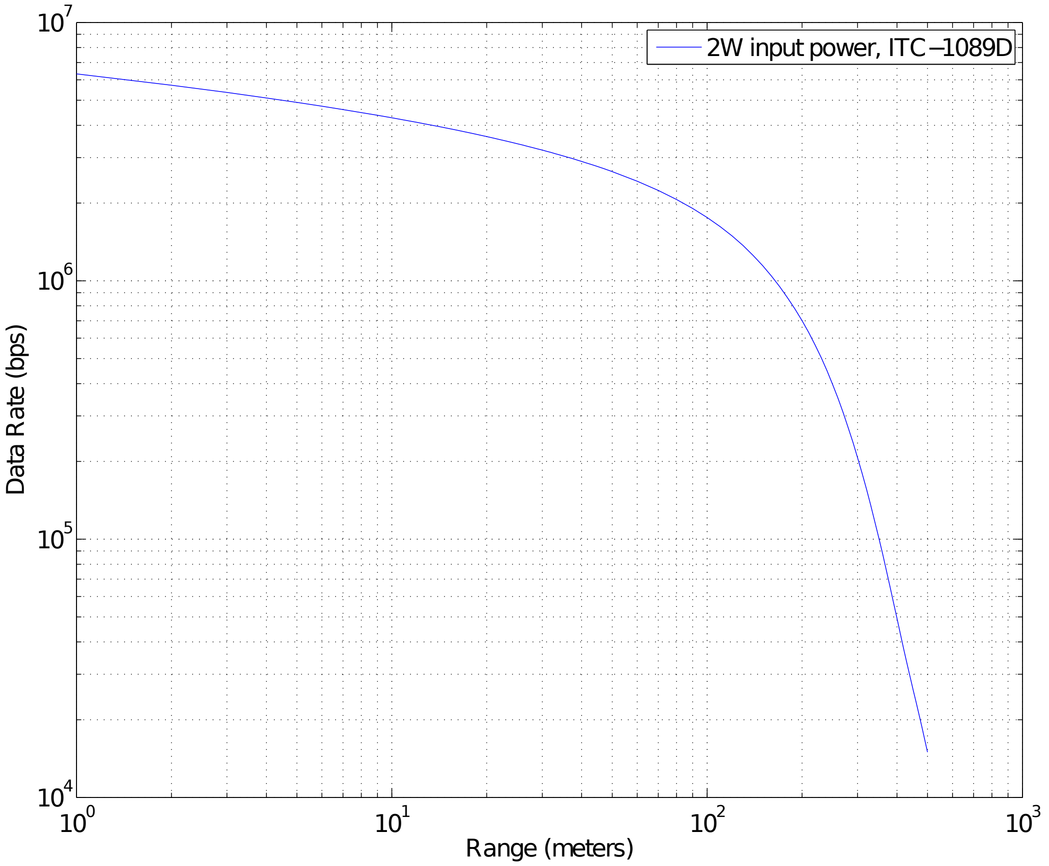

Figure 2.4: Information theoretic channel capacity between two ITC-1089D transducers using $2W$ of input power in sea water as a function of distance.

are taken into account data rates will drop, but they remain above $1Mbps$ at a range of $100m$. If an off-the-shelf transducer such as the ITC-1089D is used to emit and sense the acoustic signal, the channel capacity is about $1.75Mbps$ at $100m$ distance. Figure 2.4 illustrates the data rates that can be achieved with this transducer model at various ranges. These data rates are significantly higher than the achievable subsea radio communication data rates mentioned above.

Acoustic communication does not suffer from any of the issues of free-space optical communication and has significantly more range. Acoustic waves do attenuate more in turbid water than they do in clear seawater, but only marginally so. Acoustic attenuation depends on the concentration of particles suspended in the water. A mass concentration of $1kgm^{-3}$ is the extreme case for estuarine and coastal waters [20]. This level of concentration can, for example, be observed in shallow estuarine waters with strong turbulent tidal currents and a bed consisting of fine sand. At peak flow, mass concentrations close to $1kgm^{-3}$ have been measured [21]. For this level of concentration the attenuation of a $100kHz$ acoustic wave increases from $0.03dB/m$ for clear saltwater to $0.04dB/m$ [22]. Acoustic communication further does not require that the transmitter and receiver be aligned. Omnidirectional acoustic sources, such as the ITC-1089D transducer, are commercially available and remove the need for alignment. Also note that the same hardware - a ceramic electro-mechanical transducer - can be used for signal emission and reception. Because of these reasons, we view acoustic communication as strictly superior to EM wave-based communication underwater and hence focus on this technology in the remainder of this chapter.

The above capacity calculations ignored multi-path effects and assumed line-of-sight communication between stationary platforms. In this case, the underwater acoustic channel is well understood and can be modeled as a linear time-invariant (LTI) system with additive white Gaussian noise (AWGN) [23]. A line-of-sight between transmitter and receiver is often available underwater, but in a mobile communication scenario the assumption of stationary communication platforms is clearly invalid. There is no consensus on the statistical characterization of this type of time variability [23] and, “in the absence of good statistical models for simulation, experimental demonstration of candidate communication schemes remains a de facto standard.” In this chapter we shall introduce a novel channel model for mobile acoustic communication that builds upon the established physical principles of acoustic wave propagation and also derive communication algorithms from it that outperform all existing acoustic modems by several orders of magnitude.

Unlike in mobile radio systems on land, motion-induced Doppler effects cannot be neglected in acoustic communication systems. Remotely operated underwater vehicles (ROVs) typically move at speeds up to about $1.5m/s$ [24], autonomous underwater vehicles (AUVs) can run at speeds greater than $3m/s$ [25], modern submarines reach speeds greater than $20m/s$ [26,27], and supercavitating torpedoes propel to speeds of up to 100 m/s [27]. This leads to underwater acoustic Mach numbers v/c (v = vehicle velocity projected onto the signal path between transmitter and receiver, c = wave propagation speed in the medium) on the order of 10^{-2} and higher. In comparison, the world’s fastest train in regular commercial service - the Transrapid magnetic levitation train - operates at a top speed of $430km/h$ [28]. At this speed, the radio communication channel experiences a Mach number of only $4 * 10^{-7}$, i.e., five orders of magnitude smaller. Relative motion between the transmitter and receiver always manifests as time-varying temporal scaling of the received waveform. In radio channels, such Doppler effects are minimal and are easily correctable under the popular narrowband assumption, while in acoustic communications, they can be catastrophic if not compensated dynamically. Further, when acoustic communication signals have multiple interactions with scatterers underwater, such as the surface or the ocean bottom, harsh multi-path arises. There are several acoustic modems on the market that provide a transparent data link and can reach a net data rate of about $2.5kbps$ over $1km$ distance, but when they are mobile or multiple signal paths to the receiver exist due to reflective boundaries nearby, these modems perform poorly and only achieve a net data rate of about $100bps$ [29, 30]. Multi-path effects are typically most severe when communication signals propagate through a wave guide or in shallow water where both the surface and the bottom reflect the acoustic signal multiple times. Note that horizontal long range communication basically always occurs in a waveguide because waves are always refracted towards the horizontal layer of water at which the speed of sound is lowest. This phenomenon has been described as the Sound Fixing and Ranging (SOFAR) channel and was first discovered in the 1940s [31].

2.3 Related Work

The first underwater acoustic modems employed frequency-shift keying (FSK) which maps digital information to a sequence of discrete tones. Guard intervals between consecutive tones ensure that reverberation does not correlate them and guard bands guarantee that Doppler shifts do not cause misinterpretation at the receiver. FSK is relatively easy to implement and can be made to be robust but then uses the available time and frequency resources rather inefficiently. Underwater modems using this technique typically have a data rate less than $1kbps$ [32]. In the 1990s, it was shown that acoustic wave propagation allows phase-coherent digital communication underwater [33]. The authors combined an adaptive linear decision feedback equalizer (DFE) and a phase locked loop (PLL) to combat the channel distortion due to reverberation and Doppler effects. This system was then evaluated on data from at-sea experiments and the authors demonstrated a data rate of $10kbps$ in shallow water over $3.7km$ distance using $5kHz$ of acoustic bandwidth, a stationary $183dB$ SPL source and a stationary directional receiving element (hydrophone). This work indicated that coherent communication had the potential to significantly improve data rate and bandwidth efficiency. Note, however, that the directional hydrophone required alignment with the source and that no platform mobility was allowed. The directional hydrophone helped reject the noise generated at the surface due to wind and wave motion and also limited reverberation since the multi-path components with most delay generally impinge on the hydrophone at the widest angle. In practice, neither hydrophone alignment nor platform stability can be guaranteed. In a later follow-up paper researchers including the author of [33] recognize that the communication system devised in [33] cannot handle the level of Doppler introduced by standard mobile platforms such as autonomous underwater vehicles (AUVs) and that "its performance has been unsatisfactory under realistic field conditions" [29]. They extend the original approach and propose a two-step detection algorithm. For each received data block, the detector first obtains an estimate of the average Doppler factor over the entire transmission and then resamples (interpolates) and phase corrects the demodulated base-band signal based on this factor. In the second step, the original method from [33] is used to estimate the sent data symbols. The phase-locked loop (PLL) is employed to remove any residual Doppler distortion from the demodulated signal and the adaptive equalizer estimates the transmitted symbols from the Doppler compensated signal. They claimed to achieve a data rate of $2.5kbps$ on data from moving platforms at relative speeds up to 6 knots but did not specify over what distance. Even this extended approach, however, only works if the Doppler variation is sufficiently small and roughly constant for the duration of a block. The 'micromodem' of the Woods Hole Oceanographic Institution (WHOI), like most state-of-the-art systems, implements this algorithm for coherent communication and also offers a robust low data rate frequency-shift keying with frequency-hopping (FH-FSK) mode [29, 30, 34, 35]. The WHOI micromodem was the basis for a US Navy submarine deployment in the mid 1990s and the technology in the research and commercial community have not changed substantially in the interim. In a more recent paper, WHOI reports that for communication with AUVs the micromodem relies on its “robust FH-FSK modulation and error correction coding (ECC) scheme to communicate at long ranges ($2-4$ kilometers), in the very shallow water zone” at a data rate of $80bps$ using $4kHz$ of bandwidth and a powerful 190dB SPL source [30]. This performance corresponds to a bandwidth efficiency of only $0.02bps/Hz$.

There are many other research papers discussing extensions of the algorithm proposed in [33] and the data rates that these extensions achieve in at-sea field-tests. In [36], multiple transmitters and space-time trellis codes are used to capitalize on the benefits of the transmit diversity available in the reverberant horizontal shallow water acoustic communication channel. The highest data rate the authors could reliably achieve is $40kbps$ at a bit error rate (BER) of about 10^{-2} using four transmitters and $23kHz$ of bandwidth. The transmit and receive array were stationary and $2km$ apart. The source power level was set to $190dB$. The system suggested in [36] uses complex hardware and heavy software but only gives a bandwidth efficiency of $0.375bps/Hz$ per transmitter. In [37], two transducers were mounted onto the ends of a $10m$ pole which was then vertically submerged. The authors achieved a data rate of $150kbps$ using $25kHz$ bandwidth with an unspecified transmit power. This translates to a bandwidth efficiency of $6bps/Hz$. Note, though, that in both of these works the transmitter and the receiver were stationary which considerably simplified the conducted experiments. Motion-induced Doppler effects would have severely degraded the performance of the proposed algorithms.

There is another line of underwater acoustic communication research that investigates the use of orthogonal frequency-division multiplexing (OFDM). OFDM fundamentally assumes that the channel is linear and time invariant for the length of each ODFM symbol. Platform motion and environmental fluctuations make the underwater acoustic channel highly time-variant and the application of standard OFDM leads to communication algorithms that break down when transmitter or receiver are mobile. Several ad hoc modifications to the original OFDM receiver algorithm have been suggested and tested in at-sea field-tests [38–42]. One OFDM modification [39] essentially precedes the regular OFDM receiver with the first step of the receiver algorithm proposed in [29]. But again, this type of Doppler compensation assumes that Doppler variation is sufficiently small and roughly constant for the duration of an OFDM symbol. Since this approximation improves with shorter OFDM symbols, short OFDM symbols of a length of only 512 to 2048 carriers are used. At the same time the underwater acoustic channel is highly reverberant and long cyclic prefixes or zero padding is necessary between consecutive OFDM symbols to eliminate intersymbol interference. This means that, during a significant fraction of time, no information can be sent, leading to low achievable data rates. In [39], a data rate of 9.7kbps is achieved at a BER of 10^{-2} using a bandwidth of $12kHz$ and 2048 carriers over a distance somewhere between $50m$ and $800m$. Taking into account the additional layer of channel coding necessary to reduce the BER to below 10^{-9}, the bandwidth efficiency of this system is $0.7275bps/Hz$ at best, assuming a capacity achieving code. Some of the authors of this paper tried to commercialize this technology (AquaSeNT) but were unable to turn its SBIR funding into a commercial product. This likely failed due to motion-induced Doppler effects under which the OFDM carriers are no longer orthogonal and severe inter-carrier-inference (ICI) arises. Another team out of the University of Florida ran into similar troubles with an STTR joint with EdgeTech, citing motion-induced Doppler effects as the fundamental stumbling block.

The fatal flaw of many of these works is that the channel model is borrowed from the radio communication community and only slightly modified, if at all, and hence does not properly respect the physics of acoustic wave propagation. A popular assumption is that the Doppler is constant over the time of a data block and the remaining channel effect is linear and time-invariant, but in reality the Doppler can be highly time-varying and different wave propagation paths can experience different Doppler.

In our work, we have developed a sample-by-sample, recursive resampling technique, in which time-varying Doppler is explicitly modeled, tracked and compensated. Integrated into an iterative turbo equalization based receiver, this novel Doppler compensation technique has demonstrated unprecedented communication performance in US Navy sponsored field tests and simulations. Some of our field data stems from the MACE10 experiment conducted in the waters 100 km south of Martha's Vineyard, MA. Under challenging conditions (harsh multi-path, ranges up to 7.2 km, SNRs down to 2 dB and relative speeds up to 3 knots) our algorithms sustained error-free communication over the period of three days at a data rate of $39kbps$ at $2.7km$ distance and a data rate of $23.4kbps$ at $7.2km$ distance using a 185dB source. In this experiment we had used only $9.76kHz$ of acoustic bandwidth leading to bandwidth efficiencies of $3.99bps/Hz$ and $2.40bps/Hz$, respectively. Compared to frequency-shift keying with frequency-hopping (FH-FSK) with a bandwidth efficiency of $0.02bps/Hz$, which is the only existing acoustic communication method robust enough to handle these conditions, this implies an improvement of two orders of magnitude in data rate and bandwidth efficiency.

Since our interaction and discussions with the subsea oil and gas industry, we have begun to focus on communication over shorter distances while scaling up bandwidth and data rate. In a 1.22m x 1.83m x 49m wave-tank, we have begun to experiment with a set of ITC-1089D transducers, which have around $200kHz$ of bandwidth at a center frequency of around $300kHz$. We recently achieved $1.2Mbps$ over a distance of $12m$ using this experimental setup. In a smaller tank, we reached rates of $120Mbps$ over distances of less than $1m$. These are to the best of our knowledge by far the highest data rates ever recorded for acoustic underwater communication.

The underwater acoustic channel remains one of the most difficult communication channels [23,43] and our understanding of it is still in its infancy. The Sections 2.5, 2.6, 2.7, 2.8 and 2.9 will review the physical properties of acoustic wave propagation and introduce a novel channel model derived from the acoustic wave equation. Section 2.10 discusses the interesting connection between underwater acoustic positioning and underwater acoustic communication. Finally, in Section 2.11 we derive an efficient receiver algorithm based upon the introduced new channel model and we show its superior performance in simulations, laboratory experiments and at-sea field-tests in Section 2.12.

2.4 Notation

We will typeset vectors and sequences bold-face. The set of integers is denoted by $\mathbb{Z}$ and $\mathbb{Z}_+ = \{z \in \mathbb{Z} : z \geq 0\}$. The sets of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. The set $\mathbb{R}_> = \{x \in \mathbb{R} : x > 0\}$ and $\mathbb{R}_\geq = \{x \in \mathbb{R} : x \geq 0\}$. The sets $\mathbb{R}_<$ and $\mathbb{R}_\leq$ are defined analogously. The set $[j : n]$ denotes $\{z \in \mathbb{Z} : j \leq z \leq n\}$ with $[n] \equiv [1 : n]$. For any complex number x, $x^\star$ denotes the conjugate of x. For any function $x : \mathbb{R} \to \mathbb{R}$, the function $\dot{x}(t)$ denotes its first derivative and $x^{(k)}(t)$ its k-th derivative. The real number $\|\boldsymbol{x}\|$ denotes the Euclidean norm of the vector $\boldsymbol{x}$. When $\boldsymbol{A}$ is a matrix of dimension $n \times m$, then $A_{[i:j],[l:k]}$ denotes the matrix $\boldsymbol{B}$ of dimension $1 + j - i \times 1 + k - l$ where $B_{p,q} = A_{i-1+p,l-1+q}$.

2.5 Physical Modeling of the Problem

By definition, acoustic communication uses acoustic waves to carry information. To communicate digital information acoustically, a digitized waveform is converted into an electrical signal by a suitable waveform generator circuit and this electrical signal is then amplified and delivered to an acoustic transducer. The electrical signal stimulates the transducer to vibrate. The resulting pressure fluctuations in the medium create an acoustic signal that radiates off the transducer and propagates through the water. The transducer is typically a piezo-electric ceramic encapsulated in plastic. This type of transducer can be used for both the transmission and the reception of acoustic signals. It converts electrical signals into acoustic signals and vice versa. When a transducer is used for transmission, it is often referred to as a projector. When it is used as a receiver, it is usually called a hydrophone. At some distance from projector, the hydrophone is stimulated by the incident pressure fluctuations and generates an electrical signal. The measured electrical signal is amplified and digitized by another suitable circuit.

Given a point of reference, the position and orientation of a transducer array are uniquely determined by a six dimensional vector describing the translation in three perpendicular axes combined with the rotation about three perpendicular axes, the six degrees of freedom (6DoF). We propose a channel model that explicitly models these states for the transmit and the receive array. Figure 2.5 sketches a transmit array at position $\boldsymbol{x}_i$ with orientation $\boldsymbol{\theta}_i$ and a receive array at position $\boldsymbol{x}_l$ with orientation $\boldsymbol{\theta}_l$. When there are multiple acoustic signal paths from the transmitting array to the receive array due to reflection off nearby boundaries, each propagation path is modeled as a line of sight path from a phantom source with its own position and orientation. Figure 2.6 illustrates this idea. The p-th phantom source appears to be at position $\boldsymbol{x}_{i;p}(t)$ with orientation $\boldsymbol{\theta}_{i;p}(t)$. Along each path, some dispersion is induced due to the frequency dependent absorption loss. Each 6DoF vector, as well as the attenuations along each path, will

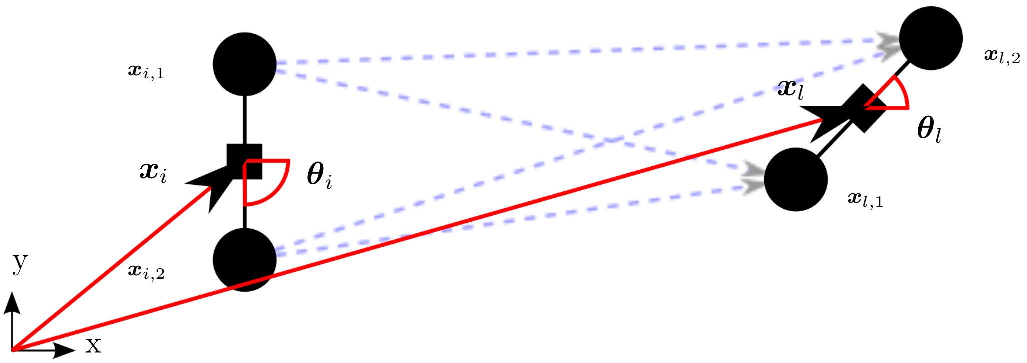

Figure 2.5: A transmit array at position $\boldsymbol{x}_i$ with orientation $\boldsymbol{\theta}_i$ and a receive array at position $\boldsymbol{x}_l$ with orientation $\boldsymbol{\theta}_l$.

be modeled as a continuous time random process. These states are observed through the acoustic pressure measurements of the receive hydrophone arrays and also possibly through inertial sensors mounted onto the transmit and receive array. Inference based on this model yields position estimates and if the sent signals are used for communication and are unknown at the receiver, they can be modeled as random processes and be estimated as well. The receiver then performs positioning and data detection jointly. There is also an interesting connection with beam-forming. Emitted wavefronts may arrive at different times on the elements of the receiver array. The receiver algorithm we shall propose essentially obtains estimates of these arrival times and then compensates the received signals such that they add constructively - a technique similar to broadband receive beam-forming. Another interesting idea is to perform transmit beam-forming based upon the known location of the receiver. This has the potential to mitigate multi-path in short range channels.

Our goal is to establish a model of the acoustic channel that is sophisticated enough to capture the dominant physical effects but simple enough to allow computationally tractable inference. We begin from first principles of acoustic wave propagation.

As a first step, let us consider the acoustic signal path starting at the projector array and ending at the receive hydrophone array as our communication channel. We will also assume for a moment that there is only one transducer element on the transmit and receive array and that their positions are $\boldsymbol{x}_1(t)$ and $\boldsymbol{x}_2(t)$, respectively, which depend on the time t. The transmitter emits the acoustic signal $\tilde{s}_1(t)$ and the receiver senses the acoustic

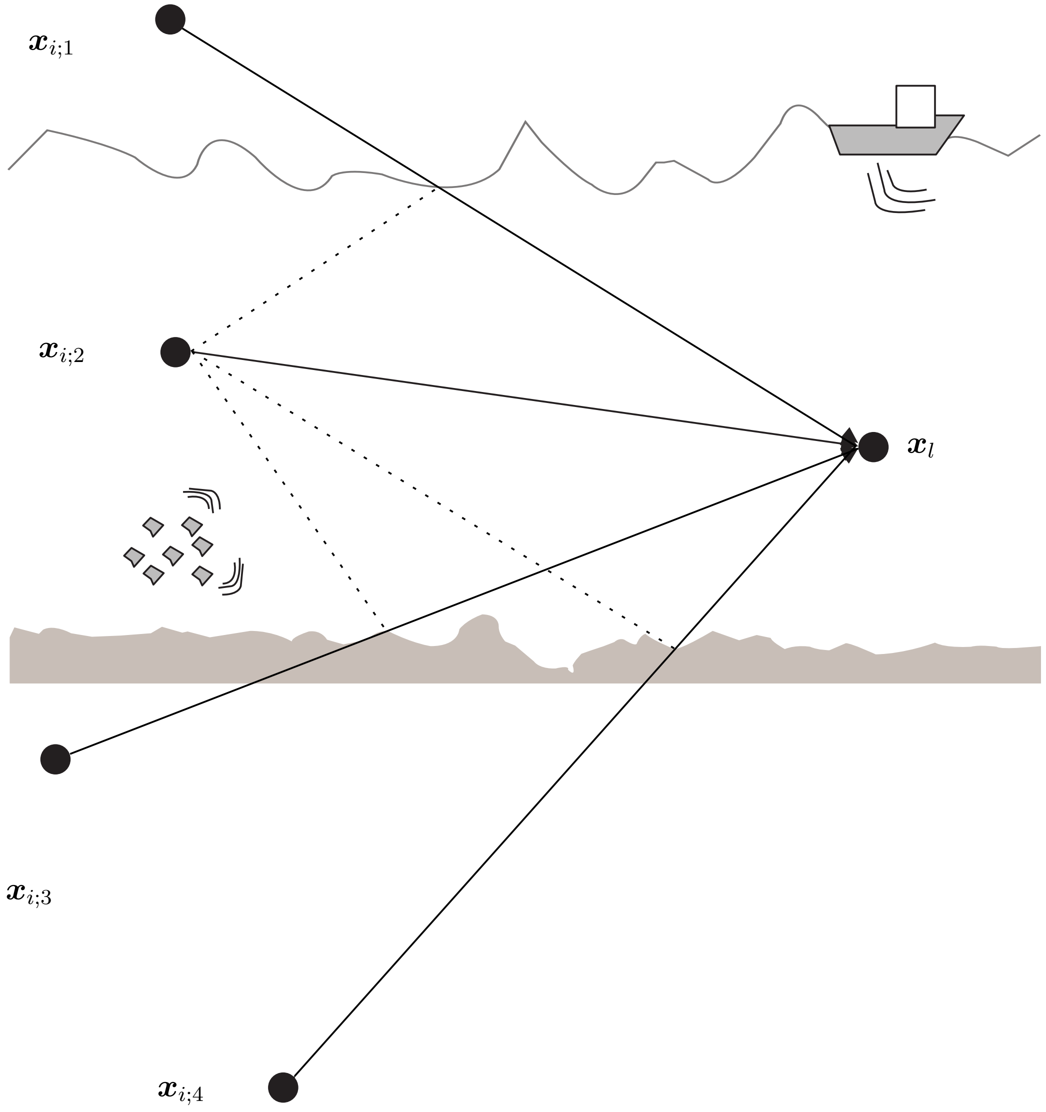

Figure 2.6: Multi-path effects: Each path interpreted as a line of sight path to a phantom source.

signal $\tilde{r}_2(t)$. If these elements were operating in an ideal fluid, where energy was conserved and there was no absorption loss and no ambient noise, the acoustic wave equation completely describes the channel:

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} - \triangle p = 4\pi \frac{\partial}{\partial t}\left\{\delta(\boldsymbol{x} - \boldsymbol{x}_1(t)) \int_{-\infty}^{t} \tilde{s}_1(\tau) d\tau \right\} \tag{2.2}$$

where $p(\boldsymbol{x}, t)$ is the sound pressure at position $\boldsymbol{x}$ and time t, c is speed of sound and $\triangle$ denotes the Laplace operator [44, 45]. Assuming there are no reflective boundaries and both transmitter and receiver move subsonically, the far field solution to this equation at position $\boldsymbol{x}_2(t)$ is

$$p^{FF}(\boldsymbol{x}_2(t), t) = \frac{\left(\frac{\partial t_e}{\partial t}\right)^2}{||\boldsymbol{x}_2(t) - \boldsymbol{x}_1(t_e)||}\tilde{s}_1(t_e) \tag{2.3}$$

where t_e is the unique solution to the implicit equation

$$t - t_e - \frac{||\boldsymbol{x}_2(t) - \boldsymbol{x}_1(t_e)||}{c} = 0 \tag{2.4}$$

[45]. The time t_e is often called the emission time or retarded time. Neglecting the near field component of the solution, we set $\tilde{r}_2(t) = p^{FF}(\boldsymbol{x}_2(t), t)$. This relationship completely describes the communication channel under the mentioned assumptions. We can write

$$\tilde{r}_2(t) = h(t)\tilde{s}_1(t_e) \tag{2.5}$$

and consider $h(t)$ a time dependent channel gain factor. Taking a close look at Equation 2.3, we notice that the gain $h(t)$ is inversely proportional to the communication distance. Further the "Doppler factor" $\frac{\partial t_e}{\partial t}$ is always positive, equal to unity when there is no motion, greater than unity when the source and receiver are moving towards each other and smaller than unity otherwise.

The solution t_e to Equation 2.4 can be interpreted as a fixed-point and can be computed by a fixed-point iteration algorithm.

Theorem 1. *Assume there are two functions $\dot{\boldsymbol{x}}_1(t) : \mathbb{R} \to \mathbb{R}^3$ and $\dot{\boldsymbol{x}}_2(t) : \mathbb{R} \to \mathbb{R}^3$, and that $\dot{\boldsymbol{x}}_1(t)$ is continuously differentiable and $||\dot{\boldsymbol{x}}_1(t)|| < c$.*

Define the function

$$F_t(t_e) = t - \frac{1}{c}||\boldsymbol{x}_2(t) - \boldsymbol{x}_1(t_e)|| \tag{2.6}$$

Then for any t and $t_e[0]$, the sequence $t_e[n], n = 0, 1, 2, ...$ with

$$t_e[n+1] = F_t(t_e[n]), n = 0, 1, 2, ... \tag{2.7}$$

converges to a real number $t_e(t)$. This number is the unique solution to the implicit equation $t_e = F_t(t_e)$, which is equivalent to Equation 2.4.

Proof. We know $f(\boldsymbol{x}) = ||\boldsymbol{x}||$ is a continuous function and derive

$$\begin{aligned} \frac{d}{dt}||\boldsymbol{x}_1(t)|| &= \lim_{\delta \to 0} \frac{||\boldsymbol{x}_1(t+\delta)|| - ||\boldsymbol{x}_1(t)||}{\delta} && (2.8) \\ &\le \lim_{\delta \to 0} \left\| \frac{\boldsymbol{x}_1(t+\delta) - \boldsymbol{x}_1(t)}{\delta} \right\| && (2.9) \\ &= \left\| \lim_{\delta \to 0} \frac{\boldsymbol{x}_1(t+\delta) - \boldsymbol{x}_1(t)}{\delta} \right\| && (2.10) \\ &= \|\dot{\boldsymbol{x}}_1(t)\| && (2.11) \end{aligned}$$

and

$$\begin{aligned} -\frac{d}{dt}||\boldsymbol{x}_1(t)|| &= \lim_{\delta \to 0} \frac{-||\boldsymbol{x}_1(t+\delta)|| + ||\boldsymbol{x}_1(t)||}{\delta} && (2.12) \\ &\le \lim_{\delta \to 0} \left\| \frac{\boldsymbol{x}_1(t+\delta) - \boldsymbol{x}_1(t)}{\delta} \right\| && (2.13) \\ &= \|\dot{\boldsymbol{x}}_1(t)\| && (2.14) \end{aligned}$$

for any $t \in \mathbb{R}$. The inequalities follow from the triangle inequality. So $|\frac{d}{dt} \|\boldsymbol{x}_1(t)\| \, | \le \|\dot{\boldsymbol{x}}_1(t)\|$. Further,

$$\begin{aligned} \left| \frac{d}{dt_e} F_t(t_e) \right| &= \frac{1}{c} \left| \frac{d}{dt_e} \|\boldsymbol{x}_2(t) - \boldsymbol{x}_1(t_e)\| \right| && (2.15) \\ &\le \frac{1}{c} \|\dot{\boldsymbol{x}}_1(t_e)\| < 1 && (2.16) \end{aligned}$$

The function $F_t(t_e)$ is hence a contraction mapping in t_e. By the Banach fixed-point theorem [46], there exists an unique t_e that solves the equation $F_t(t_e) = t_e$ and the sequence $t_e[n], n = 0, 1, 2, ...$ converges to this solution. Obviously, the implicit equation $F_t(t_e) = t_e$ is equivalent to Equation 2.4. □

We had assumed the absence of absorption in the derivation of Equation 2.5. In reality, however, emitted acoustic signals experience attenuation due to spreading and absorption, i.e., thermal consumption of energy. The absorption loss of acoustic signals in sea water increases exponentially in distance and super exponentially in frequency. The loss due to spreading is in principle the same as in electromagnetics. The total attenuation of the signal power is given by

$$A(l,f) = \frac{|\tilde{S}_1(f)|^2}{|\tilde{R}_2(f)|^2} = l^k a(f)^{l-1} \tag{2.17}$$

where f is the signal frequency, l is the transmission distance and $\tilde{S}_1(f)$ and $\tilde{R}_2(f)$ are the Fourier transforms of the signals $\tilde{s}_1(t)$ and $\tilde{r}_2(t)$, respectively. The exponent k models the spreading loss. If the spreading is cylindrical or spherical, k is equal to 1 or 2, respectively. Several empirical formulas for the absorption coefficient $a(f)$ have been suggested [47–52]. Marsh and Schulkin [47] conducted extensive field experiments and derived the following empirical formula to approximate $10\log_{10} a(f)$ in sea water at frequencies between $3kHz$ and $0.5MHz$:

$$10\log_{10} a(f) \approx 8.68 \cdot 10^3 \left(\frac{SAf_T f^2}{f_T^2 + f^2} + \frac{Bf^2}{f_T} \right) (1 - 6.54 \cdot 10^{-4} P) \quad [\text{dB/km}] \tag{2.18}$$

where $A = 2.34 \cdot 10^{-6}$, $B = 3.38 \cdot 10^{-6}$, S is salinity in promille, P is hydrostatic pressure [kg/cm^2], f is frequency in kHz and

$$f_T = 2.19 \cdot 10^{6-1520/(T+273)} \tag{2.19}$$

is a relaxation frequency [kHz], with T the temperature [°C] [53]. Figure 2.7 is a composite plot using the formulas from [47] and [52] and illustrates the dependency of $10\log_{10} a(f)$ on frequency for a salinity of 35 promille, a temperature of $5°C$ and a depth of $1000m$. It becomes evident that the bandwidth available for communication is severely limited at longer distances. For shorter distances, the bandwidth of the transducer becomes the limiting factor. A $1MHz$ sine wave experiences a $31.89dB$ absorption loss over $100m$ distance and a $318.9dB$ absorption loss over $1km$ distance. From Equation 2.17, we see that for a fixed transmission distance l, signal attenuation is

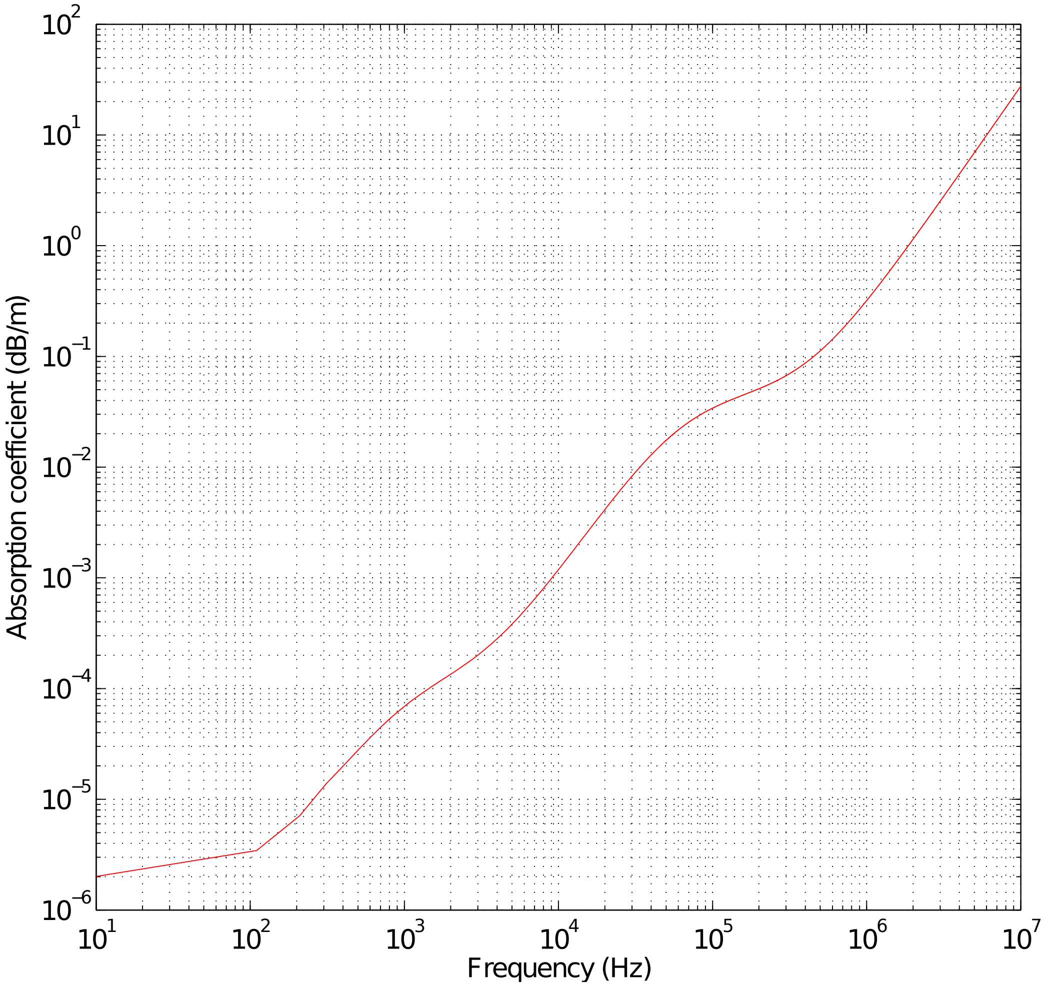

Figure 2.7: Absorption coefficient, $10\log_{10} a(f)$ in dB/m.

linear and time-invariant. When the transmitter or the receiver move, signal attenuation is still a linear effect, but it varies with time. The received acoustic signal can hence be related to the emitted acoustic signal by a time-varying convolution integral with kernel $h(t, \tau)$. We suggest the following extension to the channel model from Equation 2.5 to take this time-varying signal attenuation into account:

$$\tilde{r}_2(t) = \int_\tau h(t, \tau)\tilde{s}_1(t_e(t) - \tau)d\tau \tag{2.20}$$

Acoustic channel observations in reality also always contain some noise. There is ambient noise and site-specific noise. Site-specific noise is for example caused by underwater machines or biologics. Ambient noise arises from wind, turbulence, breaking waves, rain and distant shipping. The ambient noise can be modeled as a Gaussian process but has a colored spectrum [54]. At low frequencies $(0.1 - 10Hz)$, the main sources are earthquakes, underwater volcanic eruptions, distant storms and turbulence in the ocean and atmosphere. In the frequency band $50 - 300Hz$, distant ship traffic is the dominant noise source. In the frequency band $0.5-50kHz$ the ambient noise is mainly dependent upon the state of the ocean surface (breaking waves, wind, cavitation noise). Above $100kHz$, molecular thermal noise starts to dominate [53]. The power spectral density of the ambient noise has been measured and modeled by many researchers [55–58]. Coates [56] breaks the overall noise spectrum $N(f)$ up into a sum of four components: The turbulence noise $N_t(f)$, the shipping noise $N_s(f)$, surface agitation noise $N_w(f)$ and the thermal noise $N_{th}(f)$. These noise spectra are given in $\mu Pa^2/Hz$ as a function of frequency in kHz

$$10\log_{10} N_t(f) = 17 - 30\log_{10}(f) \tag{2.21}$$

$$10\log_{10} N_s(f) = 40 + 20(s - 0.5) + 26\log_{10}(f) - 60\log_{10}(f + 0.03) \tag{2.22}$$

$$10\log_{10} N_w(f) = 50 + 7.5w^{1/2} + 20\log_{10}(f) - 40\log_{10}(f + 0.4) \tag{2.23}$$

$$10\log_{10} N_{th}(f) = -15 + 20\log_{10}(f) \tag{2.24}$$

and sum up to give the total ambient noise $N(f)$

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \tag{2.25}$$

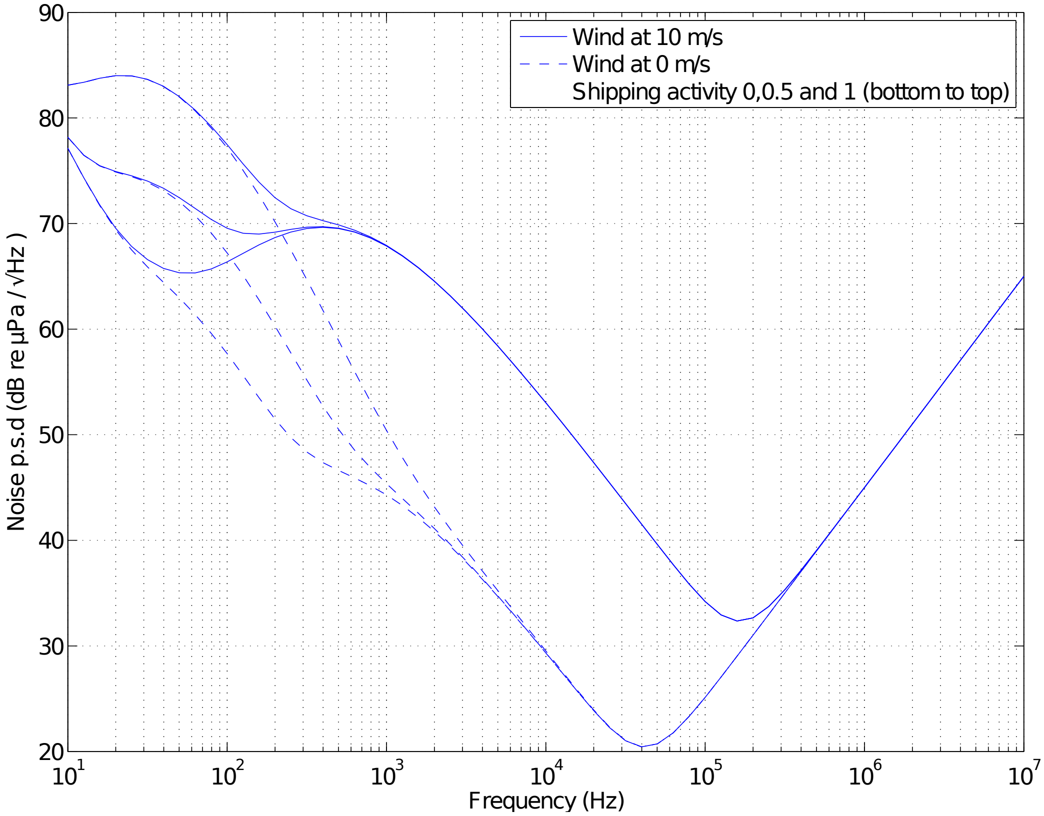

Figure 2.8: Power spectral density of the ambient noise, $N(f)$, in $(dB\, re\, \mu Pa/\sqrt{Hz})$.

In this empirical expression, s is the shipping activity factor taking values between 0 and 1 and w is the wind speed in m/s. Figure 2.8 is reproduced from [59] and plots $N(f)$ for different values of s and w. The ambient noise and the signal originating from the transmitter add at the receiver. Defining $\tilde{v}(t)$ to be an independent Gaussian random process with power spectral density given by $N(f)$, the channel model from Equation 2.20 can be further refined to

$$\tilde{r}_2(t) = \int_\tau h(t,\tau)\tilde{s}_1(t_e(t) - \tau)d\tau + \tilde{v}(t) \qquad (2.26)$$

So far we considered the acoustic signal path starting at the projector and ending at the receive hydrophone as our channel. But in reality, the involved transducers and amplifiers also shape the signal and introduce noise. We will hence now extend our notion of the communication channel to encompass the distortion effects of the involved amplifiers and transducers as well. The effect of any frequency response shaping can readily be absorbed into the kernel

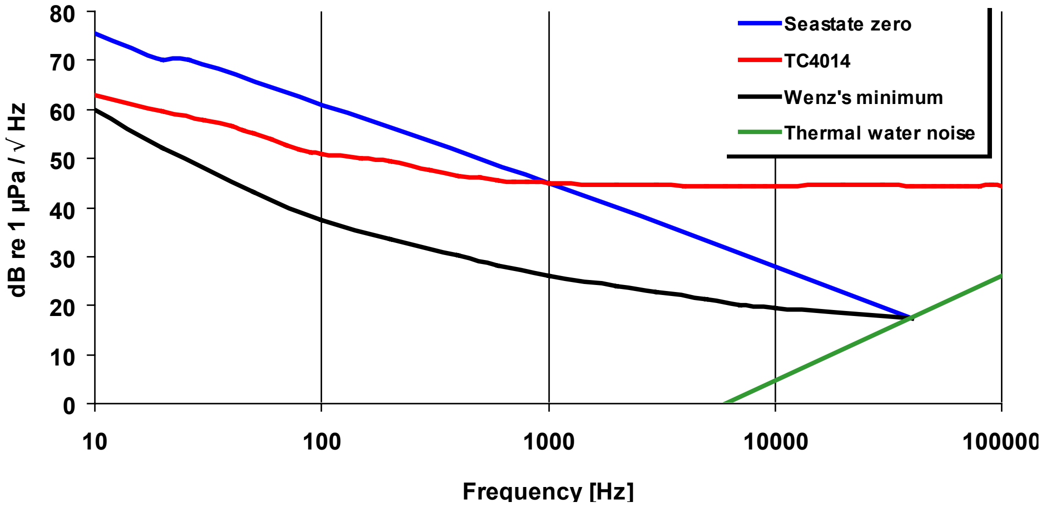

Figure 2.9: Typical self-noise referred to input of the Reson TC4014 broad band spherical hydrophone reproduced from [60].

$h(t, \tau)$. But at the receiver also significant electronic noise is added. The voltage generated by a hydrophone in response to an incident acoustic signal is small and needs to be preamplified to better match the voltage range of the digitizer. The electronic noise produced at the input stage of the preamplifier depends upon the capacitance of the hydrophone, but is usually so high that it dominates the acoustic ambient noise picked up by the hydrophone. The most sensitive high frequency hydrophones by market leading companies ITC and RESON introduce self-noise of at least $45dB\ re\ \mu Pa/\sqrt{Hz}$ referred to input. Figure 2.9 shows the typical self-noise referred to input of the Reson TC4014 broadband spherical hydrophone and compares it to seastate zero ambient noise, i.e. the ambient noise when wind waves and swell levels are minimal. Comparing Figures 2.8 and 2.9, we notice that even for high levels of wind, the hydrophone self-noise dominates the ambient noise at frequencies above about $20kHz$. Since the acoustic projectors most suited for broadband communication do not cover frequencies below about $10kHz$, we will assume that the electronic noise dominates the ambient noise in our further analysis. The electronic noise is well approximated by an independent Gaussian noise process with flat power spectral density in the band of interest and we will hence now assume that $\tilde{v}(t)$ is such a process.

Next, we will model transmission involving transmit and receive arrays with multiple transducers and consider multi-path effects arising from reflections off nearby scatterers.

We fix a Cartesian frame of reference at a known location in space. All positions and angles are given with respect to this reference system. Assume $\boldsymbol{x}_i(t)$ and $\boldsymbol{\theta}_i(t)$ are the three-dimensional position and orientation vectors of the i-th transducer array. The total number of available arrays depends on the scenario, but we will always start indexing them with the integer 1. We will have two types of arrays: A trivial array with only one element and a non-trivial array with K elements and fixed geometry. There is a function $T : \mathbb{R}^6 \to \mathbb{R}^{3\times K}$ that maps the position $\boldsymbol{x}_i(t)$ and orientation $\boldsymbol{\theta}_i(t)$ of the i-th array to the positions $\boldsymbol{x}_{i,j}(t)$, $j \in [K]$, of its omnidirectional elements. Figure 2.5 applies this notation.

The j-th transducer of the i-th array sends the signal $\tilde{s}_{i,j}(t)$ and receives $\tilde{r}_{i,j}(t)$. We assume there is no multiple access interference (MAI). So, in case there is no multi-path but only a line of sight, the received signals can be expressed as

$$\tilde{r}_{l,m}(t) = \sum_j \int_\tau h_{i,j;l,m}(t,\tau)\tilde{s}_{i,j}(t_{i,j;l,m}(t) - \tau)d\tau + \tilde{v}_{l,m}(t) \tag{2.27}$$

where $h_{i,j;l,m}(t,\tau)$ denotes the time-varying signal attenuation kernel along the path from the j-th transducer of the i-th array to the m-th transducer of the l-th array, $t_{i,j;l,m}(t)$ is the unique solution to the implicit equation

$$t - t_{i,j;l,m} - \frac{||\boldsymbol{x}_{l,m}(t) - \boldsymbol{x}_{i,j}(t_{i,j;l,m})||}{c} = 0 \tag{2.28}$$

and the $\tilde{v}_{l,m}(t)$ are independent Gaussian noise processes with flat power spectral density in the band of interest. When there is multi-path, we interpret each path as the line of sight path from a phantom source array at position $\boldsymbol{x}_{i;p}(t)$ and orientation $\boldsymbol{\theta}_{i;p}(t)$, $p \in [P_{i;l}]$, that sends out the same signals. The integer $P_{i;l}$ counts the number of paths present between array i and l. Figure 2.6 shows the real source and three phantom sources, one for each reflection. In the multi-path case, the received signals read

$$\tilde{r}_{l,m}(t) = \sum_{j\in[K],p\in P_{i;l}} \int_\tau h_{i,j;p;l,m}(t,\tau)\tilde{s}_{i,j}(t_{i,j;p;l,m}(t) - \tau)d\tau + \tilde{v}_{l,m}(t) \tag{2.29}$$

where $t_{i,j;p;l,m}(t)$ is the unique solution to the implicit equation

$$t - t_{i,j;p;l,m} - \frac{||\boldsymbol{x}_{l,m}(t) - \boldsymbol{x}_{i,j;p}(t_{i,j;p;l,m})||}{c} = 0 \tag{2.30}$$

$\boldsymbol{x}_{i,j;p}(t)$, $j \in [K]$, are the positions of the transducer elements on the p-th phantom array and $h_{i,j;p;l,m}(t, \tau)$ denotes the time-varying signal attenuation kernel along the path from the j-th transducer of the p-th phantom of the i-th array to the m-th transducer of the l-th array.

2.6 Signal Design and Sampling

We wish to design waveforms that are suitable for bandwidth efficient data communication and channel estimation. Standard single carrier source signals are well-suited for this task. It is possible to detect and track motion from the phase margin or lag with respect to the carrier (center frequency). Furthermore, modulation of the phase can be used to embed data.

A common approach to construction of such a communication signal is through varying the amplitude and phase of a collection of basis functions with limited bandwidth. Suppose the j-th transducer of the i-th array is to transmit length $N + 1$ sequences of symbols $s_{i,j}[n], n \in [0 : N]$, from a finite set of signal constellation points $\mathrm{A} \subset \mathbb{C}$. To this end, the sequence $s_{i,j}[n]$ is mapped to a waveform $s_{i,j}(t) : \mathbb{R} \to \mathbb{C}$

$$s_{i,j}(t) = \sum_{l \in [0:N]} s_{i,j}[l] p(t - lT) \tag{2.31}$$

by use of a basic pulse $p(t)$ time shifted by multiples of the symbol period T. The pulse $p(t)$ is typically assumed to have a bandwidth of no more than $1/T$. If some of the these symbols are unknown, they can usually be assumed to be i.i.d., either because the underlying symbols have been optimally compressed or randomly interleaved. This signal is then modulated to passband

$$\tilde{s}_{i,j}(t) = 2\,\mathrm{Re}\{s_{i,j}(t) e^{2\pi\sqrt{-1} f_{C\,i} t}\} \tag{2.32}$$

at carrier frequency $f_{C\,i}$. These frequencies are chosen such that there is no multiple access interference (MAI), i.e., $|f_{C\,i} - f_{C\,i'}| > 1/T$ for all $i \neq i'$.

At the receiving array, the signal $\tilde{r}_{l,m}(t)$ from Equation 2.29 is demodulated by f_{Ci} and low-pass filtered, which yields

$$r_{l,m}(t) = \sum_{j,p} \int_\tau h_{i,j;p;l,m}(t,\tau) e^{2\pi\sqrt{-1} f_{Ci}(t_{i,j;p;l,m}(t)-\tau-t)} s_{i,j}(t_{i,j;p;l,m}(t)-\tau) d\tau \\ + v_{l,m}(t) \tag{2.33}$$

where $v_{l,m}(t)$ denotes the demodulated and filtered noise processes. Motion-induced Doppler shifts might widen the bandwidth of the received signal. If the low-pass filter had only a bandwidth of $1/T$, a significant fraction of the signal could be lost. We assume that v_{max} is the maximal experienced speed. The maximum frequency of the emitted signal is designed to be $f_{Ci} + 1/2T$ and a sinusoid with that frequency would then experience a Doppler shift of at most $f_{di} = (f_{Ci} + 1/2T)\frac{v_{max}}{c}$. We hence increase the cut-off frequency of the low-pass filter by f_{di} and sample the filtered signal at the increased frequency $1/T_i = 1/T + 2f_{di}$. The sampled output equations read

$$r_{l,m}[n] = \sum_{j,p,k} h_{i,j;p;l,m}[n,k] e^{2\pi\sqrt{-1} f_{Ci}(t_{i,j;p;l,m}[n]-nT_i)} s_{i,j}(t_{i,j;p;l,m}[n]-kT_i) \\ + v_{l,m}[n] \tag{2.34}$$

where $t_{i,j;p;l,m}[n] = t_{i,j;p;l,m}(nT_i)$, $v_{l,m}[n]$ is the sampled noise process and

$$h_{i,j;p;l,m}[n,k] = T_i h_{i,j;p;l,m}(nT_i, kT_i) e^{-2\pi\sqrt{-1} f_{Ci} kT_i} \tag{2.35}$$

is the demodulated and sampled kernel. The original noise process $v_{l,m}(t)$ was Gaussian and white in the band of interest and hence the noise samples $v_{l,m}[n]$ are i.i.d. Gaussian.

Our objective is to communicate data sequences to the receiver. That is, parts of the sequences $s_{i,j}[l]$ are unknown and we would like to estimate them from the available observations $r_{l,m}[n]$. Unfortunately, the kernels $h_{i,j;p;l,m}[n,k]$ as well as the position and orientation vectors of the transmit and receive arrays are unknown as well. A possible approach to this problem is to model all these states probabilistically and then perform Bayesian estimation and estimate all these states jointly. We will propose suitable probabilistic models next.

2.7 Probabilistic Modeling of Attenuation

The channel gains $h_{i,j;p;l,m}[n]$ are random and we assume their evolution is described by the following state equations

$$h_{i,j;p;l,m}[n+1,k] = \lambda h_{i,j;p;l,m}[n,k] + u_{i,j;p;l,m}[n,k] \tag{2.36}$$

where, for each choice of the indices i, j, p, l, m and k, the random variables $u_{i,j;p;l,m}[n,k]$ form an independent white Gaussian noise process in n with variance σ_u^2. The parameter $\lambda \in (0,1)$ is the forgetting factor. More sophisticated a priori models for the evolution of these gains could be used but we will start off with this simple model. We, hence, neglect the clear dependence of the length and the attenuation of the involved signal propagation paths.

2.8 Probabilistic Modeling of Receiver Motion

Various motion models have been considered in the position tracking literature [61]. There are discrete time and continuous time models. The channel observations $r_{l,m}[n]$ depend on transmitter and receiver motion only through the emission time $t_{i,j;p;l,m}[n]$ which, by definition, is the solution to the implicit Equation 2.30 for $t = nT_i$. Clearly, the emission time is only influenced by the values of the functions $\boldsymbol{x}_l(t)$ and $\boldsymbol{\theta}_l(t)$ where $t = nT_i$, $n = 0, 1, 2, \ldots$, and we hence model the evolution of the receiver position and orientation in discrete time. Among the commonly used discrete time motion models, the discrete d-th order white noise model is among the simplest. In this model, each coordinate $x_{l;k}(t)$ of the vector $\boldsymbol{x}_l(t)$ is uncoupled and for each coordinate, k, the d-th derivative $x_{l;k}^{(d)}(t)$ is right-continuous and constant between sampling instants and $x_{l;k}^{(d)}[n] = x_{l;k}^{(d)}(nT_i), n = 0, 1, 2, \ldots$, is a white Gaussian noise process with variance σ_a^2. Iterated integration of $x_{l;k}^{(d)}(t)$ and sampling with period T_i yields the following linear discrete time state equations with

Toeplitz transition matrix:

$$\begin{pmatrix} x_{l;k}[n+1] \\ x_{l;k}^{(1)}[n+1] \\ \vdots \\ x_{l;k}^{(d-1)}[n+1] \end{pmatrix} = \begin{pmatrix} 1 & T_i & \dots & \frac{T_i^{d-1}}{(d-1)} \\ \mathbf{0} & \ddots & \ddots & \end{pmatrix} \begin{pmatrix} x_{l;k}[n] \\ x_{l;k}^{(1)}[n] \\ \vdots \\ x_{l;k}^{(d-1)}[n] \end{pmatrix} + \begin{pmatrix} \frac{T_i^d}{d} \\ \vdots \\ T_i \end{pmatrix} a_n \tag{2.37}$$

where a_n is an independent white Gaussian noise process with variance σ_a^2 and $d > 1$. Other more sophisticated motion models for example allow correlation across coordinates and take into account on-line information about the maneuver, but we postpone a more detailed modeling. Further, the orientation and position of an array are often correlated. Vehicles typically move in the direction of the orientation vector. For simplicity, we postpone the modeling of this effect as well and assume orientation and position to evolve independently but to share the same probabilistic model.

2.9 Probabilistic Modeling of Transmitter Motion

Again, the channel observations $r_{l,m}[n]$ depend on transmitter motion only through the emission time $t_{i,j;p;l,m}[n]$. If both the position $\boldsymbol{x}_{i;p}(t)$ and the orientation $\boldsymbol{\theta}_{i;p}(t)$ of the (phantom) transmit array are modeled by random processes with continuous sample paths and their speed is bounded by a sufficiently small value, then the positions $\boldsymbol{x}_{i,j;p}(t)$, $j \in [K]$, $p \in [P_{i;l}]$, of its array elements also have continuous sample paths and their speed is less than the speed of sound. In that case, by Theorem 1, there is a unique solution $t_{i,j;p;l,m}[n]$ to the implicit Equation 2.30 for $t = nT_i$ and each array element $j \in [K]$ and path $p \in [P_{i;l}]$. Note that the emission times $t_{i,j;p;l,m}[n], j \in [K], p \in [P_{i;l}]$ can be viewed as hitting times

$$t_{i,j;p;l,m}[n] = \inf \left\{ t_e : \frac{||\boldsymbol{x}_{l,m}(nT_i) - \boldsymbol{x}_{i,j;p}(t_e)||}{c} + t_e = nT_i \right\} \tag{2.38}$$

We propose to model each coordinate of the transmitter position $\boldsymbol{x}_{i;p}(t)$ and orientation $\boldsymbol{\theta}_{i;p}(t)$ as independent strong Markov processes [62]. More specifically, we propose to model the evolution of each coordinate by a bidimensional random process; the first dimension is a speed process, modeled as a Brownian motion reflected off a symmetric two-sided boundary, and the second dimension is the position process, which is the integral of the first dimension. For this setup, we conjecture that the vector $(\boldsymbol{x}_{i;p}(t), \dot{\boldsymbol{x}}_{i;p}(t), \boldsymbol{\theta}_{i;p}(t), \dot{\boldsymbol{\theta}}_{i;p}(t))$ describes a Feller process [62] and that the set of states

$$\{(\boldsymbol{x}_{i;p}(t), \dot{\boldsymbol{x}}_{i;p}(t), \boldsymbol{\theta}_{i;p}(t), \dot{\boldsymbol{\theta}}_{i;p}(t)),\ t = t_{i,j;p;l,m}[n],\ j \in [K]\} \tag{2.39}$$

indexed by the discrete time variable n, form a Markov chain of order R for each $p \in [P_{i;l}]$ given the receiver motion. The order R depends on the array geometry and the maximal speed of the above mentioned Brownian motion speed processes. We prove this conjecture for some special cases and discuss our thoughts on how these proofs could be extended to cover the general case. The first special case we will look at is that of one-dimensional motion on a line with one element transmit and receive arrays.

Let the random processes $x_i(t)$ and $x_l(t)$ denote the position of the transmitter and receiver on the real line at time t, respectively. Unfortunately, the simple model presented in Section 2.8 and Equation 2.37 is insufficient when transmitter motion is allowed. It would allow the transmitter and receiver to get arbitrarily high velocities with non-zero probability, leading to supersonic speed and non-unique emission times. Transmitter speed needs to be bounded in order for Theorem 1 to guarantee unique emission times. Further, receiver speed needs to be bounded in order for the emission times $t_e[n]$ to form a strictly increasing sequence. This is a necessary condition for the bidimensional process $(x_i(t_e[n]), \dot{x}_i(t_e[n]))$ to be Markov in n. The following definitions and theorems will make these points more precise and give an approximation of the transition kernel of the Markov chain $(x_i(t_e[n]), \dot{x}_i(t_e[n]))$.

We will drive our motion model by a Brownian motion.

Definition 1. *A stochastic process $B(t)$, $t \in \mathbb{R}_{\geq 0}$, is called a Brownian motion if*

1. $B(0) = 0$
2. *$B(t)$ is continuous almost surely*
3. *$B(t)$ has independent increments*

4. $B(t) - B(s) \sim N(0, t-s)$ for $0 \leq s < t$, where $N(0, t-s)$ is the normal distribution with zero mean and variance $t - s$.

The following motion model uses Brownian motion as the speed process, gives continuous sample paths, is strongly Markov, has Gaussian distributed independent increments and its hitting time distribution is well studied [63–66].

Definition 2. *(Integrated Brownian motion (IBM) model) For any non-negative time $t \in \mathbb{R}_{\geq 0}$, the position of the transmitter is given by*

$$x(t) = x_0 + \int_0^t \dot{x}(\tau) d\tau \tag{2.40}$$

where the speed process $\dot{x}(t)$ is given by

$$\dot{x}(t) = \dot{x}_0 + \alpha B(t) \tag{2.41}$$

the values $x_0, \dot{x}_0 \in \mathbb{R}$ and $\alpha \in \mathbb{R}_{>0}$ are model parameters and $B(t)$ is a Brownian motion as in Definition 1. For any negative time t, $x(t) = x_0 + \dot{x}_0 t$ and $\dot{x}(t) = \dot{x}_0$. The bidimensional process $\boldsymbol{\xi}(t) = (x(t), \dot{x}(t))$ defines the integrated Brownian motion (IBM) model.

The problem with this motion model is that speed is unbounded and hence emission times can become non-unique. We propose a motion model that is similar to the one above but gives position sample paths whose speed is bounded by some value smaller than the speed of sound so that there is a unique emission time. We simply reflect the above speed process off a symmetric two sided boundary to ensure it is bounded almost surely.

Definition 3. *(Integrated reflected Brownian motion (IRBM) model) For any non-negative time $t \in \mathbb{R}_{\geq 0}$, the position of the transmitter is given by*

$$x(t) = x_0 + \int_0^t \dot{x}(\tau) d\tau \tag{2.42}$$

where the speed process $\dot{x}(t)$ is given by

$$\dot{x}(t) = g(\dot{x}_0 + \alpha B(t)) \tag{2.43}$$

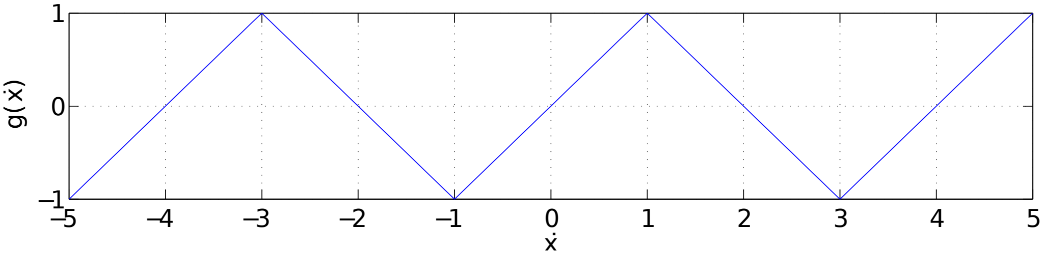

Figure 2.10: The operation of the function $g(\dot{x})$ from Definition 3 for $\dot{x}_{max} = 1$.

the values $x_0, \dot{x}_0 \in \mathbb{R}$ and $\alpha \in \mathbb{R}_{>0}$ are model parameters and $B(t)$ is a Brownian motion as in Definition 1. We have that

$$g(\dot{x}) = (-1)^{n(\dot{x})}(\dot{x} - 2\dot{x}_{max} n(\dot{x})) \tag{2.44}$$

where $n(\dot{x}) = \left\lfloor \frac{\dot{x}}{2\dot{x}_{max}} \right\rceil$, the operator $\lfloor \cdot \rceil$ denotes rounding to the nearest integer and $\dot{x}_{max} \in \mathbb{R}_{>0}$ bounds $|\dot{x}(t)|$. Both $|\dot{x}_0|$ and $\dot{x}_{max}$ are always chosen to be smaller than the speed of sound c. For any negative time t, $x(t) = x_0 + \dot{x}_0\, t$ and $\dot{x}(t) = \dot{x}_0$. The bidimensional process $\boldsymbol{\xi}(t) = (x(t), \dot{x}(t))$ defines the integrated reflected Brownian motion (IRBM) model.

For any function $\dot{x}(t)$, the function $g(\dot{x}(t))$ reflects values greater than $\dot{x}_{max}$ inwards. The operation of the function $g(\dot{x})$ is illustrated in Figure 2.10 for $\dot{x}_{max} = 1$. The function $g(\dot{x})$ has an interesting property that we will exploit in Theorem 2 below.

Lemma 1. *If $g(\dot{x})$ and $n(\dot{x})$ are the functions defined in Equation 2.44 in Definition 3 for some $\dot{x}_{max} > 0$, then*

$$g((-1)^m \dot{x} + 2\dot{x}_{max} m) = g(\dot{x}) \tag{2.45}$$

for any integer m.

Proof. We have

$$\begin{align} n((-1)^m \dot{x} + 2\dot{x}_{max} m) &= \left\lfloor \frac{(-1)^m \dot{x} + 2\dot{x}_{max} m}{2\dot{x}_{max}} \right\rceil \tag{2.46} \\ &= m + (-1)^m \left\lfloor \frac{\dot{x}}{2\dot{x}_{max}} \right\rceil \tag{2.47} \end{align}$$

and hence

$$
\begin{aligned}
& g((-1)^m\dot{x}+2\dot{x}_{max}m) && (2.48)\\
&= (-1)^{m+(-1)^m\left\lfloor\frac{\dot{x}}{2\dot{x}_{max}}\right\rceil}\left((-1)^m\dot{x}+2\dot{x}_{max}m-2\dot{x}_{max}\left(m+(-1)^m\left\lfloor\frac{\dot{x}}{2\dot{x}_{max}}\right\rceil\right)\right) && (2.49)\\
&= (-1)^{m+(-1)^m\left\lfloor\frac{\dot{x}}{2\dot{x}_{max}}\right\rceil}\left((-1)^m\dot{x}-2\dot{x}_{max}(-1)^m\left\lfloor\frac{\dot{x}}{2\dot{x}_{max}}\right\rceil\right) && (2.50)\\
&= (-1)^{(-1)^m\left\lfloor\frac{\dot{x}}{2\dot{x}_{max}}\right\rceil}\left(\dot{x}-2\dot{x}_{max}\left\lfloor\frac{\dot{x}}{2\dot{x}_{max}}\right\rceil\right) && (2.51)\\
&= (-1)^{\left\lfloor\frac{\dot{x}}{2\dot{x}_{max}}\right\rceil}\left(\dot{x}-2\dot{x}_{max}\left\lfloor\frac{\dot{x}}{2\dot{x}_{max}}\right\rceil\right) && (2.52)\\
&= g(\dot{x}) && (2.53)
\end{aligned}
$$

□

Remark 1. *For the applications of interest in this thesis, the maximum platform speed and acceleration of the underwater vehicle is about $2m/s$ and $0.3m/s^2$, respectively [67]. The parameter α in the above motion models determines the level of acceleration and is chosen such that the standard deviation of $\alpha B(T_i)$ is a third of $0.3T_i$, i.e. $\alpha = 0.1\sqrt{T_i}$. Further we choose $\dot{x}_{max} = 5m/s$.*

The integrated reflected Brownian motion (IRBM) model $\boldsymbol{\xi}(t)$ defined in Definition 3 is no longer an independent increment process, but its sample paths are continuous and we can prove that it is a Feller process. We prove this property so that we may exploit the strong Markov property that follows from it [62].

Definition 4. *(Markov Process) Let $(\Omega, \mathcal{F}, P)$ be a probability space and let $(S, \mathcal{S})$ be a measurable space. The S-valued stochastic process $\boldsymbol{\xi} = (\boldsymbol{\xi}(t),\ t \in \mathbb{R}_{\geq 0})$ with natural filtration $(\mathcal{F}_t,\ t \in \mathbb{R}_{\geq 0})$ is said to be a strong Markov process, if for each $A \in \mathcal{S}$, $s > 0$ and any stopping time τ,*

$$P(\boldsymbol{\xi}(\tau+s) \in A|\mathcal{F}_\tau) = P(\boldsymbol{\xi}(\tau+s) \in A|\boldsymbol{\xi}(\tau)) \tag{2.54}$$

where

$$\mathcal{F}_\tau = \{A \in \mathcal{F} :\ A \cap \{\tau \leq t\} \in \mathcal{F}_t \textit{ for all } t \geq 0\} \tag{2.55}$$

is the sigma algebra at the stopping time τ. If Equation 2.54 only holds for the trivial stopping times $\tau = t$ for any $t \geq 0$, then the process is just called a Markov process. The Markov transition kernel $\mu_{t,t+s}(\boldsymbol{\xi}_0, A) : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times S \times \mathcal{S}) \to [0, 1]$ is a probability measure given any initial state $\boldsymbol{\xi}_0 \in S$ and any $t, s > 0$ and further

$$\mu_{t,t+s}(\boldsymbol{\xi}(t), A) = P(\boldsymbol{\xi}(t+s) \in A|\boldsymbol{\xi}(t)) \tag{2.56}$$

almost surely for any $A \in \mathcal{S}$ and any $t, s > 0$. A Markov process is homogeneous if for any initial state $\boldsymbol{\xi}_0 \in S$, any $A \in \mathcal{S}$ and any $t, s > 0$

$$\mu_{t,t+s}(\boldsymbol{\xi}_0, A) = \mu_{0,s}(\boldsymbol{\xi}_0, A) \tag{2.57}$$

For homogeneous Markov processes, we use the notation

$$P_{\boldsymbol{\xi}_0}(\boldsymbol{\xi}(s) \in A) \equiv \mu_{0,s}(\boldsymbol{\xi}_0, A) \tag{2.58}$$

When the expected value of some random variable G is computed with respect to this probability measure, we write $\mathbb{E}_{\boldsymbol{\xi}_0}[G]$.

Definition 5. *(Feller Process) Let $\boldsymbol{\xi} = (\boldsymbol{\xi}(t),\ t \in \mathbb{R}_{\geq 0})$ be a homogeneous Markov process as defined in Definition 4. Then this process is called a Feller process, when, for all initial states $\boldsymbol{\xi}_0$,*

1. *for any $t \geq 0$, any event $A \in \mathcal{S}$ and any sequence of states $\boldsymbol{\xi}_n \in S$, $\lim_{n\to\infty} \boldsymbol{\xi}_n = \boldsymbol{\xi}_0$ implies $\lim_{n\to\infty} P_{\boldsymbol{\xi}_n}(\boldsymbol{\xi}(t) \in A) = P_{\boldsymbol{\xi}_0}(\boldsymbol{\xi}(t) \in A)$*
2. *for any $\epsilon > 0$, $\lim_{t\to 0} P(\|\boldsymbol{\xi}(t) - \boldsymbol{\xi}_0\| > \epsilon|\boldsymbol{\xi}(0) = \boldsymbol{\xi}_0) = 0$*

Theorem 2. *The bidimensional random process $\boldsymbol{\xi}(t)$ from Definition 3 is a Feller process.*

Proof. The sample paths of the process $\boldsymbol{\xi}(t)$ are continuous and hence Property 2 in Definition 5 holds. We will now prove that $\boldsymbol{\xi}(t)$ is a homogeneous Markov process and that Property 1 in Definition 5 holds as well. Let $\mathcal{F}_t^x$ and $\mathcal{F}_t^{\dot{x}}$ be the natural filtrations of the processes $x(t)$ and $\dot{x}(t)$, respectively.

We immediately establish from the definition of the function $g(\dot{x})$ in Equation 2.44 that

$$\alpha B(t) + \dot{x}_0 = g(\alpha B(t) + \dot{x}_0)(-1)^n + 2\dot{x}_{max} n \tag{2.59}$$

where we abbreviated the notation $n(\alpha B(t) + \dot{x}_0)$ by n.

Further, we note that

$$\begin{aligned}
\dot{x}(t+\tau) &= g(\alpha B(t+\tau) + \dot{x}_0) && (2.60)\\
&= g(\alpha(B(t+\tau) - B(t)) + \alpha B(t) + \dot{x}_0) && (2.61)\\
&= g(\alpha(B(t+\tau) - B(t)) + g(\alpha B(t) + \dot{x}_0)(-1)^n + 2\dot{x}_{max}n) && (2.62)\\
&= g((-1)^n(\alpha B'(\tau) + g(\alpha B(t) + \dot{x}_0)) + 2\dot{x}_{max}n) && (2.63)\\
&= g(\alpha B'(\tau) + g(\alpha B(t) + \dot{x}_0)) && (2.64)\\
&= g(\alpha B'(\tau) + \dot{x}(t)) && (2.65)
\end{aligned}$$

Equation 2.62 follows from Equation 2.59. Equation 2.64 follows from Lemma 1. The weighted difference $B'(\tau) = (-1)^n(B(t+\tau) - B(t))$ is itself a Brownian motion and independent of $\mathcal{F}_t^x$ and $\mathcal{F}_t^{\dot{x}}$.

Next, we take a look at the conditional moment-generating function [62] of the bidimensional process $\boldsymbol{\xi}(t)$.

$$\begin{aligned}
&\mathbb{E}_{x_0,\dot{x}_0}[e^{ux(t+\tau)+v\dot{x}(t+\tau)}|\mathcal{F}_t^x, \mathcal{F}_t^{\dot{x}}] && (2.66)\\
&= \mathbb{E}_{x_0,\dot{x}_0}[e^{u(x(t)+\int_0^\tau \dot{x}(t+\tau)d\tau)+v\dot{x}(t+\tau)}|\mathcal{F}_t^x, \mathcal{F}_t^{\dot{x}}] && (2.67)\\
&= \mathbb{E}_{x_0,\dot{x}_0}[e^{u(x(t)+\int_0^\tau g(\alpha B'(\tau)+\dot{x}(t))d\tau)+vg(\alpha B'(\tau)+\dot{x}(t))}|\mathcal{F}_t^x, \mathcal{F}_t^{\dot{x}}] && (2.68)\\
&= \mathbb{E}_{x_0,\dot{x}_0}[e^{u(x(t)+\int_0^\tau g(\alpha B'(\tau)+\dot{x}(t))d\tau)+vg(\alpha B'(\tau)+\dot{x}(t))}|x(t), \dot{x}(t)] && (2.69)\\
&= \mathbb{E}_{x(t),\dot{x}(t)}[e^{u(x(0)+\int_0^\tau g(\alpha B(\tau)+\dot{x}(0))d\tau)+vg(\alpha B(\tau)+\dot{x}(0))}] && (2.70)\\
&= \mathbb{E}_{x(t),\dot{x}(t)}[e^{ux(\tau)+v\dot{x}(\tau)}] && (2.71)
\end{aligned}$$

Equation 2.68 follows from Equation 2.65. Equation 2.69 follows from the Markov property of Brownian motion. Equation 2.71 follows from the fundamental theorem of calculus and Equation 2.43. So $\boldsymbol{\xi}(t)$ is a homogeneous Markov process. Now assume there are two sequences $x_n : \mathbb{Z}_+ \to \mathbb{R}$ and $\dot{x}_m : \mathbb{Z}_+ \to \mathbb{R}$ such that $\lim_{n\to\infty} x_n = x_0$ and $\lim_{m\to\infty} \dot{x}_m = \dot{x}_0$. Then

$$\begin{aligned}
&\lim_{n,m\to\infty} \mathbb{E}_{x_n,\dot{x}_m}[e^{ux(\tau)+v\dot{x}(\tau)}] && (2.72)\\
&= \lim_{n,m\to\infty} \mathbb{E}[e^{u(x_n+\int_0^\tau g(\alpha B(\tau)+\dot{x}_m)d\tau)+vg(\alpha B(\tau)+\dot{x}_m)}] && (2.73)\\
&= \mathbb{E}[e^{u(\lim_{n\to\infty} x_n+\int_0^\tau g(\alpha B(\tau)+\lim_{m\to\infty} \dot{x}_m)d\tau)+vg(\alpha B(\tau)+\lim_{m\to\infty} \dot{x}_m)}] && (2.74)\\
&= \mathbb{E}[e^{u(x_0+\int_0^\tau g(\alpha B(\tau)+\dot{x}_0)d\tau)+vg(\alpha B(\tau)+\dot{x}_0)}] && (2.75)
\end{aligned}$$

Equation 2.74 follows from the dominated convergence theorem [62]. Convergence of the moment-generating function implies convergence of the corresponding distribution and hence Property 1 in Definition 5 holds as well. □

Now assuming that the motion model for the transmitter and receiver is as defined in Definition 3, transmitter speed is bounded and there is a unique solution t_e to the implicit equation

$$t - t_e - \frac{|x_l(t) - x_i(t_e)|}{c} = 0 \tag{2.76}$$

for any t by Theorem 1. We can show that the sequence $t_e[n]$, the solutions of the implicit Equation 2.76 for $t = nT_i$, is strictly increasing in n.

Theorem 3. *Assume both transmitter and receiver motion, $\boldsymbol{\xi}_i(t)$ and $\boldsymbol{\xi}_l(t)$, are as defined in Definition 3. If $t_e[n]$ denotes the solution of the implicit Equation 2.76 for $t = nT_i$, then*

$$t_e[n+1] > t_e[n], \, \forall n \tag{2.77}$$

Further,

$$T_i \left(\frac{1 + \frac{\dot{x}_{max}}{c}}{1 - \frac{\dot{x}_{max}}{c}} \right) \geq |t_e[n+1] - t_e[n]| \geq T_i \left(\frac{1 - \frac{\dot{x}_{max}}{c}}{1 + \frac{\dot{x}_{max}}{c}} \right) \tag{2.78}$$

Proof. Evaluating the implicit Equation 2.76 for $t = nT_i$ and $t = (n+1)T_i$ gives

$$nT_i - t_e[n] - \frac{|x_l(nT_i) - x_i(t_e[n])|}{c} = 0 \tag{2.79}$$

and

$$(n+1)T_i - t_e[n+1] - \frac{|x_l((n+1)T_i) - x_i(t_e[n+1])|}{c} = 0 \tag{2.80}$$

The theorem essentially follows from iterated application of the triangle inequality. By a suitable zero-sum expansion,

$$\begin{aligned}
&|x_l((n+1)T_i) - x_i(t_e[n+1])|\\
=&|x_l((n+1)T_i) - x_l(nT_i) + x_l(nT_i) - x_i(t_e[n]) + x_i(t_e[n]) - x_i(t_e[n+1])|\\
\leq&|x_l((n+1)T_i) - x_l(nT_i)| + |x_l(nT_i) - x_i(t_e[n])| + |x_i(t_e[n]) - x_i(t_e[n+1])|
\end{aligned} \tag{2.81}$$

Subtracting Equation 2.80 from Equation 2.79, yields

$$- T_i + (t_e[n+1] - t_e[n]) \tag{2.82}$$

$$= \frac{1}{c}\left(|x_l(nT_i) - x_i(t_e[n])| - |x_l((n+1)T_i) - x_i(t_e[n+1])|\right) \tag{2.83}$$

$$\geq -\frac{1}{c}\left(|x_i(t_e[n+1]) - x_i(t_e[n])| + |x_l((n+1)T_i) - x_l(nT_i)|\right) \tag{2.84}$$

$$\geq -\frac{1}{c}\left(\dot{x}_{max}|t_e[n+1] - t_e[n]| + \dot{x}_{max}T_i\right) \tag{2.85}$$

The first inequality follows from Inequality 2.81. The second inequality follows from the fact that the involved motion processes have bounded speed. We can hence write

$$t_e[n+1] - t_e[n] \geq T_i - \frac{\dot{x}_{max}}{c}\left(|t_e[n+1] - t_e[n]| + T_i\right) \tag{2.86}$$

and conclude

$$\begin{aligned}
&(t_e[n+1] - t_e[n])\underbrace{\left(1 + \operatorname{sgn}(t_e[n+1] - t_e[n])\frac{\dot{x}_{max}}{c}\right)}_{>0}\\
&\geq T_i\left(1 - \frac{\dot{x}_{max}}{c}\right) > 0
\end{aligned} \tag{2.87}$$

and

$$|t_e[n+1] - t_e[n]| \geq T_i\left(\frac{1 - \frac{\dot{x}_{max}}{c}}{1 + \frac{\dot{x}_{max}}{c}}\right) \tag{2.88}$$

This proves Inequality 2.77 and the right-hand side inequality in Equation 2.78. If instead of expanding the argument of the right-hand side norm of Equation 2.83, the argument of the left-hand side norm of Equation 2.83 is expanded, we get the inequality

$$t_e[n+1] - t_e[n] \leq T_i + \frac{\dot{x}_{max}}{c}\left(|t_e[n+1] - t_e[n]| + T_i\right) \tag{2.89}$$

and we conclude

$$|t_e[n+1] - t_e[n]| \leq T_i \left(\frac{1 + \frac{\dot{x}_{max}}{c}}{1 - \frac{\dot{x}_{max}}{c}}\right) \tag{2.90}$$

□

The fact that the emission times $t_e[n]$ are strictly increasing allows us to prove that $\boldsymbol{\xi}_i(t_e[n])$ is Markov.

Theorem 4. *Assume both transmitter and receiver motion, $\boldsymbol{\xi}_i(t)$ and $\boldsymbol{\xi}_l(t)$, are as defined in Definition 3, but that the receiver motion $\boldsymbol{\xi}_l(t)$ is given at the times nT_i. Also, let the time $t_e[n]$ denote the solution of the implicit Equation 2.76 for $t = nT_i$. Then the sequence $\boldsymbol{\xi}_i(t_e[n])$ is Markov, i.e., for any $A \in \mathcal{B}(\mathbb{R}^2)$,*

$$\begin{aligned}&P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n+1]) \in A | \boldsymbol{\xi}_i(t_e[k-1]), k \leq n)\\ &= P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n+1]) \in A | \boldsymbol{\xi}_i(t_e[n]))\end{aligned} \tag{2.91}$$

Further,

$$\begin{aligned}&P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n+1]) \in A | \boldsymbol{\xi}_i(t_e[n]))\\ &= P_{\boldsymbol{\xi}_i(t_e[n])}(\boldsymbol{\xi}_i(\delta t_e) \in A)\end{aligned} \tag{2.92}$$

where

$$\begin{aligned}\delta t_e = \inf\Bigg\{\delta t_e : T_i - \delta t_e = \frac{1}{c}\Bigg(|x_l((n+1)T_i) - x_i(0) - \int_0^{\delta t_e} \dot{x}_i(\tau)d\tau| \ldots\\ -|x_l(nT_i) - x_i(0)|\Big)\Big\}\end{aligned} \tag{2.93}$$

Proof. The sequence of σ-algebras $\mathcal{F}_t^{\boldsymbol{\xi}_i} = \sigma\{\boldsymbol{\xi}_i(\tau)^{-1}(\mathcal{B}(\mathbb{R}^2)), 0 \leq \tau \leq t\}$ is the natural filtration of the process $\boldsymbol{\xi}_i(t)$. Let s and τ be some non-negative real numbers. The emission times $t_e[n]$ are stopping times and $\mathcal{F}_{t_e[n]+s}^{\boldsymbol{\xi}_i}$ is the stopping time σ-algebra for the stopping time $t_e[n] + s$. Since $\boldsymbol{\xi}_i(t)$ is a time-homogeneous strong Markov process, by Theorem 2, we have for any $s, \tau \geq 0$ and $A \in \mathcal{B}(\mathbb{R}^2)$ that

$$\begin{aligned}
&P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n] + s + \tau) \in A | \mathcal{F}^{\boldsymbol{\xi}_i}_{t_e[n]+s}) \\
&\quad = P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n] + s + \tau) \in A | \boldsymbol{\xi}_i(t_e[n] + s)) && (2.94) \\
&\quad = P_{\boldsymbol{\xi}_i(t_e[n]+s)}(\boldsymbol{\xi}_i(\tau) \in A) && (2.95)
\end{aligned}$$

For any two σ-algebras $\mathcal{Y}$ and $\mathcal{Z}$ of subsets of Ω, $\sigma\{\mathcal{Y}, \mathcal{Z}\}$ denotes the smallest σ-algebra that contains both $\mathcal{Y}$ and $\mathcal{Z}$. We define

$$\begin{aligned}
\breve{\mathcal{F}}^{\boldsymbol{\xi}_i}_{t_e[n]+s} = \sigma\{&\big(\boldsymbol{\xi}_i(t_e[n] + \gamma)^{-1}(\mathcal{B}(\mathbb{R}^2)), 0 \leq \gamma \leq s\big)\,, \\
&\big(\boldsymbol{\xi}_i(t_e((k-1)T_i))^{-1}(\mathcal{B}(\mathbb{R}^2)), k \leq n\big)\} && (2.96)
\end{aligned}$$

The stopping times $t_e[n]$ form a strictly increasing sequence in n by Theorem 3 and hence

$$\breve{\mathcal{F}}^{\boldsymbol{\xi}_i}_{t_e[n]+s} \subset \mathcal{F}^{\boldsymbol{\xi}_i}_{t_e[n]+s} \tag{2.97}$$

By the tower property of conditional expectation and Equations 2.94, 2.95 and 2.97, for any $A \in \mathcal{B}(\mathbb{R}^2)$,

$$\begin{aligned}
&P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n] + s + \tau) \in A | \breve{\mathcal{F}}^{\boldsymbol{\xi}_i}_{t_e[n]+s}) \\
&\quad = P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n] + s + \tau) \in A | \boldsymbol{\xi}_i(t_e[n] + s)) && (2.98) \\
&\quad = P_{\boldsymbol{\xi}_i(t_e[n]+s)}(\boldsymbol{\xi}_i(\tau) \in A) && (2.99)
\end{aligned}$$

So the process $\boldsymbol{\xi}_i(t)$ renews itself after any stopping time $t_e[n]$.

Let $\delta t_e[n+1] = t_e[n+1] - t_e[n]$. By the definition of the emission times $t_e[n]$,

$$\begin{aligned}
\delta t_e[n+1] = \inf \{&\delta t_e : T_i - \delta t_e = \dots \\
&\frac{1}{c}\left(|x_l((n+1)T_i) - x_i(t_e[n] + \delta t_e)| - |x_l(nT_i) - x_i(t_e[n])|\right)\Big\}
\end{aligned} \tag{2.100}$$

or equivalently

$$
\begin{aligned}
\delta t_e[n+1] = \inf\{\delta t_e : T_i - \delta t_e = \dots \\
&\frac{1}{c}\left(|x_l((n+1)T_i) - x_i(t_e[n]) - \int_0^{\delta t_e} \dot{x}_i(t_e[n]+\tau)d\tau| \dots \right. \\
&\left. -|x_l(nT_i) - x_i(t_e[n])|\right)\} \qquad (2.101)
\end{aligned}
$$

Note that $\delta t_e[n+1]$ is independent of $\mathcal{F}^{\boldsymbol{\xi}_i}_{t_e[n]}$ given $\boldsymbol{\xi}_i(t_e[n])$ and hence

$$
\begin{aligned}
&P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n]+\delta t_e[n+1]) \in A | \boldsymbol{\xi}_i(t_e((k-1)T_i)), k \leq n) \\
&= P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(t_e[n]+\delta t_e[n+1]) \in A | \boldsymbol{\xi}_i(t_e[n])) \qquad (2.102) \\
&= P_{\boldsymbol{\xi}_i(t_e[n])}(\boldsymbol{\xi}_i(\delta t_e) \in A) \qquad (2.103)
\end{aligned}
$$

where δt_e is as defined in Equation 2.93. □

We do not have an exact solution to the kernel $P_{\boldsymbol{\xi}_i(t_e[n])}(\boldsymbol{\xi}_i(\delta t_e) \in A)$ from the previous theorem, but we can find a damn good approximation.

Theorem 5. *Assume both transmitter and receiver motion, $\boldsymbol{\xi}_i(t)$ and $\boldsymbol{\xi}_l(t)$, are as defined in Definition 3, but that the receiver motion $\boldsymbol{\xi}_l(t)$ is given at the times nT_i. Further assume that the motion $\boldsymbol{\xi}'_i(t)$ is as defined in Definition 2. The value $\boldsymbol{\xi}_i(0)$ is given, it is the initial condition for the motion processes $\boldsymbol{\xi}_i(t)$ and $\boldsymbol{\xi}'_i(t)$ and it is such that*

$$|x_l((n+1)T_i) - x_i(0)| > \dot{x}_{max}\delta t_{max} \qquad (2.104)$$

where

$$\delta t_{max} \equiv T_i\left(\frac{1+\frac{\dot{x}_{max}}{c}}{1-\frac{\dot{x}_{max}}{c}}\right) \qquad (2.105)$$

Then, for any $A \in \mathcal{B}(\mathbb{R}^2)$,

$$
\begin{aligned}
&|P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(\delta t_e[n+1]) \in A) - P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}'_i(\delta t'_e[n+1]) \in A)| \\
&\leq 2\,\mathrm{erfc}\,(\eta) - \mathrm{erfc}\,(3\eta) \qquad (2.106)
\end{aligned}
$$

where

$$\delta t_e[n+1] = \inf\left\{\delta t_e : T_i - \delta t_e = \frac{1}{c}\left(|x_l((n+1)T_i) - x_i(0) - \int_0^{\delta t_e} \dot{x}_i(\tau)d\tau| \ldots\right.\right.$$
$$\left.\left. -|x_l(nT_i) - x_i(0)|\right)\right\} \tag{2.107}$$

$$\delta t_e'[n+1] = \inf\left\{\delta t_e' : T_i - \delta t_e' = \frac{1}{c}\left(|x_l((n+1)T_i) - x_i'(0)| \ldots\right.\right.$$
$$\left.\left. - \operatorname{sgn}(x_l((n+1)T_i) - x_i'(0)) \int_0^{\delta t_e'} \dot{x}_i'(\tau)d\tau - |x_l(nT_i) - x_i'(0)|\right)\right\} \tag{2.108}$$

and

$$\eta = \frac{\dot{x}_{max} - |\dot{x}_i(0)|}{\alpha\sqrt{2\delta t_{max}}}. \tag{2.109}$$

Proof. For all $\delta t_e \leq \delta t_{max}$, Inequality 2.104 ensures

$$\left|\int_0^{\delta t_e} \dot{x}_i(\tau)d\tau\right| \leq \dot{x}_{max}\delta t_{max} < |x_l((n+1)T_i) - x_i(0)| \tag{2.110}$$

and hence

$$|x_l((n+1)T_i) - x_i(0) - \int_0^{\delta t_e} \dot{x}_i(\tau)d\tau|$$
$$= |x_l((n+1)T_i) - x_i(0)| - \operatorname{sgn}(x_l((n+1)T_i) - x_i(0)) \int_0^{\delta t_e} \dot{x}_i(\tau)d\tau \tag{2.111}$$

Note that by Theorem 3 the inequality $\delta t_e[n+1] \leq \delta t_{max}$ holds almost surely. We can thus write

$$\delta t_e[n+1] = \inf\left\{\delta t_e : T_i - \delta t_e = \frac{1}{c}\left(|x_l((n+1)T_i) - x_i(0)| \ldots\right.\right.$$
$$\left.\left. - \operatorname{sgn}(x_l((n+1)T_i) - x_i(0)) \int_0^{\delta t_e} \dot{x}_i(\tau)d\tau - |x_l(nT_i) - x_i(0)|\right)\right\} \tag{2.112}$$

By the law of total probability,

$$\begin{aligned} &P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(\delta t_e[n+1]) \in A) \\ &= P_{\boldsymbol{\xi}_i(0)}(\{\boldsymbol{\xi}_i(\delta t_e[n+1]) \in A\} \cap \{|\dot{x}_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}) + \dots \\ &\qquad + P_{\boldsymbol{\xi}_i(0)}(\{\boldsymbol{\xi}_i(\delta t_e[n+1]) \in A\} \cap \{|\dot{x}_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}^C) \end{aligned} \tag{2.113}$$

By Definition 2 and 3 and Equation 2.111,

$$\begin{aligned} &P_{\boldsymbol{\xi}_i(0)}(\{\boldsymbol{\xi}_i(\delta t_e[n+1]) \in A\} \cap \{|\dot{x}_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}) \\ &= P_{\boldsymbol{\xi}_i(0)}(\{\boldsymbol{\xi}'_i(\delta t'_e[n+1]) \in A\} \cap \{|\dot{x}'_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}) \end{aligned} \tag{2.114}$$

because given $\{|\dot{x}_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}$, the integrated Brownian motion model and the integrated reflected Brownian motion model coincide. Further by monotonicity

$$0 \leq P_{\boldsymbol{\xi}_i(0)}(\{\boldsymbol{\xi}_i(\delta t_e[n+1]) \in A\} \cap \{|\dot{x}_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}^C) \tag{2.115}$$

$$\leq P_{\boldsymbol{\xi}_i(0)}(\{|\dot{x}_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}^C) \tag{2.116}$$

And by the Fréchet inequalities [68, 69],

$$\max\big(0, P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}'_i(\delta t'_e[n+1]) \in A) + P_{\boldsymbol{\xi}_i(0)}(|\dot{x}'_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}) - 1\big)$$

$$\leq P_{\boldsymbol{\xi}_i(0)}(\{\boldsymbol{\xi}'_i(\delta t'_e[n+1]) \in A\} \cap \{|\dot{x}'_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}) \tag{2.117}$$

$$\leq P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}'_i(\delta t'_e[n+1]) \in A) \tag{2.118}$$

We conclude

$$|P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(\delta t_e[n+1]) \in A) - P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}'_i(\delta t'_e[n+1]) \in A)|$$

$$\leq P_{\boldsymbol{\xi}_i(0)}(\{|\dot{x}_i(\delta t)| < \dot{x}_{max}, 0 \leq \delta t \leq \delta t_{max}\}^C) \tag{2.119}$$

$$\leq P\left(\left\{|B(\delta t)| < \frac{\dot{x}_{max} - |\dot{x}_i(0)|}{\alpha}, 0 \leq \delta t \leq \delta t_{max}\right\}^C\right) \tag{2.120}$$

We are now going to give an expression and an upper bound for the last term. We define the square wave

$$s_{B_{max}}(b) = \sum_{n=-\infty}^{\infty} (-1)^n \mathbf{1}_{\{2n-1<b/B_{max}<2n+1\}}. \tag{2.121}$$

for $B_{max} = \frac{\dot{x}_{max}-|\dot{x}_i(0)|}{\alpha} > 0$. This function is antisymmetric around B_{max} and $-B_{max}$. Let τ be the first time the Brownian motion $B(t)$ hits either of those values. Then, by the reflection principle,

$$B'(t) = B(t) + \mathbf{1}_{\{t\geq\tau\}} 2(B(\tau) - B(t)) \tag{2.122}$$

is also a Brownian motion. We have

$$\mathbf{1}_{\{\tau>\delta t_{max}\}} = \frac{1}{2}\left(s_{B_{max}}(B(\delta t_{max})) + s_{B_{max}}(B'(\delta t_{max}))\right) \tag{2.123}$$

Applying the expectation operator on both sides gives

$$\begin{aligned} &P\left(|B(\delta t)| < B_{max}, 0 \leq \delta t \leq \delta t_{max}\right) \\ &= \mathbb{E}[(s_{B_{max}}(B(\delta t_{max}))] && (2.124) \\ &= \int_{-\infty}^{\infty} s_{B_{max}}(b) p_{\delta t_{max}}(b) db && (2.125) \end{aligned}$$

where $p_{\delta t_{max}}(b)$ is the density of the $N(0, \delta t_{max})$ Gaussian distribution. This integral can easily be bounded by truncating the sum $s_{B_{max}}(b)$, because $|s_{B_{max}}(b)| = 1$ and the density $p_{\delta t_{max}}(b)$ is decreasing in $|b|$:

$$\int_{-\infty}^{\infty} s_{B_{max}}(b) p_{\delta t_{max}}(b) db \geq \int_{-B_{max}}^{B_{max}} p_{\delta t_{max}}(b) db - 2\int_{B_{max}}^{3B_{max}} p_{\delta t_{max}}(b) db \tag{2.126}$$

$$= 2\,\mathrm{erf}\left(\frac{B_{max}}{\sqrt{2\delta t_{max}}}\right) - \mathrm{erf}\left(\frac{3B_{max}}{\sqrt{2\delta t_{max}}}\right) \tag{2.127}$$

And hence

$$\begin{aligned} &|P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}_i(\delta t_e[n+1]) \in A) - P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}'_i(\delta t'_e[n+1]) \in A)| \\ &\leq 1 - 2\,\mathrm{erf}\left(\frac{B_{max}}{\sqrt{2\delta t_{max}}}\right) + \mathrm{erf}\left(\frac{3B_{max}}{\sqrt{2\delta t_{max}}}\right) && (2.128) \end{aligned}$$

□

For large real η, the following asymptotic expansion of the complementary error function exists [70]:

$$\operatorname{erfc}(\eta) = \frac{e^{-\eta^2}}{\eta\sqrt{\pi}} \sum_{n=0}^{\infty} (-1)^n \frac{(2n-1)!!}{(2\eta^2)^n} \tag{2.129}$$

The realistic values $\dot{x}_{max} = 5$, $T_i = 10^{-5}$, $|\dot{x}_i(0)| = 2$ and $\alpha = 0.1\sqrt{T_i}$, yield a $\eta = 2.1143 \times 10^6$. The corresponding error

$$2\operatorname{erfc}(\eta) - \operatorname{erfc}(3\eta) < 10^{-10^{12}} \tag{2.130}$$

and is negligible.

We will now give an expression for the approximate transition probability $P_{\boldsymbol{\xi}_i(0)}(\boldsymbol{\xi}'_i(\delta t'_e[n+1]) \in A)$ from the previous theorem.

Theorem 6. *Assume the transmitter motion $\boldsymbol{\xi}'_i(t)$ is as defined in Definition 2, the receiver motion $\boldsymbol{\xi}_l(t)$ is given at the times nT_i and $\boldsymbol{\xi}'_i(0)$ is the initial condition for the motion process $\boldsymbol{\xi}'_i(t)$. Let*

$$\begin{aligned}\delta t'_e[n+1] = \inf\Bigg\{\delta t'_e : T_i - \delta t'_e = \frac{1}{c}\Big(&|x_l((n+1)T_i) - x'_i(0)| \dots \\ &- \operatorname{sgn}(x_l((n+1)T_i) - x'_i(0)) \int_0^{\delta t'_e} \dot{x}'_i(\tau)d\tau - |x_l(nT_i) - x'_i(0)|\Big)\Bigg\}\end{aligned} \tag{2.131}$$

Then

$$x'_i(\delta t'_e[n+1]) = x'_i(0) + \delta t'_e[n+1]\dot{x}'_i(0) - \alpha \operatorname{sgn}(x_l((n+1)T_i) - x'_i(0))I' \tag{2.132}$$

with

$$I' = -\beta - \delta t'_e[n+1]\gamma \tag{2.133}$$

$$\beta = \frac{1}{\alpha}(-cT_i + |x_l((n+1)T_i) - x'_i(0)| - |x_l(nT_i) - x'_i(0)|) \tag{2.134}$$

$$\gamma = \frac{1}{\alpha}(c - \operatorname{sgn}(x_l((n+1)T_i) - x'_i(0))\dot{x}'_i(0)) \tag{2.135}$$

and

$$\dot{x}_i'(\delta t_e'[n+1]) = \dot{x}_i'(0) - \alpha \operatorname{sgn}(x_l((n+1)T_i) - x_i'(0))B' \tag{2.136}$$

The random variables $\delta t_e'[n+1]$ and B' have the joint distribution

$$\begin{aligned} P_{\beta,\gamma}(\delta t_e'[n+1] \in dt; B' \in dz) = |z| \, [p_t(\beta,\gamma;0,z) - \\ \int_0^t \int_0^\infty m(s,-|z|,\mu) p_{t-s}(\beta,\gamma;0,-\epsilon\mu) d\mu ds \Big] \mathbf{1}_R(z) dz dt \end{aligned} \tag{2.137}$$

where $R = [0,\infty]$ if $\beta < 0$, $R = (-\infty, 0]$ if $\beta > 0$, $\epsilon = \operatorname{sgn}(-\beta)$, the function

$$m(t,y,z) = \frac{3z}{\pi\sqrt{2}t^2} e^{-(2/t)(y^2-|y|z+z^2)} \left(\int_0^{4|y|z/t} e^{-3\theta/2} \frac{d\theta}{\sqrt{\pi\theta}} \right) \mathbf{1}_{[0,\infty]}(z) dz dt \tag{2.138}$$

and

$$\begin{aligned} p_t(u,v;x,y) = \frac{\sqrt{3}}{\pi t^2} \exp \Big[-\frac{6}{t^3}(u-x-ty)^2 \\ + \frac{6}{t^2}(u-x-ty)(v-y) - \frac{2}{t}(v-y)^2 \Big] \end{aligned} \tag{2.139}$$

Proof. First, we manipulate the equation in the definition of the hitting time $\delta t_e[n+1]$ in the theorem statement. This equation reads

$$\begin{aligned} T_i - \delta t_e' = \frac{1}{c} \big(|x_l((n+1)T_i) - x_i'(0)| \ldots \\ - \operatorname{sgn}(x_l((n+1)T_i) - x_i'(0)) \int_0^{\delta t_e'} \dot{x}_i'(\tau) d\tau - |x_l(nT_i) - x_i'(0)| \Big) \end{aligned} \tag{2.140}$$

Note that

$$\dot{x}_i'(\tau) = \dot{x}_i'(0) + \alpha B(\tau) \tag{2.141}$$

and that

$$B'(\tau) = -\operatorname{sgn}(x_l((n+1)T_i) - x_i'(0)) B(\tau) \tag{2.142}$$

is again a Brownian motion. Equation 2.140 is hence equivalent to

$$0 = \beta + \delta t'_e \gamma + \int_0^{\delta t'_e} B'(\tau) d\tau \tag{2.143}$$

and we have

$$\delta t'_e[n+1] = \inf \left\{ \delta t'_e : 0 = \beta + \delta t'_e \gamma + \int_0^{\delta t'_e} B'(\tau) d\tau \right\} \tag{2.144}$$

We define the random variable $B' = B'(\delta t_e[n+1])$. The joint distribution of the random variables $\delta t_e[n+1]$ and B' is known [63,64,66]. The function $p_t(u, v; x, y)$ in Equation 2.139 is the transition density of the bidimensional process $(\int_0^t B'(\tau)d\tau, B'(t))$, i.e.,

$$\begin{aligned} &P(\int_0^{t+s} B'(\tau)d\tau \in du, B'(t+s) \in dv | \int_0^s B'(\tau)d\tau = x, B'(s) = y) \\ =& p_t(u, v; x, y) \end{aligned} \tag{2.145}$$

for any $t, s > 0$. □

In summary, the above theorems show that the sampled bidimensional process $(x_i(t_e[n]), \dot{x}_i(t_e[n]))$ is Markov in n and Theorem 6 gives an excellent approximation of the transition kernel of this Markov chain.

The above derivations hold for the case of one-dimensional motion. We will now discuss how these ideas can be extended to the case of three-dimensional motion. Assume $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_l(t)$ denote the position of the transmitter and receiver in three-dimensional space at time t, respectively. We associate each coordinate of the transmitter and receiver position with an independent integrated reflected Brownian motion (IRBM) model as defined in Definition 3. We set $\dot{x}_{max}$ in this definition to be smaller than $c/\sqrt{3}$, so that $\|\dot{\boldsymbol{x}}_i(t)\| < c$ and $\|\dot{\boldsymbol{x}}_l(t)\| < c$. Then by Theorem 1, the emission times $t_e[n]$ are unique and, by a trivial extension of Theorem 3, they form a strictly increasing sequence of stopping times. The six-dimensional processes $\boldsymbol{\xi}_i(t) = (\boldsymbol{x}_i(t); \dot{\boldsymbol{x}}_i(t))$ and $\boldsymbol{\xi}_l(t) = (\boldsymbol{x}_l(t); \dot{\boldsymbol{x}}_l(t))$ are both Feller processes and sampling $\boldsymbol{\xi}_i(t)$ at $t = t_e[n]$ yields a homogeneous Markov chain assuming the receiver motion $\boldsymbol{\xi}_l(t)$ is given at all times nT_i. We have some ideas on how the transition kernel of this Markov chain could be approximated but have not established bounds to quantify the quality of our approximations. The remainder of this section will elaborate on these ideas. We will derive an approximation of the transition kernel $P(\boldsymbol{\xi}_i(t_e[n+1])|\boldsymbol{\xi}_i(t_e[n]))$ given the receiver motion $\boldsymbol{\xi}_l(t)$ at all times nT_i.

Let again $\delta t_e[n+1] = t_e[n+1] - t_e[n]$. By the definition of the emission times $t_e[n]$,

$$\begin{aligned} \delta t_e[n+1] = \inf \{ & \delta t_e : T_i - \delta t_e = \ldots \\ & \frac{1}{c} \left(\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n] + \delta t_e)\| - \|\boldsymbol{x}_l(nT_i) - \boldsymbol{x}_i(t_e[n])\| \right) \Big\} \end{aligned} \tag{2.146}$$

or equivalently

$$\begin{aligned} \delta t_e[n+1] = \inf \{ & \delta t_e : T_i - \delta t_e = \ldots \\ & \frac{1}{c} \left(\left\| \boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n]) - \int_0^{\delta t_e} \dot{\boldsymbol{x}}_i(t_e[n] + \tau) d\tau \right\| \ldots \right. \\ & \left. - \|\boldsymbol{x}_l(nT_i) - \boldsymbol{x}_i(t_e[n])\| \right) \} \end{aligned} \tag{2.147}$$

Assume $\boldsymbol{V}$ is a rotation matrix such that $\boldsymbol{V}(\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])) = \boldsymbol{e}_1 \|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\|$, where $\boldsymbol{e}_1$ is the unit vector that has all coordinates set equal zero other than the first one. We denote the q-th component of the vector $\boldsymbol{V}\dot{\boldsymbol{x}}_i(t_e[n] + \tau)$ by $\dot{x}^v_{i;q}(t_e[n] + \tau)$. Similarly to the assumption in Equation 2.104 in Theorem 5, we will assume that

$$\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\| > \sqrt{3}\dot{x}_{max}\delta t_{max} \tag{2.148}$$

where

$$\delta t_{max} = T_i \left(\frac{1 + \frac{\sqrt{3}\dot{x}_{max}}{c}}{1 - \frac{\sqrt{3}\dot{x}_{max}}{c}} \right) \tag{2.149}$$

can be shown to upper bound $\delta t_e[n+1]$ by the same arguments made in the proof of Theorem 3 for the one dimensional case. This implies that

$$\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\| > \int_0^{\delta t_e} \dot{x}^v_{i;1}(t_e[n] + \tau) d\tau \tag{2.150}$$

For the one-dimensional case, the proof of Theorem 5 shows that the transition kernel is essentially unaffected when we condition on the event $\{|\dot{x}_i(t_e[n]+\tau)| < \dot{x}_{max}, 0 \leq \tau \leq \delta t_{max}\}$. We conjecture that an analogous statement holds for the three-dimensional case, i.e., we conjecture that the transition kernel $P(\boldsymbol{\xi}_i(t_e[n+1])|\boldsymbol{\xi}_i(t_e[n]))$ is essentially unaffected when we condition on the event $\{|\dot{x}_{i;q}(t_e[n]+\tau)| < \dot{x}_{max}, 0 \leq \tau \leq \delta t_{max}, q \in [3]\}$ where we denoted the q-th component of the vector $\dot{\boldsymbol{x}}_i(t_e[n]+\tau)$ by $\dot{x}_{i;q}(t_e[n]+\tau)$. We will assume this condition for our derivations below. Note that we can then write

$$\dot{\boldsymbol{x}}_i(t_e[n]+\tau) = \dot{\boldsymbol{x}}_i(t_e[n]) + \alpha \boldsymbol{B}(\tau) \tag{2.151}$$

for some three-dimensional Brownian motion $\boldsymbol{B}(\tau), \tau \geq 0$, that is independent of $\dot{\boldsymbol{x}}_i(t_e[n]+\tau), \tau \leq 0$. Due to the spherical symmetry of Brownian motion, $\boldsymbol{B}'(t) = \boldsymbol{V}\boldsymbol{B}(t)$ is again a three dimensional Brownian motion.

We then have

$$\begin{aligned}
&\left\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n]) - \int_0^{\delta t_e} \dot{\boldsymbol{x}}_i(t_e[n]+\tau)d\tau\right\| \\
&= \left\|\boldsymbol{V}(\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])) - \int_0^{\delta t_e} \boldsymbol{V}\dot{\boldsymbol{x}}_i(t_e[n]+\tau)d\tau\right\| && (2.152) \\
&= \left\|\boldsymbol{e}_1 \left\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\right\| - \int_0^{\delta t_e} \boldsymbol{V}\dot{\boldsymbol{x}}_i(t_e[n]+\tau)d\tau\right\| && (2.153) \\
&\approx \left|\left\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\right\| - \int_0^{\delta t_e} \dot{x}^v_{i;1}(t_e[n]+\tau)d\tau\right| && (2.154) \\
&= \left\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\right\| - \int_0^{\delta t_e} \dot{x}^v_{i;1}(t_e[n]+\tau)d\tau && (2.155) \\
&= \left\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\right\| - \dot{x}^v_{i;1}(t_e[n])\delta t_e - \alpha \int_0^{\delta t_e} {B'}_1(d\tau) && (2.156)
\end{aligned}$$

Equation 2.152 holds because $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$. By the far-field approximation, we can neglect the contribution of the second and third vector component in the argument of the norm in Equation 2.153. Equation 2.155 follows from Inequality 2.150 and Equation 2.156 follows from Equation 2.151.

Substituting $\left\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n]) - \int_0^{\delta t_e} \dot{\boldsymbol{x}}_i(t_e[n]+\tau)d\tau\right\|$ in the definition of the hitting time $\delta t_e[n+1]$ in Equation 2.147 by the approximation in Equation 2.156 gives the hitting time

$$\delta t'_e[n+1] = \inf \left\{ \delta t_e : T_i - \delta t_e = \frac{1}{c} \left(\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\| \ldots \right.\right.$$
$$\left.\left. - \dot{x}^v_{i;1}(t_e[n])\delta t_e - \alpha \int_0^{\delta t_e} B'_1(d\tau)d\tau - \|\boldsymbol{x}_l(nT_i) - \boldsymbol{x}_i(t_e[n])\| \right) \right\} \tag{2.157}$$

The far-field approximation above allows us to approximate $\delta t_e[n+1]$ by $\delta t'_e[n+1]$. This simplifies the problem significantly and allows us to use the machinery we have developed for the case of one-dimensional motion. The joint distribution of $\delta t'_e[n+1]$ and $B'_1(\delta t'_e[n+1])$ is identical to the joint distribution of $\delta t'_e[n+1]$ and B' given in Theorem 6 when $x_l((n+1)T_i) - x'_i(0)$ is set to $\|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\|$, $\dot{x}'_i(0)$ is set to $\dot{x}^v_{i;1}(t_e[n])$ and $|x_l(nT_i) - x'_i(0)|$ is set to $\|\boldsymbol{x}_l(nT_i) - \boldsymbol{x}_i(t_e[n])\|$. Further, by the definition of $\delta t'_e[n+1]$ in Equation 2.157,

$$\int_0^{\delta t'_e[n+1]} B'_1(d\tau)d\tau = \alpha^{-1} \left(-c(T_i - \delta t'_e[n+1]) + \|\boldsymbol{x}_l((n+1)T_i) - \boldsymbol{x}_i(t_e[n])\| \right.$$
$$\left. \ldots - \dot{x}^v_{i;1}(t_e[n])\delta t'_e[n+1] - \|\boldsymbol{x}_l(nT_i) - \boldsymbol{x}_i(t_e[n])\| \right) \tag{2.158}$$

The far-field approximation we made above essentially renders the second and third coordinate of the three-dimensional Brownian motion $\boldsymbol{B}'(t)$ independent of $\delta t'_e[n+1]$. Given $\delta t'_e[n+1]$, the bidimensional random vectors $(\int_0^{\delta t'_e[n+1]} B'_2(\tau)d\tau, B'_2(\delta t'_e[n+1]))$ and $(\int_0^{\delta t'_e[n+1]} B'_3(\tau)d\tau, B'_3(\delta t'_e[n+1]))$ are hence independent and distributed like $(\int_0^{\delta t'_e[n+1]} B'(\tau)d\tau, B'(\delta t'_e[n+1]))$ for some independent Brownian motion $B'(t)$. The transition probability of the bidimensional process $(\int_0^t B'(\tau)d\tau, B'(t))$ is known and given in Equation 2.145 [64]. Multiplying the joint densities of these bidimensional random vectors, we can compute the distribution

$$P\left(\int_0^{\delta t'_e[n+1]} \boldsymbol{B}'(\tau)d\tau \in d\boldsymbol{u}, \boldsymbol{B}'(\delta t'_e[n+1]) \in d\boldsymbol{v} \right)$$

Since

$$\boldsymbol{x}_i(t_e[n+1]) = \boldsymbol{x}_i(t_e[n]) + \int_0^{\delta t'_e[n+1]} \dot{\boldsymbol{x}}_i(t_e[n]+\tau)d\tau \tag{2.159}$$

$$= \boldsymbol{x}_i(t_e[n]) + \dot{\boldsymbol{x}}_i(t_e[n])\delta t'_e[n+1] + \alpha \boldsymbol{V}^T \int_0^{\delta t'_e[n+1]} \boldsymbol{B}'(d\tau)d\tau \tag{2.160}$$

and

$$\dot{\boldsymbol{x}}_i(t_e[n+1]) = \dot{\boldsymbol{x}}_i(t_e[n]) + \alpha \boldsymbol{V}^T \boldsymbol{B}'(\delta t'_e[n+1]) \tag{2.161}$$

the random vector $(\boldsymbol{x}_i(t_e[n+1]); \dot{\boldsymbol{x}}_i(t_e[n+1]))$ can be obtained from the random vector $(\int_0^{\delta t'_e[n+1]} \boldsymbol{B}'(\tau)d\tau; \boldsymbol{B}'(\delta t'_e[n+1]))$ by a simple affine transformation.

2.10 Bayesian State Inference

On a high level, the above sections introduced a prior distribution on all relevant system states: the transmitted symbols (Equation 2.31), the channel gains (Equation 2.36), the receiver motion (Equation 2.37) and the transmitter motion (Theorem 4 and 6). Further, we defined likelihood functions of the observable data given these states (Equation 2.34). Theoretically, this is sufficient to deduce the a posteriori distribution of the states and hence obtain estimates according to any given cost function. But we have found inference to be tractable only in some special cases, where we abstain from trying to jointly estimate all states, but instead assume that some of the states are known. We will first look at the case of a stationary transmitter.

2.10.1 Stationary Transmitter

We assume array i rests in the origin and transmits and array l is mobile and receives. If the transmit array has only one element, assuming our isotropic spreading model the generated acoustic field is spherically symmetric and the receiver cannot uniquely determine its position and orientation. In fact, the locus of possible positions is a sphere. However, if there are at least three elements on the receive array and the receiver has access to a compass and a tilt sensor, this symmetry can be broken. Accelerometers are inexpensive and can determine tilt reliably. The accuracy of magnetic compasses is compromised by a submarine's shielding ferric hull, but gyro-compasses do not have this problem and are well suited for this task, because they rely on the effect of gyroscopic precession instead of the Earth's magnetic field [71]. Measurements from inertial sensors can easily be included for state inference as we will explain below. But for now we will assume that the transmitter has more than three elements and hence circumvent this problem. We will further assume that there is no multi-path and that the transmitted signals are known.

Given these assumptions, the sampled output equations from Equation 2.34 specialize to

$$r_{l,m}[n] = \sum_{j,k} h_{i,j;l,m}[n,k] e^{2\pi\sqrt{-1} f_{Ci}(t_{i,j;l,m}[n]-nT_i)} s_{i,j}(t_{i,j;l,m}[n] - kT_i) + v_{l,m}[n] \tag{2.162}$$

where $t_{i,j;l,m}[n]$ can now be solved for explicitly

$$t_{i,j;l,m}[n] = nT_i - \frac{||\boldsymbol{x}_{l,m}(nT_i) - \boldsymbol{x}_{i,j}||}{c} \tag{2.163}$$

and the noise $v_{l,m}[n]$ is i.i.d.. We model the channel gains $h_{i,j;l,m}[n,k]$ as described in Section 2.7 and model the receiver position $\boldsymbol{x}_l(nT_i)$ and orientation $\boldsymbol{\theta}_l(nT_i)$ as described in Section 2.8. The signals $s_{i,j}(t)$ are assumed to be known.

Equations 2.36 and 2.37 define a linear state space system driven by Gaussian noise and Equations 2.162 define non-linear output equations. Several inference methods have been developed for such systems. The extended Kalman filter (EKF) is popular and basically linearizes the equations around the current estimate in each step and then applies the standard Kalman filter equations. This algorithm is considered the de facto standard in navigation systems and GPS [72]. When the state equations or the output equations are highly non-linear as in Equation 2.162, the EKF can, however, give poor performance [73].

The application of the Kalman filter to a nonlinear system requires the computation of the first two moments of the state vector and the observations. This problem can be viewed as a specific case of a more general problem: the calculation of the statistics of a random vector after a nonlinear transformation. The unscented transformation attacks this problem with a deterministic sampling technique. It determines a set of points (called sigma points) that accurately capture the true mean and covariance of the sampled random vector. The nonlinear transformation is then applied on each of these points, which results in samples of the transformed random vector and a new sample mean and covariance can be computed. It can be shown analytically that the resulting unscented Kalman filter (UKF) is superior to the EKF but has the same computational complexity [73]. Developing the Taylor series expansions of the posterior mean and covariance shows that sigma points capture these moments accurately to the second order for any nonlinearity. For the EKF, only the accuracy of the first order terms can be guaranteed [74].

We implemented an UKF for inference on the model presented here. We played a known signal (a $100Hz$ wide pulse at a center frequency of $25kHz$) from a speaker and fed the UKF with the measurements from a moving microphone. For this simple one-dimensional setup, we verified that this approach yields position estimates with a precision of less than a millimeter.

Inertial sensors provide additional information about the trajectory to be tracked. Accelerometers, for example, provide noisy observations of the acceleration that the sensor experiences and gyroscopes measure experienced angular velocity. When we use such sensors on the mobile receiver, the generated observations and measurements are easily taken into account by adding additional output equations to the state space system describing the position and orientation of the receiver array. This combination of sensory data is called sensor fusion. Measurements $a_{l;k}[n]$ of the acceleration values $x^{(2)}_{l;k}(nT_i)$ could for example be incorporated by adding the output equations

$$a_{l;k}[n] = x^{(2)}_{l;k}(nT_i) + u_k[n] \tag{2.164}$$

where $u_k[n]$ is assumed to be white Gaussian noise.

Ideally, we would like to not only track the receiver position and orientation, but also communicate data. As described in Section 2.6 we use broadband transmission signals $s_{i,j}(t)$ of the form

$$s_{i,j}(t) = \sum_{l \in [0:N]} s_{i,j}[l] p(t - lT) \tag{2.165}$$

In order to communicate information from the transmitter to the receiver, we could assume some of the symbols $s_{i,j}[l]$ to be unknown, i.i.d. random variables with a uniform distribution over the possible constellation points and then estimate those unknown symbols jointly with the channel attenuation and motion states. However, we find joint estimation of all these states difficult and hard to implement for several reasons.

We want the pulse $p(t)$ to be band-limited because as discussed in Section 2.5 the channel is band-limited. But in order for $p(t)$ to have most of its energy in a finite band, the pulse length needs to be large and hence, for any time t , the value of $s_{i,j}(t)$ depends on many symbols $s_{i,j}[l]$. The signals $s_{i,j}(t)$ are sampled at the random times $t_{i,j;l,m}[n] - kT_i$ in Equation 2.162 and

$$s_{i,j}(t_{i,j;l,m}[n] - kT_i) = \sum_{l \in [N]_0} s_{i,j}[l] p(t_{i,j;l,m}[n] - kT_i - lT) \tag{2.166}$$

The computational complexity of the EKF or the UKF is quadratic in the dimension of the state vector and both methods require that a state space model for the states to be estimated is available. The only state space system for the unknown symbols in the sequence, $s_{i,j}[l]$, we could find is a trivial one with very large dimensionality:

$$s_{i,j}[l] = s_{i,j}[l], \, \forall j \in [K] \text{ and } l \in S_u \tag{2.167}$$

where the set S_u contains the indices of the unknown symbols. This would make the complexity of each EKF or UKF step quadratic in the size of S_u, which is impractical. Another idea would be to run a particle filter on this high dimensional state space system and then to only update those indices in S_u in each step, which are in the vicinity of $\lfloor t_{i,j;l,m}[n]/T_i - k \rceil$. We have not investigated this approach further but instead focused on a low complexity deterministic approach for joint data and channel estimation.

2.11 Deterministic Inference

This section will describe a method that facilitates reliable communication over the underwater acoustic channel and at the same time is computationally tractable enough to allow for an implementation on modern embedded computing platforms. For a moment we will assume that array i is mobile and transmits while array l is stationary and receives. Array i is trivial and only carries one transducer. Array l carries K transducers. We account for multi-path effects but assume that the Doppler is the same on all paths. This is a good approximation when all phantom sources are near each other, as is the case in the long range shallow water channel for example. Since the transmit array is assumed trivial, no index is needed to enumerate its elements. Without loss of generality we can assume the parameters i and l fixed. In what follows there will be no ambiguity as to which of the two arrays we are referring to and we hence drop the indices i and j for the sake of notational simplicity.

We send a signal $s(t)$ of the form described in Section 2.6 and choose the symbols $s[l]$ from an q-ary QAM constellation. Some of these symbols are known and used for training. Some are unknown and used for data communication.

Under the above assumptions the demodulated received signals from Equation 2.33 simplify to

$$r_m(t) = \int_\tau h_m(t,\tau) e^{2\pi\sqrt{-1} f_C (t_m(t)-\tau-t)} s(t_m(t)-\tau) d\tau + v_m(t) \tag{2.168}$$

where m indexes the receiving transducers and the emission time $t_m(t)$ solves the implicit equation

$$t - t_m(t) - \frac{||\boldsymbol{x}_m(t) - \boldsymbol{x}(t_m(t))||}{c} = 0 \tag{2.169}$$

The sent signal $s(t)$ has a bandwidth of $1/T$ and we can hence represent the integral in Equation 2.168 as a sum:

$$r_m(t) = \sum_k h_{m;k}(t) e^{2\pi\sqrt{-1} f_C (t_m(t)-t)} s(t_m(t)-kT) + v_m(t) \tag{2.170}$$

where

$$h_{m;k}(t) = Th_m(t, kT)e^{-2\pi\sqrt{-1}f_C kT} \tag{2.171}$$

is the demodulated and sampled kernel.

We define the sequence of arrival times $t_m^{-1}[n]$ as the solutions to the implicit equation

$$t_m^{-1}[n] - nT - \frac{||\boldsymbol{x}_m(t_m^{-1}[n]) - \boldsymbol{x}(nT)||}{c} = 0 \tag{2.172}$$

We abbreviate $\boldsymbol{x}(nT)$ by $\boldsymbol{x}[n]$ and the derivative of $\boldsymbol{x}(t)$ at time nT by $\dot{\boldsymbol{x}}[n]$. Since the receiver was assumed stationary, we can solve for $t_m^{-1}(nT)$ explicitly

$$t_m^{-1}[n] = nT + \frac{||\boldsymbol{x}_m - \boldsymbol{x}(nT)||}{c} \tag{2.173}$$

The arrival times $t_m^{-1}[n]$ are the inverse of the function $t_m(t)$ evaluated at the times nT. They specify when a hypothetical impulse sent from the transmitter at time nT, would arrive at the m-th receiving transducer.

If we sample the signal from Equation 2.170 at $t = t_m^{-1}[n]$, we get

$$r_m(t_m^{-1}[n]) = \sum_k h_{m;k}(t_m^{-1}[n])e^{2\pi\sqrt{-1}f_C(nT - t_m^{-1}[n])}s[n-k] + v_m(t_m^{-1}[n]) \tag{2.174}$$

And if we further multiply both sides of this equation by $e^{-2\pi\sqrt{-1}f_C(nT - t_m^{-1}[n])}$, we obtain

$$e^{-2\pi\sqrt{-1}f_C(nT - t_m^{-1}[n])}r_m(t_m^{-1}[n]) = \sum_k h_m[n,k]s[n-k] + v_m[n] \tag{2.175}$$

where $h_m[n,k] = h_{m;k}(t_m^{-1}[n])$ and $v_m[n]$ is some noise sequence.

These equations motivate a direct equalization estimator for the symbols $s[n]$ of the following form:

$$s[n] \approx \hat{s}_n = \sum_{m,k} w[n,m,k]r_m(t_m^{-1}[n-k])e^{-2\pi\sqrt{-1}f_C((n-k)T - t_m^{-1}[n-k])} \tag{2.176}$$

where $w[n,m,k]$ are the complex-valued equalizer weights. We will assume that k ranges from $-M_A$ to M_C for some positive integers M_A and M_C

and that $M = M_A + M_C + 1$. To reduce the number of parameters of this estimator, we define the function

$$t^{-1}_{m;n,k}(\boldsymbol{x}[n], \dot{\boldsymbol{x}}[n]) = (n-k)T + \frac{||\boldsymbol{x}_m - \boldsymbol{x}[n] + kT\dot{\boldsymbol{x}}[n]||}{c} \tag{2.177}$$

and substitute $t^{-1}_m[n-k]$ by $t^{-1}_{m;n,k}(\boldsymbol{x}[n], \dot{\boldsymbol{x}}[n])$ in Equation 2.176. The resulting estimator is $\hat{s}_n(\boldsymbol{\theta}[n])$, where the parameter vector $\boldsymbol{\theta}[n] \in \mathbb{R}^{2MK+6}$ is such that its components $\theta_z[n]$ satisfy

$$\theta_z[n] = \begin{cases} \mathrm{Re}(w[n,m,k-M_A]); & z = 2kK + 2m - 1, k \in [0\!:\!M-1], m \in [K] \\ \mathrm{Im}(w[n,m,k-M_A]); & z = 2kK + 2m, k \in [0\!:\!M-1], m \in [K] \\ x_q[n]; & z = 2MK + q, q \in [3] \\ \dot{x}_q[n]; & z = 2MK + 3 + q, q \in [3] \end{cases} \tag{2.178}$$

This notation formalizes that the equalizer weights $w[n,m,k-M_A], k \in [0 : M-1], m \in [K]$, the position $\boldsymbol{x}[n]$ and the velocity $\dot{\boldsymbol{x}}[n]$ are concatenated into one real-valued parameter vector, the vector $\boldsymbol{\theta}[n]$. The function $\hat{s}_n(\boldsymbol{\theta})$ reads

$$\begin{aligned} \hat{s}_n(\boldsymbol{\theta}) = \sum_{k\in[0:M-1],m\in[K]} & \left(\theta_{2kK+2m-1} + \sqrt{-1}\,\theta_{2kK+2m}\right) \cdot \ldots \\ & r_m(t^{-1}_{m;n,k-M_A}(\theta_{2MK+[3]}, \theta_{2MK+3+[3]})) \cdot \ldots \\ & e^{-2\pi\sqrt{-1}f_C((n-k+M_A)T - t^{-1}_{m;n,k-M_A}(\theta_{2MK+[3]},\theta_{2MK+3+[3]}))} \end{aligned} \tag{2.179}$$

We define the objective function

$$\begin{aligned} L_n = & \frac{1}{2}(\boldsymbol{\theta}[0] - \hat{\boldsymbol{\theta}})^T \boldsymbol{C}^{-1}(\boldsymbol{\theta}[0] - \hat{\boldsymbol{\theta}}) + \sum_{l=0}^{n} |s[l] - \hat{s}_l(\boldsymbol{\theta}[l])|^2 \sigma_s^{-2} \\ & + \frac{1}{2}\sum_{l=0}^{n-1}(\boldsymbol{\theta}[l+1] - \boldsymbol{T}\boldsymbol{\theta}[l])^T \boldsymbol{Q}^{-1}(\boldsymbol{\theta}[l+1] - \boldsymbol{T}\boldsymbol{\theta}[l]) \end{aligned} \tag{2.180}$$

for some number of known training symbols $s[l], l = 0, \ldots, n$ and choose the parameter vector $\boldsymbol{\theta}[n]$ such that

$$\boldsymbol{\theta}[n] = \underset{\boldsymbol{\theta}[n]}{\mathrm{argmin}} \min_{\boldsymbol{\theta}[l], l\in[0,n-1]} L_n \tag{2.181}$$

The vector $\hat{\boldsymbol{\theta}}$ is the initial guess we have about the parameter vector $\boldsymbol{\theta}[0]$ and $\boldsymbol{C}$ is a covariance matrix specifying how much confidence we have in this guess. The scalar σ_s^{-2} is a weighting factor and the matrix $\boldsymbol{Q}^{-1}$ is a weighting matrix. The matrix $\boldsymbol{T}$ is a transition matrix with $\boldsymbol{T}\boldsymbol{\theta}[n]$ specifying an estimate of $\boldsymbol{\theta}[n+1]$. Note that we allow for some error $(\boldsymbol{\theta}[n+1]-\boldsymbol{T}\boldsymbol{\theta}[n])$ in this plant model. We choose a simple transition matrix $\boldsymbol{T}$ with components $T_{z,u}$ such that

$$T_{z,u} = \begin{cases} 1; & z = u, u \in [2MK+6] \\ T; & z = 2MK+q, u = 2MK+3+q, q \in [3] \\ 0; & \text{otherwise} \end{cases} \tag{2.182}$$

The 6×6 submatrix on the bottom right of $\boldsymbol{T}$ is the transition matrix for the position $\boldsymbol{x}[n]$ and the velocity $\dot{\boldsymbol{x}}[n]$ according to the motion model presented in Section 2.8 for $d = 2$.

The extended Kalman filter is known to find an approximate solution to the least squares problem in Equation 2.180 when run on the state space system

$$\boldsymbol{\theta}[n+1] = \boldsymbol{T}\boldsymbol{\theta}[n] + \boldsymbol{\nu}_{n+1} \tag{2.183}$$

and the output equations

$$s[n] = \hat{s}_n(\boldsymbol{\theta}[n]) + v_n \tag{2.184}$$

for $n \geq 0$ [75]. The noise values v_n are independent, mean-zero, circular symmetric complex Gaussian random variables with variance σ_s^2. We denote the estimate of $\boldsymbol{\theta}[n+1]$ given the symbols $\{s[l], l \in [0:n]\}$ by $\hat{\boldsymbol{\theta}}[n+1,n]$. The initial state estimate $\hat{\boldsymbol{\theta}}[0,-1]$ is a Gaussian random vector with mean $\hat{\boldsymbol{\theta}}$ and covariance $\boldsymbol{C}$. The random vectors $\boldsymbol{\nu}_n$ are independent and mean-zero. Each vector $\boldsymbol{\nu}_n$ is Gaussian with covariance $\boldsymbol{Q}$. The covariance matrix $\boldsymbol{Q}$ is chosen such that its components $Q_{z,u}$ satisfy

$$Q_{z,u} = \begin{cases} \sigma_w^2; & z = u, u \in [2MK] \\ \sigma_a^2 \frac{T^4}{4}; & z = u, u \in 2MK + [3] \\ \sigma_a^2 T^2; & z = u, u \in 2MK + 3 + [3] \\ \sigma_a^2 \frac{T^3}{2}; & z = 2MK + q, u = 2MK + 3 + q, q \in [3] \\ \sigma_a^2 \frac{T^3}{2}; & z = 2MK + 3 + q, u = 2MK + q, q \in [3] \\ 0; & \text{otherwise} \end{cases} \tag{2.185}$$

for some variances σ_w^2 and σ_a^2. The 6×6 submatrix on the bottom right of $\boldsymbol{Q}$ is the covariance matrix for the position $\boldsymbol{x}[n]$ and the velocity $\dot{\boldsymbol{x}}[n]$ according to the motion model presented in Section 2.8 for $d = 2$. The $2MK \times 2MK$ submatrix on the top left of $\boldsymbol{Q}$ is diagonal and hence renders the evolution of all equalizer weights independent. The equalizer weights are assumed to be independent of the position and velocity of the transmitter.

We perform a method akin to decision directed equalization to obtain estimates of the symbols $s[n]$ that are unknown and used for communication. In total $N + 1$ symbols $s[n]$ are sent. We assume the first N_{pre} symbols $\{s[l], l \in [0 : N_{pre} - 1]\}$ and also a fraction of the subsequent symbols to be known. Let the set $S_u \subset [0 : N]$ contain the indices of the unknown symbols. We first run the extended Kalman filter on the known first N_{pre} symbols $\{s[l], l \in [0 : N_{pre} - 1]\}$ and obtain $\hat{\boldsymbol{\theta}}[N_{pre}, N_{pre} - 1]$. Now if $N_{pre} \in S_u$, then we find the point in the symbol constellation A that is closest to $\hat{s}_n(\hat{\boldsymbol{\theta}}[N_{pre}, N_{pre} - 1])$ and declare that point to be $s[N_{pre}]$. The operation of mapping a complex number to its nearest constellation point is called slicing. Now regardless of whether $N_{pre} \in S_u$, the symbol $s[N_{pre}]$ is available. So the extended Kalman filter can be updated and the next prediction $\hat{\boldsymbol{\theta}}[N_{pre} + 1, N_{pre}]$ can be computed. Now we check again if $N_{pre} + 1 \in S_u$ and, if so, we slice $\hat{s}_n(\hat{\boldsymbol{\theta}}[N_{pre} + 1, N_{pre}])$ and declare the slicer output to be the symbol $s[N_{pre} + 1]$. We iterate the Kalman update and prediction steps and the conditional slicing operation until the last symbol $s[N]$ is reached. At a high level, the estimator $\hat{s}_n(\boldsymbol{\theta})$ first resamples the received waveforms to undo any timing distortions and then filters the resampled signal to remove any frequency selectivity present in the channel. We hence call this estimator a resampling equalizer (RE). Algorithm 1 describes the operation of this equalizer in pseudocode. The function slice($\cdot$) performs the slicing operation.

```
Data: The transition matrix $\boldsymbol{T}$, the covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{C}$, the
      variance $\sigma_s^2$ and the initial estimate $\hat{\boldsymbol{\theta}}$ are given. Further, the set
      $S_u \subset [0:N]$ and the values of $s[n]$ for $n \notin S_u$ are given.
Result: The sequence of symbol estimates $\hat{s}_n, n \in [0, N]$, and the sequence
        of hard decisions $\bar{s}_n, n \in [0, N]$.
% initialization:
$\boldsymbol{P} = \boldsymbol{C}$;
for $n = [0:N]$ do
    $\hat{s}_n = \hat{s}_n(\hat{\boldsymbol{\theta}})$;
    if $n \in S_u$ then
        $\bar{s}_n = \text{slice}(\hat{s}_n)$;
    else
        $\bar{s}_n = s[n]$;
    end
    compute $\boldsymbol{g} \in \mathbb{C}^{1 \times 2MK+6}$, the numerical approximation to the gradient
    $\left.\frac{\partial \hat{s}_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}$;
    % perform Kalman update step:
    $e = \bar{s}_n - \hat{s}_n$;
    $\boldsymbol{S} = [\text{Re}(\boldsymbol{g}); \text{Im}(\boldsymbol{g})]\boldsymbol{P}[\text{Re}(\boldsymbol{g}); \text{Im}(\boldsymbol{g})]^T + \frac{1}{2}\sigma_s^2 \boldsymbol{I}$;
    $\boldsymbol{K} = \boldsymbol{P}[\text{Re}(\boldsymbol{g}); \text{Im}(\boldsymbol{g})]^T \boldsymbol{S}^{-1}$;
    $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} + \boldsymbol{K}[\text{Re}(e); \text{Im}(e)]$;
    $\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{K}[\text{Re}(\boldsymbol{g}); \text{Im}(\boldsymbol{g})])\boldsymbol{P}$;
    % perform Kalman prediction step:
    $\hat{\boldsymbol{\theta}} = \boldsymbol{T}\hat{\boldsymbol{\theta}}$;
    $\boldsymbol{P} = \boldsymbol{T}\boldsymbol{P}\boldsymbol{T}^T + \boldsymbol{Q}$;
end
```

Algorithm 1: The operation of the resampling equalizer (RE).

The extended Kalman filter requires the values of the partial derivatives $\frac{\partial \hat{s}_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ evaluated at $\hat{\boldsymbol{\theta}}[n, n-1], n \in [0:N]$. We approximate these numerically as shown in Algorithm 2.

Of course, it is not guaranteed that the slicer actually recovers the original symbol each time. The rate at which the slicer misses is called the symbol error rate (SER). We have found in our experiments that as long as the SER is below 20%, the Kalman filter remains stable. Each of the unknown QAM symbols $\{s[l], l \in S_u\}$ corresponds to a bit pattern. The receiver maps the sliced symbols back to their corresponding bit pattern and ideally the resulting bit sequence agrees with the bit sequence that was sent originally. In most cases, however, there will be bit errors and the rate at which these occur is called the bit error rate (BER). If the BER at the equalizer output is too

```
Data: The state vector $\hat{\boldsymbol{\theta}}$ is given. The constants $\epsilon, \delta > 0$ are some small
      real numbers.
Result: The vector $\boldsymbol{g} \in \mathbb{C}^{1\times 2MK+6}$ which approximates the gradient $\left.\frac{\partial \hat{s}_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}$.
% initialization:
$\boldsymbol{g} = \mathbf{0}_{1\times 2MK+6}$;
for $k = [0 : M-1]$ do
    for $m = [K]$ do
        $g_{2kK+2m-1} = r_m(t^{-1}_{m;n,k-M_A}(\theta_{2MK+[3]}, \theta_{2MK+3+[3]})) \cdot \ldots$
        $e^{-2\pi\sqrt{-1} f_C((n-k+M_A)T - t^{-1}_{m;n,k-M_A}(\theta_{2MK+[3]}, \theta_{2MK+3+[3]}))}$;
        $g_{2kK+2m} = \sqrt{-1}\, g_{2kK+2m-1}$;
    end
end
$\hat{\boldsymbol{\theta}}^{+} = \hat{\boldsymbol{\theta}}$;
for $q = [3]$ do
    $\hat{\theta}^{+}_{2MK+q} = \hat{\theta}^{+}_{2MK+q} + \epsilon$;
    $g_{2MK+q} = (\hat{s}_n(\hat{\boldsymbol{\theta}}^{+}) - \hat{s}_n(\hat{\boldsymbol{\theta}}))/\epsilon$;
    $\hat{\theta}^{+}_{2MK+q} = \hat{\theta}_{2MK+q}$;
    $\hat{\theta}^{+}_{2MK+3+q} = \hat{\theta}^{+}_{2MK+3+q} + \delta$;
    $g_{2MK+3+q} = (\hat{s}_n(\hat{\boldsymbol{\theta}}^{+}) - \hat{s}_n(\hat{\boldsymbol{\theta}}))/\delta$;
    $\hat{\theta}^{+}_{2MK+3+q} = \hat{\theta}_{2MK+3+q}$;
end
```

Algorithm 2: Numerical approximation of $\left.\frac{\partial \hat{s}_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}$.

high for a given application, channel coding can be used at the transmitter to reduce the BER at the expense of the rate the sequence of information bits is transmitted [76–78]. Channel coding adds redundancy to the sequence of information bits that is to be communicated. The enlarged bit sequence is mapped to QAM symbols. These symbols are unknown at the receiver and we call them information symbols. Our equalizer needs training, before any unknown symbols can be estimated, and we hence add in some known QAM symbols into this stream of information symbols. The resulting sequence carries information symbols at the indices $n \in S_u$ and known symbols at the other indices. At the receiver, the bit stream from the slicer output is fed into a channel decoder that uses the added redundancy to reduce the BER on the sequence of sent information bits. The required amount of redundancy depends on the equalizer output BER and the maximal permissible BER on the sequence of information bits. BER performance can be improved significantly if the equalizer and the channel decoder collaborate. There is vast literature on the field of iterative equalization and decoding (also known as turbo equalization) that describes how this collaboration should be furnished [79–84]. For these results to apply, the equalizer needs to be capable of leveraging soft information from the decoder and further needs to produce soft output instead of sliced hard decisions. There are standard methods available to extend direct equalizers like the one we introduced in this section so they fit this bill [81, 82, 85] and we refer to the given references for the details. When used in the setting of turbo equalization, we refer to our equalizer as a turbo resampling equalizer (TRE).

Let $\boldsymbol{x}[n{+}1, n]$ and $\dot{\boldsymbol{x}}[n{+}1, n]$ denote the position estimate $\hat{\boldsymbol{\theta}}_{2MK+[3]}[n{+}1, n]$ and the velocity estimate $\hat{\boldsymbol{\theta}}_{2MK+3+[3]}[n{+}1, n]$, respectively. In our simulations and experiments, we found that, in order for the Kalman filter to converge, the initial estimates of the transmitter position $\boldsymbol{x}[0, -1]$ and velocity $\dot{\boldsymbol{x}}[0, -1]$ must be accurate enough such that $t^{-1}_{m;0,0}(\boldsymbol{x}[0, -1], \dot{\boldsymbol{x}}[0, -1])$ deviates from $t^{-1}_{m;0,0}(\boldsymbol{x}[0], \dot{\boldsymbol{x}}[0])$ by at most about one symbol period T, for all $m \in [K]$. The trilateration method can be used to obtain estimates of $\boldsymbol{x}[0]$ and $\dot{\boldsymbol{x}}[0]$ [86]. We transmit two chirps before any QAM symbols are sent and then measure when each of these two chirps arrives at the receive transducers. Trilateration computes two estimates of the transmitter position from these arrival time measurements - one estimate for each transmitted chirp [86]. If we assume that the first chirp was sent at time $t = t_{C1}$ and that the second chirp was sent at a later time $t = t_{C2}$, then this method obtains estimates of the positions $\boldsymbol{x}(t_{C1})$ and $\boldsymbol{x}(t_{C2})$. The difference quotient $\left(\boldsymbol{x}(t_{C2}) - \boldsymbol{x}(t_{C1})\right)/(t_{C2} - t_{C1})$ gives the average velocity between the two times $t = t_{C1}$ and $t = t_{C2}$. We set $\dot{\boldsymbol{x}}[0, -1]$ equal to this average velocity and further set $\boldsymbol{x}[0, -1] = \boldsymbol{x}(t_{C2}) - t_{C2}\dot{\boldsymbol{x}}[0, -1]$.

For the derivation above we had assumed that the receiver array is stationary. If this assumption does not hold, we still use the introduced equalizer for communication and accept that the states $\boldsymbol{x}[n]$ and $\dot{\boldsymbol{x}}[n]$ no longer correspond to the position and velocity of the transmitter with respect to a fixed Cartesian frame of reference.

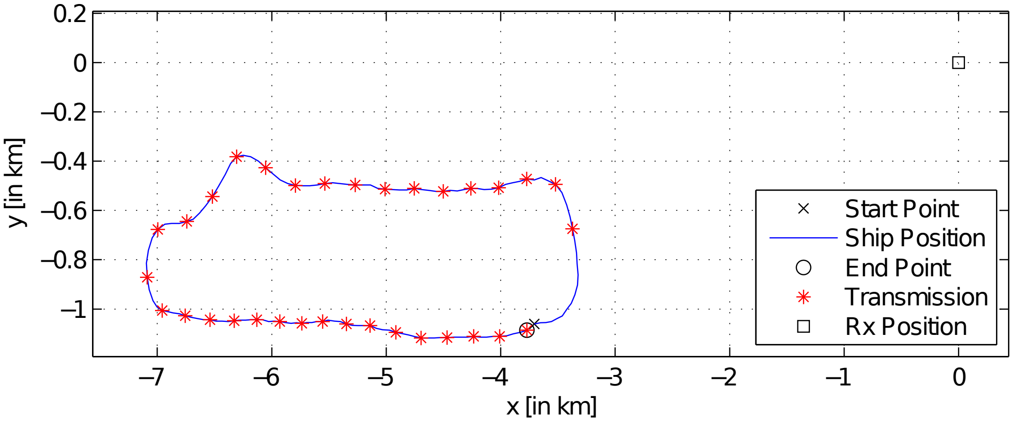

Figure 2.11: MACE10 Transmission Map Day 177.

2.12 Experimental Results

Our turbo resampling equalizer (TRE) has demonstrated unprecedented communication performance in US Navy sponsored field tests and simulations.

Some of our real data stems from the Mobile Acoustic Communications Experiment (MACE) conducted in June 2010 about 100 km south of Martha's Vineyard, MA. The depth at the site is approximately 100 m. A mobile V-fin with an array of transmit projectors attached was towed along a "race track" course approximately 3.8 km long and 600m wide. The maximum tow speed was 3 knot (1.5 m/s) and the tow depth varied between 30 and 60 m. The receive hydrophone array was moored at a depth of 50 m. Figure 2.11 shows a map centered around the location of the hydrophone array.

The red stars indicate the location of the projector array during the transmissions on day 177. The range between the transmit and receive array varied between $2.7km$ and $7.2km$. The weather was good throughout the 4 day experiment. Wednesday, June 23, was foggy and warm. The winds were calm. The signal transmission started on Thursday (day 175). The winds picked up to 10.6 m/s that day but laid down again Friday and Saturday.

One projector was used for signal emission and 2 hydrophones were used for reception. We employed a rate 1/2, $(131, 171)$ RSC code and puncturing to obtain an effective code rate of 2/3. Blocks of 19800 bits were generated, interleaved, and mapped to 16-QAM symbols. The carrier frequency was 13 kHz. The receive sampling rate was 39.0625 k samples/second. Data was transmitted at a symbol rate of 9.765625 k symbols/second. Taking into account the 10% overhead from equalizer training, we achieved a net data rate of $23.438kbps$. At a distance of $2.7km$ the equalizer output BER was below 10^{-6} and the overhead from equalizer training was 1%. The net data rate hence increased to about $39kbps$ A raised cosine filter with a roll-off factor 0.2 was used in both the transmitter and the receiver. Two chirps at the beginning of the data transmission and the measurement of their time dilation are used to find initial values for the transmitter velocity.

Figures 2.12, 2.13 and 2.14 summarize the bit error rate (BER) performance of our receiver on the MACE 2010 data set. Zero is displayed as 10^{-10} in the BER plots. For all transmissions our receiver converged to the right code word after two or less cycles. Figure 2.15 shows that the projected speed between transmitter and receiver fluctuated significantly giving rise to highly time-varying Doppler. Due to the shallow water at the experiment site, the channel exhibited severe multi-path as illustrated in Figure 2.16.

Since our interaction and discussions with the subsea oil and gas industry, we have begun to focus on communication over shorter distances while scaling up bandwidth and data rate. On our campus, in a $1.22m \times 1.83m \times 49m$ wave-tank, we have begun to experiment with a set of ITC-1089D transducers, which have around $200kHz$ of bandwidth at a center frequency of around $300kHz$. We recently achieved $1.2Mbps$ over a distance of $12m$ using this experimental setup. A 64-QAM constellation was employed and the equalizer output BER was about 10^{-3}. In a smaller tank, we reached rates of $100Mbps$ over distances of less than $1m$. For this experiment, we repurposed high frequency ultrasound transducers with a bandwidth of $20MHz$ and a center frequency of $20MHz$ and again transmitted 64-QAM symbols. The BER at the equalizer output was about 2×10^{-2}.

Table 2.1 compares the performance of our TRE method with competing approaches both from academia and industry. Speed values are maximum values with BER $< 10^{-9}$. The LinkQuest modem is representative of commercially available acoustic modems. The LinkQuest modem uses some proprietary spread spectrum (SS) method for communication. The WHOI modem uses frequency shift keying (FSK) for its robust $80bps$ mode. Both of these methods handle motion well but only provide damn low data rates. For their high data rate experiments, WHOI uses a combination of a phase-locked loop and standard linear decision feedback equalization (DFE) as devised in [33]. This method yields higher data rates than their FSK method but requires both transmitter and receiver to be near stationary. The at-sea

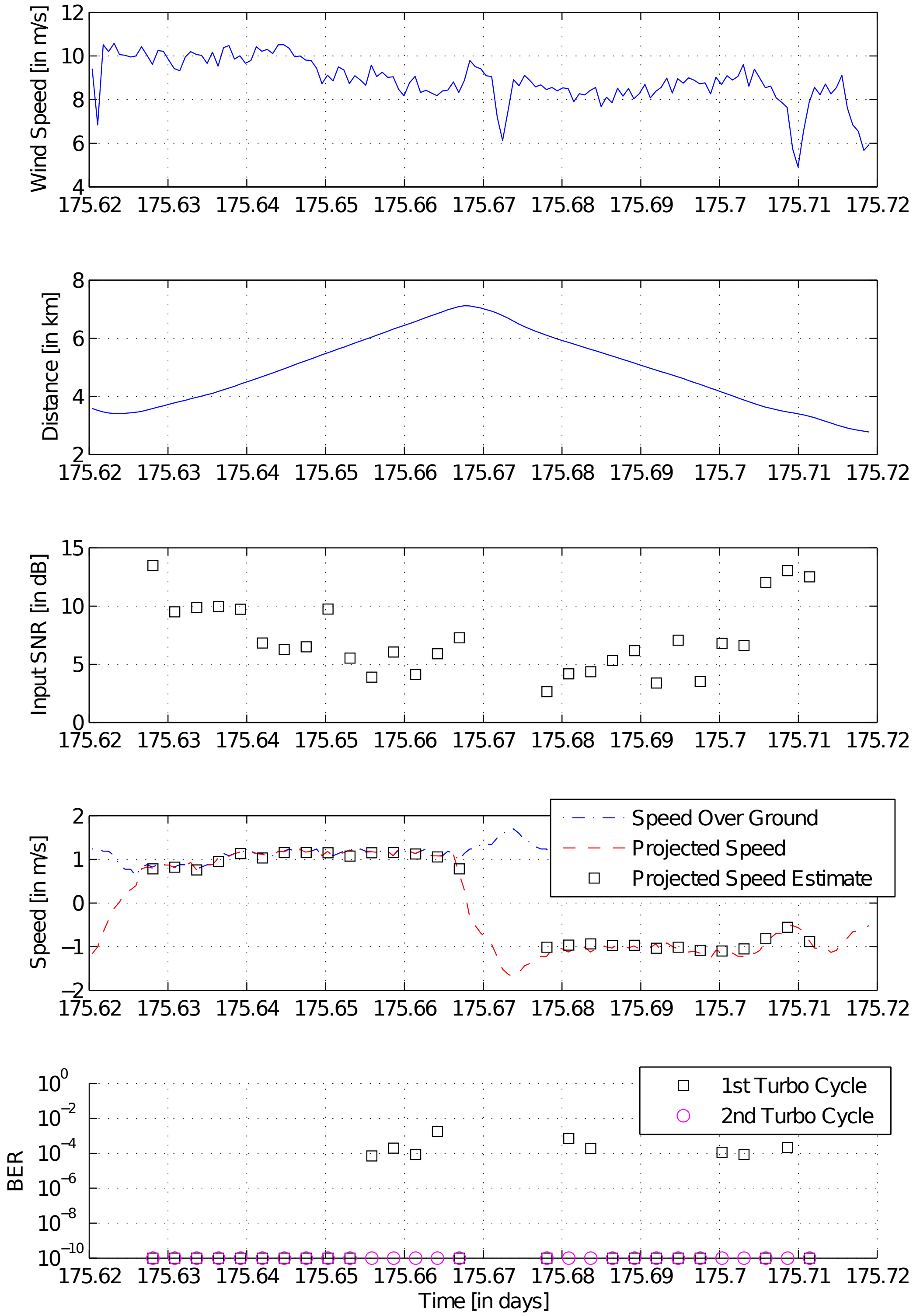

Figure 2.12: MACE10 Evaluation Day 175.

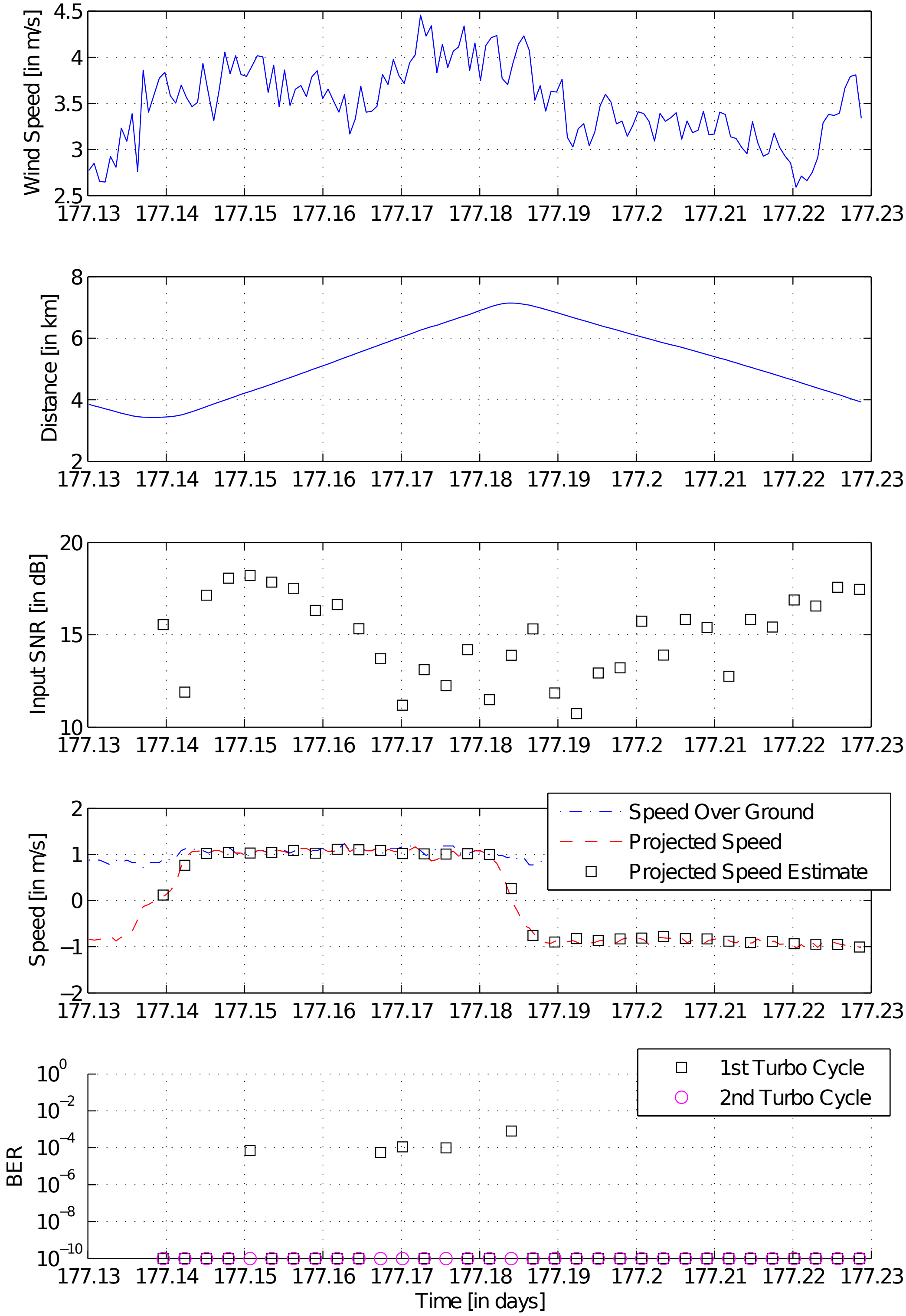

Figure 2.13: MACE10 Evaluation Day 177.

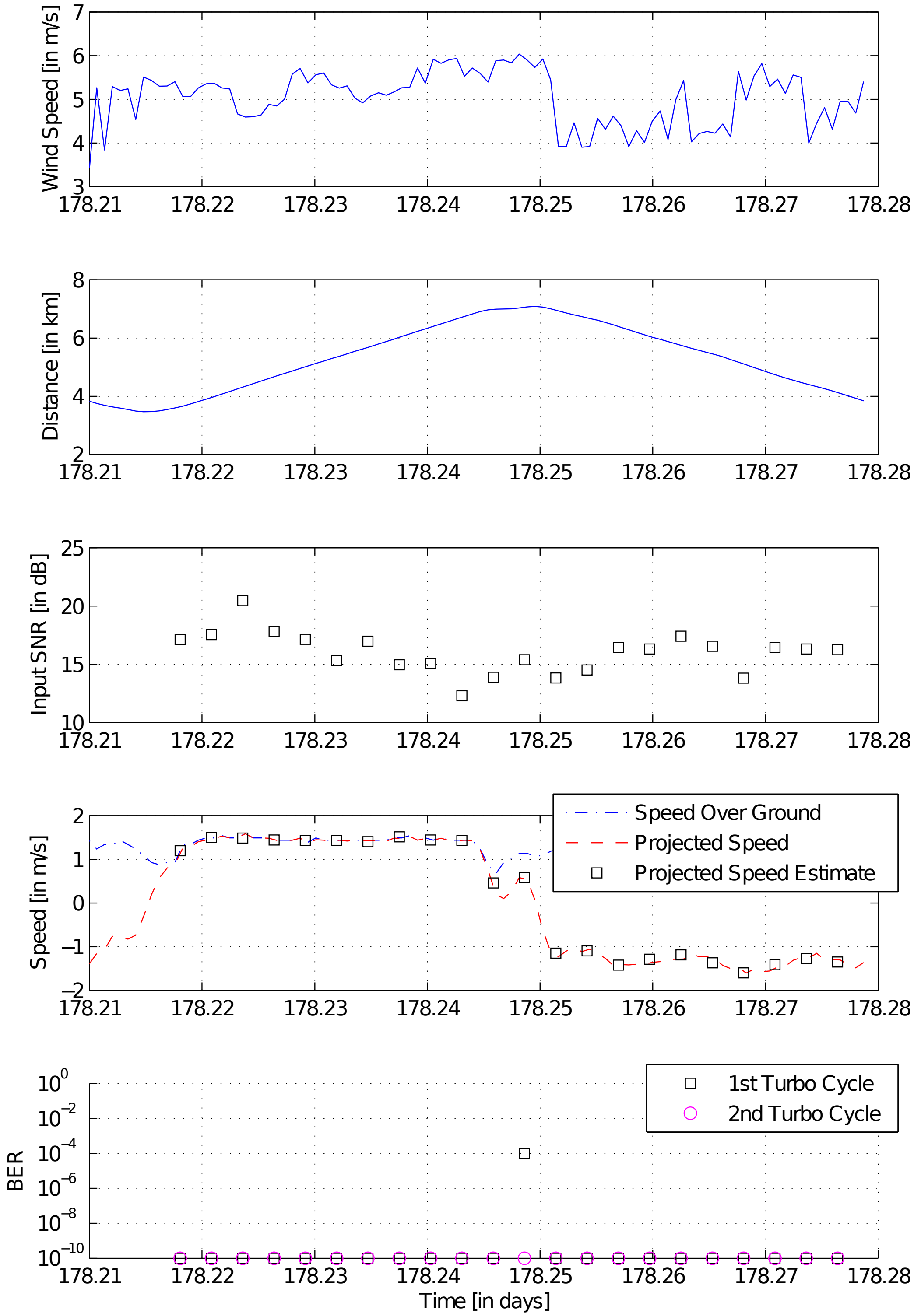

Figure 2.14: MACE10 Evaluation Day 178.

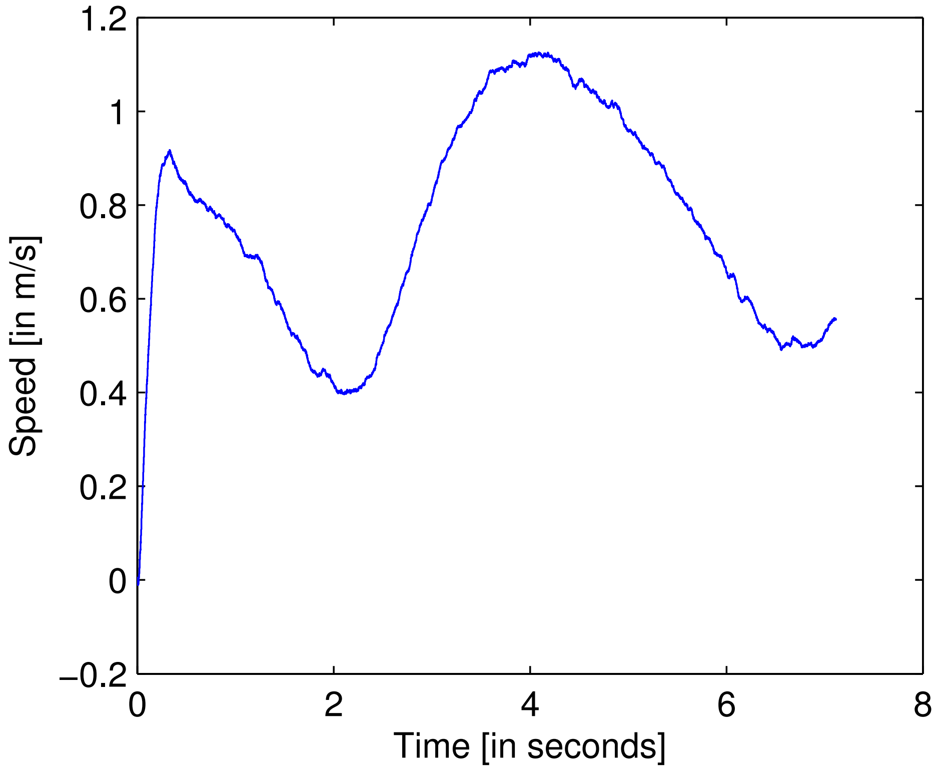

Figure 2.15: Speed as estimated by our Doppler compensator during an example MACE10 transmission.

Table 2.1: Performance of different underwater communication methods in past field-tests.

Team	Data Rate	Range	Speed	Power	Method
LinkQuest	$80bps$	$4km$	$> 1.5m/s$	$48W$	SS
MIT/WHOI	$80bps$	$4km$	$> 1.5m/s$	$50W$	FH-FSK
MIT/WHOI	$2.5kbps$	$1km$	$< 0.05m/s$	$50W$	DFE
MIT/WHOI	$150kbps$	$9m$	$0m/s$	$\sim 10W$	DFE
Our Team	$23.4kbps$	$> 7.2km$	$> 1.5m/s$	$15W$	TRE
Our Team	$39kbps$	$2.7km$	$> 1.5m/s$	$15W$	TRE
Our Team	$1.2Mbps$	$12m$	$> 1.5m/s$	$0.33W$	TRE
Our Team	$100Mbps$	$< 1m$	$0m/s$	$1W$	TRE

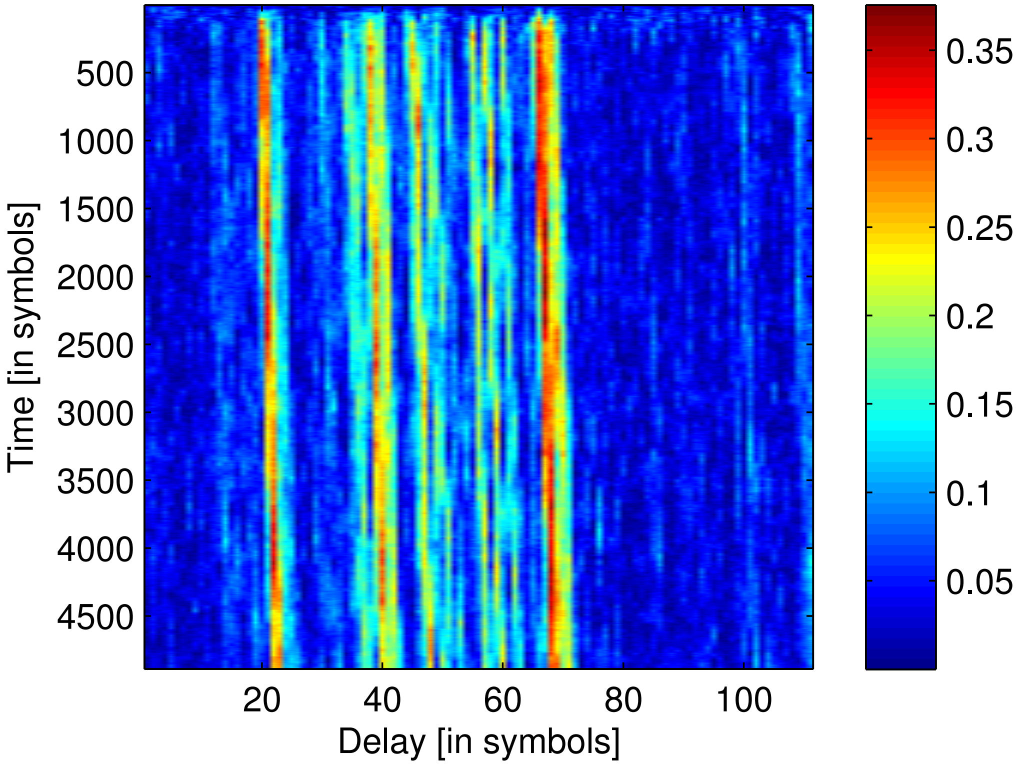

Figure 2.16: Absolute value of channel impulse response as estimated during an example MACE10 transmission.

experiments in [33] show that at a carrier frequency of $15kHz$ this method tolerates phase variations up to about $2rad/s$ which corresponds to a speed of only $0.0318m/s$. Our TRE method is robust to all levels of Doppler that we were able to simulate in laboratory experiments and at-sea tests so far ($> 1.5m/s$) and still reliably obtains the highest data rates ever recorded for acoustic underwater communication. The ultrasound equipment we used for our $100Mbps$ experiment did not allow the transmitter or receiver to move so only the stationary case could be tested.

2.13 Conclusions

Current wireless underwater modems suffer a significant performance degradation when communication platforms are mobile and Doppler effects corrupt the transmitted signals. FSK can be made to be robust to Doppler effects but then uses the available time and frequency resources rather inefficiently and typically only obtains a data rate of $80bps$. Coherent communication has the potential to significantly improve data rate and bandwidth efficiency. Existing approaches, however, only work if the Doppler variation is sufficiently small and roughly constant for the duration of a block. In our work, time-varying Doppler is explicitly modeled, tracked and compensated. We propose to resample the received waveforms non-uniformly and adapt the sampling rate on-the-fly. The resulting signals are then filtered to remove any intersymbol interference caused by time dispersion and multi-path effects. Integrated into an iterative turbo equalization based receiver, this novel resampling equalizer has demonstrated unprecedented communication performance in US Navy sponsored field tests and simulations. We achieved a data rate of $39kbps$ at a distance of $2.7km$ and a data rate of $1.2Mbps$ at a distance of $12m$. The latter link is capable of streaming video in real-time, a first in wireless underwater communication.

CHAPTER 3

FINITE BLOCK-LENGTH ACHIEVABLE RATES FOR QUEUING TIMING CHANNELS

3.1 Introduction

While most communication systems convey information by controlling the amplitudes of signals at each time instant, information can also be sent by controlling the timing at which events occur. For example it is widely believed that neurons exchange information by sending spike trains [87], where information is contained in the random lengths of the interspike intervals. Another example is packet switching networks, where forwarding moves packets from their source toward their ultimate destination. The sources can choose when to send packets, but a queuing mechanism in the forwarding nodes obscures the timing information.

The landmark paper "Bits through Queues" [88] characterizes such channels. Suppose the "packets" are identical and only their arrival time carries information. The times at which the sender puts packets on the network encodes a message as illustrated in Figure 3.1. The packets go through a first-come, first-serve single-server queue with exponential service times. The decoder observes when the packets depart from the queue and then chooses one of the possible messages. For an arrival process constrained to be of rate λ packets per second, it was demonstrated [88] that for an exponential service time distribution of rate $\mu > \lambda$ the capacity $C(\lambda)$ is given by

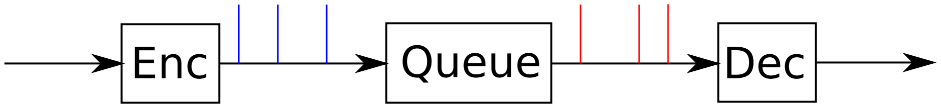

Figure 3.1: Conveying information through packet timings in a queueing system.

$$C(\lambda) = \lambda \log_2 \frac{\mu}{\lambda} \ \text{ nats/s}$$

A point process viewpoint version of the problem with the same fundamental limits was considered in [89, 90]. Rather than considering n inter-arrival times and n inter-departure times of the queue, the time axis was fixed to be $[0, T_n]$ at the encoder and $[0, T_n]$ at the decoder. In [91], Bedekar and Azizoglu considered a discrete time analog to the continuous-time model studied in [88] where packets arrive to and depart from a discrete-time single-server queue with i.i.d. geometrically distributed service times. For an arrival process constrained to be of rate λ packets per time slot, it was demonstrated [91] that for a queue with service times of rate $\mu > \lambda$ the capacity $C(\lambda)$ is given by

$$C(\lambda) = H(\lambda) - \frac{\lambda}{\mu} H(\mu) \ \text{ nats/slot}$$

where $H(\cdot)$ denotes the binary entropy function.

The timing channels with memoryless service times (i.e. exponential in the continuous case and geometric in the discrete case) are known to be the simplest, and in some sense canonical, queuing timing channels. This chapter focuses on the discrete time model with memoryless service times and discusses the maximal achievable rate of communication when there is a practical finite-length restriction on the codewords.

When each codeword corresponds to the timing of packets in n time units and the probability of error may not exceed ϵ, the maximal achievable rate can be substantially less than capacity. By using Markov chain analysis, we prove a lower bound on the maximal channel coding rate achievable at blocklength n and error probability ϵ. We shall show that the maximal channel coding rate is lower bounded by

$$C(\lambda) - n^{-1/2} \sigma Q^{-1}(\epsilon) - \frac{\log n}{2n} + O(n^{-1})$$

where $C(\lambda)$ is the channel capacity whose closed form expression is given above, $Q(\cdot)$ denotes the Q-function and σ^2 is the asymptotic variance of the underlying Markov chain for which we give a closed form expression below. Dropping the last two terms in this expression yields a good approximation which in turn can be used to anticipate the achievable rate on this channel in the finite block length regime.

Asymptotic bounds on the maximal channel coding rate were studied extensively in the 1960s for the case of memoryless channels [92–94]. Wolfowitz introduced hypothesis testing to information theory in [95]. Strassen built on his results in [94], where he combined hypothesis testing arguments (Neyman-Pearson lemma), Feinstein's lemma [96] and bounds on the convergence rate of the central limit theorem [97, 98] to give the strongest result to date. At the time non-asymptotic bounds were constructed in [95, 96, 99]. Recently, this research has been readdressed for memoryless channels [100] and new non-asymptotic bounds were derived in [101]. Strassen's asymptotic expansion gives very accurate estimates when compared to the tightest bounds available [101].

The queuing timing channel considered here has memory and previous results therefore do not apply. We show that the bound provided by the Berry-Esseen theorem in the memoryless channel case still holds and then prove the asymptotic result above by use of Feinstein's lemma. Further, as mentioned before, we obtain a closed form expression for the asymptotic variance σ^2. Finding such an expression for a given Markov chain is generally hard and significant research in the area of steady-state stochastic simulation [102, 103] yields a closed form solution only for the class of homogeneous birth-death processes.

3.2 Basic Definitions and Conventions

- For $x \in [0,1]$, denote $\bar{x} \triangleq 1 - x$.
- Denote Bern(p) to be the Bernoulli distribution with parameter p.
- Denote the binary entropy function $H(p) = -p \log p - \bar{p} \log \bar{p}$.
- X denotes a random variable, $\mathbb{E}[X]$ denotes an expectation, and x denotes a realization.
- $\boldsymbol{x}$ denotes a vector $(x_1, x_2, ..., x_n)$.
- A random process $\varPhi = (\varPhi_1, \varPhi_2, \ldots)$ on a probability space $(\varOmega, \mathcal{F}, P)$ is

a *Markov process* if for any n,

$$\begin{aligned} & P(\mathit{\Phi}_1 \in A_1, \mathit{\Phi}_2 \in A_2, \ldots, \mathit{\Phi}_n \in A_n) \\ = \; & P(\mathit{\Phi}_1 \in A_1) \prod_{i=2}^{n} P(\mathit{\Phi}_i \in A_i | \mathit{\Phi}_{i-1} \in A_{i-1}). \end{aligned}$$

- $\mathbb{Z}$ is the set of all integers and $\mathbb{Z}_+ = \{z \in \mathbb{Z} : z \geq 0\}$.

- Denote $[n]_j = \{j, \ldots, n\}$ with $[n] \equiv [n]_1$.

- Denote X^n to be the sequence of counting functions on $[n]_0$, i.e. the set of functions $x_0, \ldots, x_n$ for which $x_i \in \mathbb{Z}_+$ and $x_i \geq x_{i-1}$. Denote Y^n to be the set of counting functions y on $[n]_0$ for which $y_0 = 0$.

- For a sequence of input sets and output sets $\{\mathsf{X}^n, \mathsf{Y}^n : n \geq 1\}$, a channel is a a sequence of conditional distributions $\{P_{Y^n|X^n}(\cdot|\boldsymbol{x}^n) : x^n \in \mathsf{X}^n, n \geq 1\}$.

- Given a distribution P_{X^n} on X^n and channel $P_{Y^n|X^n}(\cdot|\boldsymbol{x}^n)$, denote P_{Y^n} as the induced output distribution.

- Denote the information density as $i(\boldsymbol{x}^n, \boldsymbol{y}^n) \triangleq \log \frac{P_{Y^n|X^n}(\boldsymbol{y}^n|\boldsymbol{x}^n)}{P_{Y^n}(\boldsymbol{y}^n)}$.

- For any $\epsilon \in (0,1)$, an (M, n, ϵ) code is a sequence $\{(\boldsymbol{x}^{(i)}, \boldsymbol{D}^{(i)}), i = 1, \ldots, M\}$ where $\boldsymbol{x}^{(i)} \in \mathsf{X}^n$ and $\{\boldsymbol{D}^{(i)}\}$ are mutually disjoint with $P(\boldsymbol{D}^{(i)}|\boldsymbol{x}^{(i)}) > 1 - \epsilon \; \forall i$.

- The rate of an (M, n, ϵ) code is denoted by $R = \frac{\log M}{n}$.

- Borrowing notation from [94, 95], $N(\epsilon, n, \lambda)$ denotes the supremum of the integers M such that an (M, n, ϵ)-code exists and $\mathbb{E}[X_n/n] = \lambda$.

- Denote the rate-constrained capacity as $C(\lambda) \triangleq \lim_{\epsilon \to 0} \lim_{n \to \infty} \frac{\log N(\epsilon, n, \lambda)}{n}$.

- We drop subscripts whenever they are clear from the context. For example $P_{Y^n|X^n}(\boldsymbol{y}^n|\boldsymbol{x}^n) = P(\boldsymbol{y}^n|\boldsymbol{x}^n)$.

3.3 System Description and Preliminaries

Throughout this document, we consider a discrete-time point process version of the problem, analogous to [89, 90]. The communication channel we consider is an interesting example of a channel with memory. It is essentially a probabilistic single server queuing system with the length of the queue being the memory of channel. At each discrete time instance i, the random variable $\tilde{X}_i$, $i \in [n-1]$, indicates if there was an arrival at the back of the queue at time i, and $\tilde{Y}_i$ indicates if there was a departure from the front of the queue at time i. Further, X_i (Y_i) counts the total number of arrivals (departures), Q_i denotes the length of the queue at time i, and the initial length of the queue $Q_0 \equiv X_0$ is a non-negative integer-valued random variable with distribution $P_{Q_0} = P_{X_0}$. Then note that we have

$$X_i = Q_0 + \sum_{l=1}^{i} \tilde{X}_l \tag{3.1}$$

$$Y_i = 0 + \sum_{l=1}^{i} \tilde{Y}_l \tag{3.2}$$

$$Q_i = Q_0 + X_i - Y_{i-1} = Q_{i-1} + \tilde{X}_i - \tilde{Y}_{i-1} \tag{3.3}$$

This is illustrated in Figure 3.2. Note that there is a bijection between $(Q_0, \tilde{X}_1, \ldots, \tilde{X}_n)$ and the channel input, $X^n \triangleq (X_0, X_1, \ldots, X_n)$. For geometrically distributed service times, the binary random variables $\tilde{Y}_i$ are conditionally independent given Q_i and are distributed according to the conditional law pertaining to a Z channel

$$P_{\tilde{Y}_i|Q_i}(\tilde{Y}_i|Q_i) = \begin{cases} 1; & \tilde{Y}_i = 0, Q_i = 0 \\ \bar{\mu}; & \tilde{Y}_i = 0, Q_i > 0 \\ 0; & \tilde{Y}_i = 1, Q_i = 0 \\ \mu; & \tilde{Y}_i = 1, Q_i > 0 \end{cases} \tag{3.4}$$

The vector $\boldsymbol{Y}^n \in \mathsf{Y}^n$ is the channel output vector and there is a bijection between $\boldsymbol{Y}^n \in \mathsf{Y}^n$ and $(\tilde{Y}_i : i \in [n])$. With this the channel law reads

$$P(\boldsymbol{y}^n|\boldsymbol{x}^n) = \prod_{i=0}^{n-1} P(\tilde{y}_i|q_i) \tag{3.5}$$

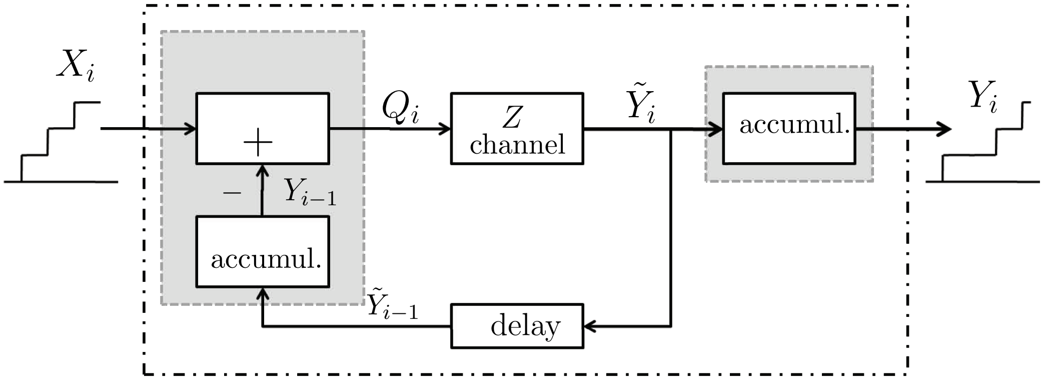

Figure 3.2: A simple time-invariant description of the queuing timing channel.

We assume the queue to be stable and hence the arrival rate $\lambda = \frac{\mathbb{E}[X_n]}{n}$ to be smaller than the serving rate μ. We now state the following theorem:

Theorem 7. *[91]: For the queueing timing channel of rate μ given by* (3.5),

$$C(\lambda) = H(\lambda) - \frac{\lambda}{\mu} H(\mu) \tag{3.6}$$

The optimal $P^*_{X^n}$ is given by $\tilde{X}_i$ drawn i.i.d. with Bern(λ) distribution and Q_0 independently drawn with π_Q given by

$$\pi_Q(q) = \begin{cases} \frac{\bar{\lambda}\mu - \lambda\bar{\mu}}{\mu}; q = 0 \\ \frac{\bar{\lambda}\mu - \lambda\bar{\mu}}{\bar{\mu}\mu}\rho^q; q > 0 \end{cases} \tag{3.7}$$

where $\rho \triangleq \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}$.

Note that $\rho < 1$ if and only if $\lambda < \mu$. We will also exploit how under $P^*_{X^n}$, the output $\tilde{Y}_i$'s are i.i.d.:

Theorem 8. (Burke's Theorem) *For any n, for the channel given by* (3.5) *and $P_{X^n} = P^*_{X^n}$ given in Theorem 7, the outputs $\tilde{Y}_i$'s are i.i.d. with Bern(λ) distribution.*

Proof. The proof is similar to the one for continuous time queues and can be found in [104]. □

Throughout the remainder of this chapter, we assume that $P_{X^n} = P^*_{X^n}$.

By the above theorem

$$P(\boldsymbol{y}^n) = \prod_{i=0}^{n-1} P(\tilde{y}_i) \tag{3.8}$$

It is also well-known from Burke's theorem that under $P^*_{X^n}$, the $(Q_i : i \geq 0)$ form a Markov chain and likewise for the random process $\Big((Q_i, \tilde{Y}_i) : i \geq 0\Big)$. The transition probabilities for the Markov chain pertaining to $(Q_i : i \geq 0)$ are given by

$$P_{Q_{i+1}|Q_i}(q_{i+1}|q_i) = \begin{cases} \lambda; & q_{i+1} = q_i + 1, q_i = 0 \\ \bar{\lambda}; & q_{i+1} = q_i, q_i = 0 \\ \bar{\lambda}\mu; & q_{i+1} = q_i - 1, q_i > 0 \\ \lambda\bar{\mu}; & q_{i+1} = q_i + 1, q_i > 0 \\ 1 - \lambda\bar{\mu} - \bar{\lambda}\mu; & q_{i+1} = q_i, q_i > 0 \end{cases} \tag{3.9}$$

If and only if $\lambda < \mu$, there exists a probability measure π_Q on $\mathbb{N}_0$ that solves the system of equations

$$\sum_{q_i \in \mathbb{N}_0} \pi_Q(q_i) P_{Q_{i+1}|Q_i}(q_{i+1}|q_i) = \pi_Q(q_{i+1}) \tag{3.10}$$

for all $q_{i+1} \in \mathbb{N}_0$ and this measure is called the invariant measure. Note that for irreducible Markov chains the existence of such a probability measure is equivalent to positive recurrence. For the transition probabilities given it can be checked that $\pi_Q(q_i)$ as defined in Theorem 7 is the solution.

The following lemma uses arguments introduced by Feinstein [96] to give a lower bound on $N(\epsilon, n, \lambda)$.

Lemma 2. (Feinstein) *For any distribution P_{X^n} and any $\theta \in \mathbb{R}$ there exists an (M, n, ϵ) code such that*

$$M \geq e^{\theta} \left\{\epsilon - P(i(\boldsymbol{x}^n, \boldsymbol{y}^n) \leq \theta)\right\} \tag{3.11}$$

3.4 Finite-Length Scaling

Recall that by (3.5) and (3.8) the distributions $P(\boldsymbol{y}^n|\boldsymbol{x}^n)$ and $P(\boldsymbol{y}^n)$ factor and hence

$$i(\boldsymbol{x}^n, \boldsymbol{y}^n) = \sum_{i=0}^{n-1} \log \frac{P(\tilde{y}_i|q_i)}{P(\tilde{y}_i)} = \sum_{i=0}^{n-1} f(\tilde{y}_i, q_i) \tag{3.12}$$

where

$$f(\tilde{y}_i, q_i) \triangleq \log \frac{P(\tilde{y}_i|q_i)}{P(\tilde{y}_i)} \tag{3.13}$$

The composed state $\psi_i = (\tilde{y}_i, q_i)$ again forms a positive recurrent Markov chain whose transition probabilities are illustrated in Figure 3.3. The invariant measure π_Ψ for this chain is only a slight extension to π_Q:

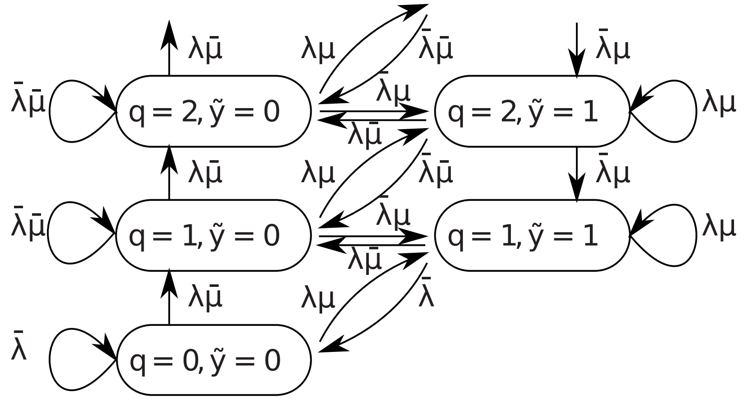

Figure 3.3: Possible transitions in the Markov chain $(\tilde{Y}, Q)$.

$$\pi_\Psi(\tilde{y}, q) = P_{\tilde{Y}_i|Q_i}(\tilde{y}|q)\pi_Q(q) \tag{3.14}$$

The proof of the following theorem is one of the main contributions of this chapter because it can be used to proof an asymptotic expansion of the quantity $N(\epsilon, n, \lambda)$.

Theorem 9. *The asymptotic variance*

$$\sigma^2 = \lim_{n\to\infty} \frac{1}{n} \operatorname{Var}(i(\boldsymbol{x}^n, \boldsymbol{y}^n)) \tag{3.15}$$

is well defined, positive and finite, and

$$\sigma^2 = \mathrm{Var}(f(\Psi_0)) + 2\sum_{i=1}^{\infty} \mathrm{Cov}(f(\Psi_0), f(\Psi_i)) \tag{3.16}$$

Further the following Berry-Esseen type bound holds:

$$\sup_{\xi\in\mathbb{R}} \left| P\left(\frac{i(\boldsymbol{x}^n, \boldsymbol{y}^n) - nC(\lambda)}{\sigma\sqrt{n}} \le \xi \right) - \Phi(\xi) \right| \le \mathcal{O}(n^{-1/2}) \tag{3.17}$$

Proof. A detailed proof can be found in the appendix. We only give a sketch here. The Markov chain Ψ is aperiodic and irreducible. The state space of Ψ_i can be chosen to be $\mathbb{X} = \{0,1\} \times \mathbb{N} \cup \{(0,0)\}$. First we verify that there exists a Lyapunov function $V : \mathbb{X} \to (0, \infty]$, finite at some $\psi_0 \in \mathbb{X}$, a finite set $\mathbb{S} \subset \mathbb{X}$, and $b < \infty$ such that

$$\mathbb{E}[V(\Psi_{i+1}) - V(\Psi_i) | \Psi_i = \psi] \le -1 + b\mathbf{1}_{\mathbb{S}}(\psi), \quad \psi \in \mathbb{X} \tag{3.18}$$

The chain is skip-free and the found Lyapunov function is linear and hence also Lipschitz. These properties imply that the chain is geometric ergodic [105, 106] and the bound in (3.17) hence holds by arguments made in [107]. □

Another approach towards proofing Berry-Esseen type bounds for Markov chains is to verify a mixing condition but the resulting bounds are weaker [108].

Remark 2. *An explicit solution to the asymptotic variance of a general irreducible positive recurrent Markov chain is not available.*

Significant research in the area of steady-state stochastic simulation has focused on obtaining an expression for the asymptotic variance [102, 103] and has yielded a closed form solution only for the class of homogeneous birth-death processes when $f(\psi_i)$ simply returns the integer valued state itself.

We build upon an idea introduced in [109] to give an explicit closed form solution to the asymptotic variance in (3.15).

Theorem 10. *The asymptotic variance defined in* (3.15) *has a closed form* *solution:*

$$\sigma^2 = -\operatorname{Var}(f(\mathit{\Psi}_0)) + 2\sum_{i=0}^{\infty} \operatorname{Cov}(f(\mathit{\Psi}_0), f(\mathit{\Psi}_i)) \tag{3.19}$$

where

$$\begin{aligned}
\operatorname{Var}(f(\mathit{\Psi}_0)) =& \log^2(\frac{1}{\bar{\lambda}})\pi_Q(0) + \log^2(\frac{\mu}{\lambda})\mu\overline{\pi_Q(0)} \\
&+ \log^2(\frac{\bar{\mu}}{\bar{\lambda}})\bar{\mu}\overline{\pi_Q(0)} - C^2 \qquad (3.20) \\
\sum_{i=0}^{\infty} \operatorname{Cov}(f(\mathit{\Psi}_0), f(\mathit{\Psi}_i)) =& \log\frac{1}{\bar{\lambda}}(-c_{\tilde{M}}\frac{\rho}{1-\rho} - c_{M0}\rho) \\
&+ \log\frac{\mu}{\lambda}c_{M0}\frac{\rho}{1-\rho} + \log\frac{\bar{\mu}}{\bar{\lambda}}\frac{\rho}{1-\rho}(c_{\tilde{M}} - \rho c_{M0}) \qquad (3.21)
\end{aligned}$$

and we define

$$\begin{aligned}
c_{M0} &= \frac{\bar{\lambda}}{\bar{\mu}}\left(\mu\log(\frac{\mu}{\lambda}) + \bar{\mu}\log(\frac{\bar{\mu}}{\bar{\lambda}}) - C\right) && (3.22) \\
c_{\tilde{M}} &= \left\{\frac{c_{M0}}{\mu} + (C - \log\frac{\mu}{\lambda})\frac{\pi_Q(0)}{\bar{\mu}}\right\} && (3.23)
\end{aligned}$$

Proof. Again we only sketch the proof here and refer to the appendix for a detailed version. For the computation of the sum $\sum_{i=0}^{\infty}\operatorname{Cov}(f(\mathit{\Psi}_0), f(\mathit{\Psi}_i))$ we will setup and solve a recursion.

We define

$$r(\psi, i) = \sum_{\psi' \in \mathbb{X}} (f(\psi') - C)\pi_{\mathit{\Psi}}(\psi')p_{\mathit{\Psi}_i|\mathit{\Psi}_0}(\psi|\psi') \tag{3.24}$$

Clearly

$$r(\psi, 0) = (f(\psi) - C)\pi_{\mathit{\Psi}}(\psi) \tag{3.25}$$

and

$$\begin{aligned}
\operatorname{Cov}(f(\mathit{\Psi}_0), f(\mathit{\Psi}_i)) &= \sum_{\psi\in\mathbb{X}} (f(\psi) - C)r(\psi, i) && (3.26) \\
&= \sum_{\psi\in\mathbb{X}} f(\psi)r(\psi, i) && (3.27)
\end{aligned}$$

Note, however, that for the computation of the asymptotic variance we actually do not even need to know this covariance for each i. It is sufficient to know its sum. So we define

$$R(\psi) = \sum_{i=0}^{\infty} r(\psi, i) \tag{3.28}$$

exchange limits

$$\sum_{i=0}^{\infty} \mathrm{Cov}(f(\Psi_0), f(\Psi_i)) = \sum_{\psi \in \mathbb{X}} f(\psi) R(\psi) \tag{3.29}$$

and derive and solve a recursion for the sequence $R(\psi)$. □

Using the result stated in Theorem 9 we can prove the final contribution of this chapter:

Theorem 11.

$$\log N(n, \epsilon, \lambda) \geq nC(\lambda) - \sqrt{n}\sigma Q^{-1}(\epsilon) - \frac{1}{2}\log n + O(1) \tag{3.30}$$

where $C(\lambda)$ is given by (3.6) and σ is defined as in Theorem 9.

Proof. By Theorem 9 $\exists A > 0$:

$$|P((i(\boldsymbol{x}, \boldsymbol{y}) - nC)/\sqrt{n\sigma^2} \leq \xi_1) - \Phi(\xi_1)| \leq \frac{A}{\sqrt{n}} \; \forall \xi_1 \in \mathbb{R} \tag{3.31}$$

Let $B > A$ and $\xi_1 = \Phi^{-1}(\epsilon - \frac{B}{\sqrt{n}}) < \Phi^{-1}(\epsilon) = \xi_0$. Set $\theta = \sqrt{n}\sigma\xi_1 + nC$ and the application of Feinstein's Lemma [96, 110] yields

$$\begin{aligned} &\log N(n, \epsilon, \lambda) - nC - \sqrt{n}\sigma\xi_0 \\ \geq &\log\left(\epsilon - P\left(\frac{i(\boldsymbol{x}, \boldsymbol{y}) - nC}{\sqrt{n}\sigma} \leq \xi_1\right)\right) + \sqrt{n}\sigma(\xi_1 - \xi_0) \qquad (3.32) \\ \geq &\log\left(\epsilon - \Phi(\xi_1) - \frac{A}{\sqrt{n}}\right) + \sqrt{n}\sigma O(\frac{1}{\sqrt{n}}) \qquad (3.33) \end{aligned}$$

□

Remark 3. *We believe that the above result can be strengthened by dropping the term $\frac{1}{2}\log n$ from the right hand side of the inequality. By use of hypothesis testing arguments Strassen [94] was able to prove the above bound without* *the $\frac{1}{2}\log n$ term for the class of discrete memoryless channels. The used arguments can probably be extended to hold for the non-memoryless channel considered here as well since the information density factors and by Theorem 9 a Berry-Esseen type bound holds.*

Theorem 11 confirms that $C(\lambda)$ is the operational capacity of the channel and any rate $R < C(\lambda)$ is achievable. For illustration we plotted the approximation

$$C(\lambda) - n^{-1/2}\sigma Q^{-1}(\epsilon) \tag{3.34}$$

to the achievable coding rate for blocklengths ranging between 50 and 3000, various values for ϵ and the example values $\lambda = 0.2$, $\mu = 0.8$ in Figure 3.4.

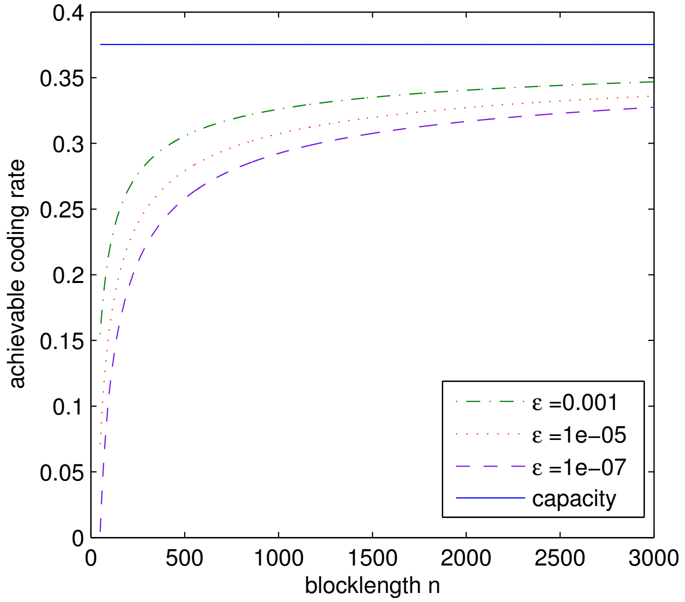

Figure 3.4: Channel coding rate in the finite block-length regime.

CHAPTER 4

CAN YOU HEAR MY VOICE NOW? IP OVER VOICE-OVER-IP FOR CENSORSHIP CIRCUMVENTION

4.1 Introduction

The Internet is playing an ever-increasing role in connecting people from across the world, facilitating the free circulation of speech, ideas and information. This poses serious threats to repressive regimes as it elevates their citizens' awareness and provides them a powerful medium to arrange coordinated opposition movements. The recent unrest in the Middle East [111] demonstrates the very strong power of the Internet in arranging nation-wide protests that, in several cases, resulted in revolutionizing or even overthrowing repressive regimes. In response to such threats, repressive regimes make use of different technologies to restrict and monitor their citizens' access to the Internet; i.e., they *censor* the Internet. Censorship devices leverage various techniques [112, 113] ranging from simple IP address blocking and DNS hijacking to the more complicated and resource-intensive deep packet inspection (DPI) in order to enforce their blocking and monitoring. Citizens identified as non-complying with the censors' restrictions can face different consequences ranging from Internet service disruption to severe life-threatening punishments [114].

To help censored users gain open access to the Internet, different systems and technologies have been designed and developed [115–121], generally referred to as *censorship circumvention* tools. These systems are composed of computer and networking technologies that allow Internet users to evade monitoring, blocking, and tracing of their activities. We observe that *the biggest challenge facing the existing circumvention systems is the lack of "unobservability"*: while these systems can, under certain conditions, circumvent censorship they are not effectively able to hide the fact that their users are making use of them [115–119]. For instance, the Tor [118] anonymity network is not able to effectively evade censorship as a censor can block all of the publicly advertised IP addresses of Tor relays. This has two major consequences: first, users caught (by censors) leveraging these circumvention systems may face various punishments such as imprisoning. Second, and even more catastrophic, this lack of unobservability usually leads to the lack of *availability*; i.e., circumvention systems with observable communication are easily blocked by censors. Censors proactively [122] look for Internet services that help with censorship circumvention and either block any access to them by their citizens, or leave them (partially) open to identify their users. In particular, censors rigorously look for IP addresses belonging to circumvention technologies (e.g., HTTP/SOCKS proxies) and add them to the IP blacklists maintained by their censoring firewalls [112, 123]. Consequently, citizens under repressive regimes often find it difficult to access the existing circumvention systems. For instance, the popular Tor network has frequently been/is blocked by several repressive regimes [122, 124].

To provide unobservable circumvention, different approaches have been taken by the research community. Several systems [115, 117, 125] provide unobservability by *pre-sharing secrets* with their intended clients. The Tor system, for instance, has recently deployed Tor bridges [125], which are volunteer proxies whose IP addresses are distributed among Tor users in a selective manner. This makes Tor bridges less prone to be identified by censors, as compared to the publicly-advertised Tor entry nodes; however, there are serious challenges in distributing their IP addresses among users [126, 127]. In a similar manner, Infranet [115] and Collage [117] aim for unobservability by pre-sharing some secret information with their users. This, however, is neither scalable nor effective as it is challenging to share secrets with a large number of real users, while keeping them secret from censors at the same time [128–130].

As another approach to provide unobservability, several systems use various *obfuscation* techniques. For instance, Ultrasurf [131] and Psiphon [132] try to confuse content filtering tools by obfuscating their design and traffic patterns. Such obfuscation, however, jeopardizes users' security, as analyzed in a recent study [133]. Appelbaum et al. propose *pluggable transports* [134] for Tor, a platform that allows one to build protocol-level obfuscation plugins for Tor traffic. These plugins obfuscate a Tor client's traffic to Tor bridges by shaping it to look like another protocol that is allowed by censors. Obfsproxy [135] is the first Tor pluggable transport. It adds an additional layer of encryption to Tor traffic to obfuscate Tor's content identifiers, like the TLS parameters; however, it does not remove Tor's statistical patterns like packet timings and sizes. Murdoch et al. [136] mention several weaknesses for obfsproxy, including being susceptible to either an active or passive attacker who has recorded the initial key exchange. StegoTorus [137] provides better unblockability, but comes with a much higher overhead [136]. SkypeMorph [138] morphs Tor traffic into Skype video calls in order to make it undetectable against deep-packet inspection and statistical analysis. The common issue with the aforementioned traffic obfuscation techniques is that they only obfuscate communication patterns, but not the end-hosts. In other words, while a censor may find it hard to detect the obfuscated traffic using traffic analysis, it will be able to identify the end-hosts that obfuscate the traffic through other active/passive attacks, e.g., SkypeMorph and StegoTorus relays can be enumerated using prevalent port knocking techniques [122,139], zig-zag [140] attack, and insider attack [141]. Once the identity of a circumventing end-host is known to a censor, the unobservability is completely lost and the end-host is easily blocked by the censor. CensorSpoofer [141] is another recent proposal that performs traffic obfuscation by mimicking VoIP traffic. Like most of the other designs noted above, CensorSpoofer needs to pre-share some secret information with the clients, posing a scalability challenge. In addition, it requires a usable upstream channel for its operation since its circumvented traffic is unidirectional.

As another recent trend, several proposals have sought unobservability by integrating circumvention into the Internet infrastructure [120, 121]. For instance Telex [120] and Cirripede [121] conceal the circumvented traffic inside the regular HTTPS traffic thanks to friendly ISPs that deflect/manipulate the intercepted connections. The real-world deployment of such circumvention systems requires collaboration of several trusted ISPs that make software and/or hardware modifications to their infrastructure; this does not seem to be realized in short-time until there are enough financial/political motives for the ISPs. Moreover, a recent study [142] shows that an adversary capable of changing routing decisions is able to block these systems.

In this chapter we propose *FreeWave*, a censorship circumvention infrastructure that is highly unobservable (hence, highly available). The main idea of FreeWave, as shown in Figure 4.1, is to tunnel Internet traffic inside

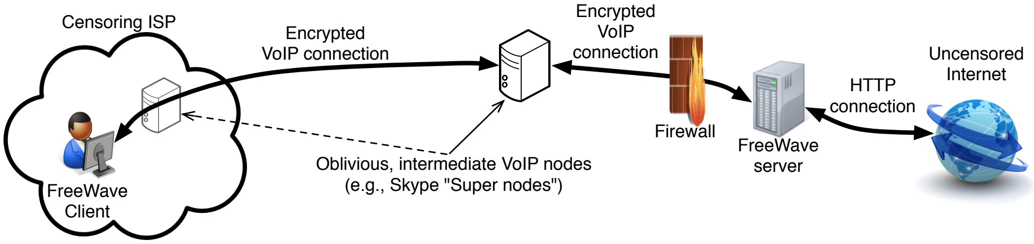
Censoring ISP
Encrypted VoIP connection
Encrypted VoIP connection
Uncensored Internet
HTTP connection
Firewall
FreeWave server
Oblivious, intermediate VoIP nodes (e.g., Skype "Super nodes")
FreeWave Client

Figure 4.1: The main architecture of FreeWave.

non-blocked VoIP communications by modulating them into acoustic signals that are carried over VoIP connections. For a censored user to use FreeWave for circumvention, she needs to setup a VoIP account with a public VoIP provider, and also to install FreeWave's client software on her machine. Part of the FreeWave system is a FreeWave server that listens on several *public*ly advertised VoIP IDs to serve FreeWave clients. To make a FreeWave connection, a user's FreeWave client software makes VoIP connections to FreeWave server's VoIP IDs. The client and server, then, tunnel the circumvented Internet traffic inside the established VoIP connections, by modulating network packets into acoustic signals carried by the established VoIP connections.

We claim that FreeWave provides strong unobservability by performing two kinds of obfuscations: *traffic obfuscation* and *server obfuscation*. First, as FreeWave tunnels Internet traffic inside *actual*, encrypted VoIP connections, its traffic patterns are very hard to distinguish from benign VoIP connections. Traffic obfuscation is also aimed for by recent morphing-based techniques like SkypeMorph [138] and StegoTorus [137]; however, FreeWave provides stronger traffic obfuscation as it completely runs the target protocol instead of partially imitating it. The second obfuscation performed by FreeWave, which is *unique* to FreeWave, is *server obfuscation*, which prevents censors from detecting circumvented traffic by matching the destination addresses of traffic. Server obfuscation is an important feature that similar circumvention systems such as SkypeMorph [138] and StegoTorus [137] fail to provide. As we describe later in this chapter, the way the FreeWave server is connected to the Internet results in getting FreeWave's VoIP traffic relayed by various, *oblivious* VoIP peers, preventing a censor from blocking/identifying FreeWave's VoIP traffic based on IP addresses (see Figure 4.1). For instance, FreeWave connections made through Skype get relayed by Skype *supern-*

odes [143], which are oblivious Skype users residing *outside*[1] *the censorship region.* As another example, if FreeWave uses Google Voice, FreeWave connections will get relayed by Google servers that are oblivious to the circumvention process. Server obfuscation, as defined above, is missing in *all* previous designs except CensorSpoofer [141]. For instance, in the case of Tor pluggable transports like SkypeMorph [138] and StegoTorus [137], once the IP address of the deploying Tor bridge is revealed to a censor (e.g., using port knocking [122, 126, 127, 139]), the unobservability is lost and the censor will be able to identify/block users connecting to that Tor bridge. In FreeWave, on the other hand, *even if a censor identifies the IP address belonging to a FreeWave server it will not be able to block connections to it* since users' connections to that FreeWave server are not direct connections, but are relayed through varying, oblivious VoIP nodes. We provide a thorough comparison of FreeWave with similar obfuscation-based techniques in Section 4.9.

The strong unobservability of FreeWave makes it highly unblockable (i.e., available). FreeWave's availability is tied to the availability of the VoIP service: Since *the operation of FreeWave is not bound to a specific VoIP provider*, in order to block FreeWave a censor needs to block *all* VoIP connections with the outside world. This is not desirable by the censoring ISPs due to different business and political implications. VoIP constitutes an important part of today's Internet communications [146–148]; a recent report [147] shows that about one-third of U.S. businesses use VoIP solutions to reduce their telecommunications expenses, and the report predicts the VoIP penetration to reach 79% by 2013, a 50% increase compared to 2009.

We implement a prototype of FreeWave over the popular VoIP service of Skype and measure its performance. To achieve reliable communication over VoIP connections we design a communication encoder/decoder tailored for the VoIP's lossy communication channel. Specifically, we take advantage of Turbo codes and QAM modulation techniques [84, 149] in order to reliably encode the circumvented traffic inside the VoIP connections. Our evaluations show that FreeWave provides connection bit rates that are suitable for regular web browsing. We validate FreeWave's usability by clients that are

[1] The supernodes assigned to a particular Skype client by the Skype protocol are geographically close to that client for better quality of service; hence a FreeWave server is expected to use nearby supernodes. In addition, a FreeWave server can adjust the list of its Skype supernodes [144, 145], as described later.

geographically far away from the FreeWave server.

Contributions: In this chapter we make the following main contributions:

1. We propose FreeWave, a novel infrastructure for censorship circumvention that works by modulating Internet traffic into the acoustic signals carried over VoIP connections. The use of actual VoIP connections, as well as being relayed by oblivious VoIP nodes, provides promising unobservability for FreeWave.

2. We design communication encoders and decoders to efficiently modulate Internet traffic into acoustic signals.

3. We prototype FreeWave on the popular VoIP service of Skype and evaluate its performance and security.

The rest of this chapter is organized as follows: In Section 4.2 we review our threat model and the goals in designing our circumvention system. We describe the design of our proposed circumvention system, FreeWave, in Section 4.3, and Section 4.4 discusses our design details. In Section 4.5, we discuss the features of our designed circumvention system. We thoroughly analyze the security of FreeWave in Section 4.6. In Section 4.7 we describe the design of MoDem, the communication block of FreeWave software. We describe our prototype implementation in Section 4.8 along with the evaluation results. In Section 4.9 we compare FreeWave with two recent proposals of SkypeMorph [138] and CensorSpoofer [141]; this is followed by additional related work in Section 4.10. In Section 4.11 we discuss FreeWave's limitations and several recommendations. Finally, the chapter is concluded in Section 4.12.

4.2 Preliminaries

4.2.1 Threat Model

We assume that a FreeWave client is connected to the Internet through a censoring ISP, e.g., an ISP that is controlled and regulated by a repressive regime. Based on the regulations of the censoring ISP its users are not allowed to connect to certain Internet destinations, called the *censored destinations*. The users are also prohibited from using censorship circumvention technologies that would help them to evade the censoring regulations. The censoring ISP uses a set of advanced technologies to enforce its censoring regulations, including IP address blocking, DNS hijacking, and deep packet inspection [112,113]. The censoring ISP also monitors its users' network traffic to identify and block any usage of censorship circumvention tools; traffic analysis can be used by the censor as a powerful technique for this purpose.

We assume that the censoring ISP enforces its regulations such that it does not compromise the *usability* of the Internet for its users, due to different political and economic reasons. In other words, the enforced censorship does not disable/disrupt key Internet services. In particular, we consider VoIP as a key Internet service in today's Internet [146, 148, 150], and we assume that, even though a censor may block certain VoIP providers, the censor will not block *all* VoIP services. VoIP constitutes a key part in the design of FreeWave.

4.2.2 Design Goals

We consider the following goals in the design and evaluation of FreeWave. Later in Section 4.5, we discuss these features for the FreeWave circumvention system proposed in this chapter and compare FreeWave with related work.

Unblockability: The main goal of a censorship circumvention system is to help censored users gain access to censored Internet destinations. As a result, the most trivial property of a circumvention system is being accessible by censored users, i.e., it should be unblockable by censors.

Unobservability: Unobservability is to hide users' utilization of a circumvention system from censorship authorities, which is a challenging feature to achieve due to the recent advances in censorship technologies [112]. The importance of unobservability is two-fold; first, an observable circumvention can jeopardize the safety of a user who has been caught by the censor while using the circumvention system. Second, a weak unobservability commonly results in a weak unblockability, as it allows censors to more easily identify, hence block, traffic generated by the circumvention system.

Security: Several security considerations should be made once analyzing a circumvention system. These considerations include users' anonymity, confidentiality, and privacy against various parties including the censors, the circumvention system, and third parties.

Deployment feasibility: An important feature of a circumvention system is the amount of resources (e.g., hardware, network bandwidth, etc.) required for it to be deployed in the real world. A circumvention system is also desired to have few dependencies on other systems and entities in order to make it more reliable, secure, and cost-effective.

Quality of service: A key feature in making a circumvention system popular in practice is the quality of service provided by it in establishing circumvented connections. Two important factors are connection bandwidth and browsing latency.

4.3 FreeWave Scheme

In this section, we describe the design of FreeWave censorship circumvention. Figure 4.1 shows the main architecture of FreeWave. In order to get connected through FreeWave, a user installs a *FreeWave client* on her machine, which can be obtained from an out-of-band channel, similar to other circumvention systems. The user sets up the installed FreeWave client by entering her own VoIP ID and also the publicly advertised VoIP ID of FreeWave server. Once the FreeWave client starts up, it makes a VoIP audio/video call to FreeWave server's VoIP ID. As discussed in Section 4.4.2, the FreeWave server is configured such that VoIP connections initiated by clients are relayed through various *oblivious VoIP peer*s, e.g., Skype supernodes; this is a key security feature of FreeWave as it prevents a censor from blocking FreeWave's VoIP connections using IP address blocking. Also, since FreeWave's VoIP connections are end-to-end encrypted, a censor will not be able to identify FreeWave's VoIP connections by analyzing traffic contents, e.g., by looking for the VoIP IDs. Using the established VoIP connection, a FreeWave client circumvents censorship by modulating its user's Internet traffic into acoustic signals that are carried over by such VoIP connections. FreeWave server demodulates a client's Internet traffic from the received acoustic signals, and proxies the demodulated traffic to the requested Internet destinations.

Next, we introduce the main components used in FreeWave and describe how these components are used in the design of FreeWave's client and server.

4.3.1 Components of FreeWave

In this section, we introduce the main elements used in the design of FreeWave client and server software. The first three components are used by both FreeWave client and FreeWave server, while the fourth element is only used by FreeWave server.

VoIP client VoIP client is a Voice-over-IP (VoIP) client software that allows VoIP users to connect to one (or more) specific VoIP service(s). In Section 4.4.2, we discuss the choices of the VoIP service being used by FreeWave.

Virtual sound card (VSC) A virtual sound card is a software application that uses a physical sound card installed on a machine to generate one (or more) isolated, virtual sound card interfaces on that machine. A virtual sound card interface can be used by any application running on the host machine exactly the same way a physical sound card is utilized. Also, the audio captured or played by a virtual sound card does not interfere with that of other physical/virtual sound interfaces installed on the same machine. We use virtual sound cards in the design of FreeWave to isolate the audio signals generated by FreeWave from the audio belonging to other applications.

MoDem FreeWave client and server software use a modulator/demodulator (MoDem) application that translates network traffic into acoustic signals and vice versa. This allows FreeWave to tunnel the network traffic of its clients over VoIP connections by modulating them into acoustic signals. We provide a detailed description of our MoDem design in Section 4.7.

Proxy FreeWave server uses an ordinary network proxy application that proxies the network traffic of FreeWave clients, received over VoIP connections, to their final Internet destinations. Two popular choices for FreeWave's proxy are the HTTP proxy [151] and the SOCKS proxy [152]; a SOCKS proxy supports proxying of a wide range of IP protocols, while an HTTP proxy only supports proxying of HTTP/HTTPS traffic, but it can perform HTTP-layer optimizations like pre-fetching of web contents. Several proxy solutions support both protocols.

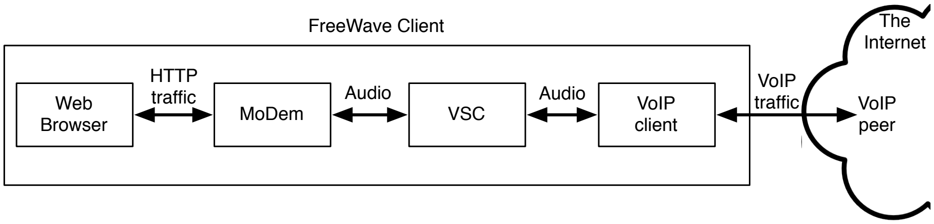

Figure 4.2: The main components of FreeWave client.

4.3.2 Client Design

The FreeWave client software, installed by a FreeWave user, consists of three main components described above: a VoIP client application, a virtual sound card (VSC), and the MoDem software. Figure 4.2 shows the block diagram of the FreeWave client design. MoDem transforms the data of the network connections sent by the web browser into acoustic signals and sends them over to the VSC component. The FreeWave MoDem also listens on the VSC sound card to receive specially formatted acoustic signals that carry modulated Internet traffic; MoDem extracts the modulated Internet traffic from such acoustic signals and sends them to the web browser. In a sense, the client web browser uses the MoDem component as a network proxy, i.e., the listening port of MoDem is entered in the HTTP/SOCKS proxy settings of the browser.

The VSC sound card acts as a bridge between MoDem and the VoIP client component, i.e., it transfers audio signals between them. More specifically, the VoIP client is set up to use the VSC sound card as its “speaker” and “microphone” devices (VoIP applications allow a user to select physical/virtual sound cards). This allows MoDem and the VoIP client to exchange audio signals that contain the modulated network traffic, isolated from the audio generated/recorded by other applications on the client machine.

For the FreeWave client to connect to a particular FreeWave server it *only* needs to know the VoIP ID belonging to that FreeWave server, but not the IP address of the FreeWave server. Every time the user starts up the FreeWave client application on her machine, the VoIP application of FreeWave client initiates an audio/video VoIP call to the known VoIP ID of the FreeWave server.

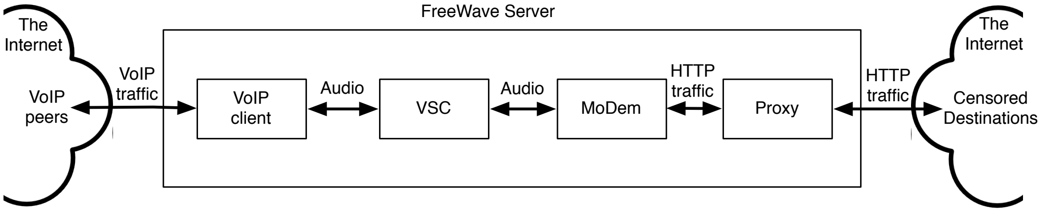

Figure 4.3: The main components of FreeWave server.

4.3.3 Server Design

Figure 4.3 shows the design of FreeWave server, which consists of four main elements. FreeWave server uses a VoIP client application to communicate with its clients through VoIP connections. A FreeWave server chooses one or more VoIP IDs, which are provided to its clients, e.g., through public advertisement.

The VOIP client of the FreeWave server uses one (or more) virtual sound cards (VSC) as its "speaker" and "microphone" devices. The number of VSCs used by the server depends on the deployment scenario, as discussed in Section 4.4.1. The VSC(s) are also used by the MoDem component, which transforms network traffic into acoustic signals and vice versa. More specifically, MoDem extracts the Internet traffic modulated by FreeWave clients into audio signals from the incoming VoIP connections and forwards them to the last element of the FreeWave server, FreeWave *proxy*. MoDem also modulates the Internet traffic received from the proxy component into acoustic signals and sends them to the VoIP client software through the VSC interface. The FreeWave proxy is a regular network proxy, e.g., an HTTP proxy, that is used by the FreeWave server to connect FreeWave clients to the open Internet. As mentioned above in Section 4.3.2, the web browser of a FreeWave client targets its traffic to a network proxy; such proxied traffic is received and handled by FreeWave server's proxy server (through the VoIP connections, as described).

4.4 Other Design Details

4.4.1 Deployment Scenarios

The FreeWave system proposed in this chapter can be deployed by "good" entities that run FreeWave servers to help censored users gain an uncensored access to the Internet. We consider the following scenarios for a real-world deployment of FreeWave. In Section 4.6, we discuss the security considerations for each of these scenarios.

Personal deployment: A person having an open access to the Internet can set up a personal FreeWave server on her personal machine, *anonymously* helping censored users evade censorship. Such a person can, then, advertise her VoIP ID (used with her FreeWave server) publicly (e.g., through social networks) and anyone learning this ID would be able to connect to the Internet by running FreeWave client software. To save bandwidth, she can configure her FreeWave server to enforce restrictions on the quality of service provided to clients.

Central VoIP-center: FreeWave service can be deployed and maintained by a central authority, e.g., a for-profit or non-profit organization. The deploying organization can build and run FreeWave servers that are a capable of serving large numbers of FreeWave clients. To do so, the deployed FreeWave servers should utilize several physical/virtual sound cards in parallel. Also, by creating VoIP accounts on several different VoIP service providers, such a central FreeWave system will be able to service FreeWave clients who use various VoIP services. Such a central deployment of FreeWave can operate for commercial profit, e.g., by charging clients for the used bandwidth, or can be established as a non-profit system, e.g., being funded by NGOs or pro-freedom governments.

Central phone-center: As an alternative approach, FreeWave can be deployed using an automated telephone center. More specifically, instead of VoIP IDs, FreeWave will publicize several phone numbers, which are used by clients to connect to the FreeWave server. FreeWave users need to use the exact same FreeWave client software, except that instead of making VoIP calls to a VoIP IDs they will make VoIP calls to FreeWave server's phone numbers. Compared to the "central VoIP-center" scenario, this has the big advantage that clients can arbitrarily choose any VoIP service provider for the client software, while in the “central service” design users need to choose from the VoIP systems supported by FreeWave server (though a powerful FreeWave server can support many VoIP systems).

Distributed service: FreeWave service can also be deployed in a distributed architecture, similar to that of Tor [118] anonymity network. More specifically, a FreeWave network can be built consisting of a number of volunteer computers that run instances of FreeWave server software on their machines. A central authority can manage the addition of new volunteer nodes to the system and also the advertisement (or distribution) of their VoIP IDs to the clients.

4.4.2 The Choice of VoIP Systems

There are numerous free/paid *VoIP service provider*s that can be utilized by the FreeWave system, e.g., Skype[2], Vonage[3], iCal[4], etc. A VoIP service provider usually supplies its VoIP client software to its users, but there are also some VoIP software that can be used for different VoIP accounts, e.g., PhonerLite[5]. In this section, we mention some candidate VoIP services that can be used by FreeWave.

Skype

Skype is a peer-to-peer VoIP system that provides voice calls, instant messaging, and video calls to its clients over the Internet. Skype is one of the most popular VoIP service providers with over 663 million users as of September 2011 [153].

Skype uses an undisclosed proprietary design, which has been partly reverse-engineered in some previous research [144, 145, 154]. These studies find that Skype uses a peer-to-peer overlay network with the Skype users as its peers. There are two types of nodes on Skype: *ordinary nodes*, and *supernodes (SN)*. Any Skype client with a public IP address, having sufficient CPU, memory,

[2] `http://www.skype.com`

[3] `http://www.vonage.com`

[4] `http://www.icall.com/`

[5] `http://www.phonerlite.de/index_en.htm`

and network bandwidth serves as a supernode, and all the other nodes are ordinary nodes. In addition, Skype uses a central *login server* that keeps users' login credentials and is used by Skype users to register into Skype's overlay network. Apart from the login server, all Skype communications work in a peer-to-peer manner, including the user search queries and online/offline user information.

A key feature that makes Skype an ideal choice for FreeWave is its peer-to-peer network. Depending on its network setting [143], an ordinary Skype user deploys some supernodes as her proxies to connect to the Skype network, to make/receive calls, and to update her status. In particular, a Skype call made toward an ordinary Skype node gets relayed to her by her supernodes [144, 145]. Each ordinary node maintains a *supernode-cache* [145] table that keeps a list of reachable (usually nearby) supernodes, discovered by the Skype protocol. We use this feature to provide server obfuscation for FreeWave: By having our FreeWave server act as an ordinary Skype node, the VoIP connections that it receives will be relayed by alternative supernodes, rendering IP address blocking impossible. We discuss this further in Section 4.6. Also note that a censor cannot map a FreeWave server to its supernodes since the supernode-cache table is a large, dynamic list; further, a Skype client can change its supernodes more frequently by *flush*ing [144,145] its supernode-cache.

Based on the criteria mentioned for a supernode, an easy way to be treated as an ordinary node by Skype is to reside in a firewalled, NATed network subnet [143, 145]. As another interesting feature of Skype for FreeWave is that all Skype connections are secured by end-to-end encryption [144, 145].

SIP-Based VoIP

Session Initiation Protocol (SIP) [155] is a lightweight, popular signaling protocol and is widely used by VoIP providers, e.g., SFLphone[6], Zfone[7], and Blink[8], to establish calls between clients. A SIP-based VoIP system consists of three main elements [155]: 1) *user agents* that try to establish SIP connections on behalf of users, 2) a *location service* that is a database

[6] http://sflphone.org/

[7] http://zfoneproject.com/

[8] http://icanblink.com/

keeping information about the users, and 3) a number of *servers* that help users in establishing SIP connections. In particular, there are two types of SIP servers; *registrar* servers receive registration requests sent by user agents and update the location service database. The second types of SIP servers are *proxy* servers that receive SIP requests from user agents and other SIP proxies and help in establishing the SIP connections.

Once a SIP connection is established between two user agents a media delivery protocol is used to transfer media between the users. Most of the SIP-based VoIP systems use the Real-time Transport Protocol (RTP) [156] to exchange audio data, and the Real-Time Transport Control Protocol (RTCP) [156] protocol to control the established RTP connections. User agents in a SIP-based VoIP system are allowed to use an encryption-enabled version of RTP, called Secure Real-time Transport Protocol (SRTP) [157], in order to secure their VoIP calls. Note that the encryption supported by SRTP is performed end-to-end by SIP agents and VoIP servers are not required to support encryption. We mandate the SIP-based design of FreeWave to use SRTP for media transfer.

Similar to Skype, if a user agent is behind NAT or a firewall, it will use an intermediate node to establish its VoIP connections. In particular, two popular techniques used by VoIP service providers to bypass NAT and firewalls are *session border controller* (SBC) [158] and *RTP bridge server*s [159]. As in the case of the Skype-based FreeWave, putting a FreeWave server behind a firewall masks its IP address from censors, as the VoIP calls to it will be relayed through oblivious intermediate nodes. However, better care needs to be taken in this case since, unlike Skype, SIP-based VoIP systems are not peer-to-peer.

Centralized VoIP

Several VoIP providers use their own servers to relay VoIP connections, in order to improve connectivity, regardless of the VoIP protocol that they use. One interesting example is the Google Voice[9], which relays all of its calls through Google servers, hence disguising a callee's IP address from a censor. Also note that the calls in Google Voice are encrypted.

[9] `https://www.google.com/voice`

4.5 Evaluation of the Design Goals

In Section 4.2.2, we listed several features that we consider in designing an effective circumvention system. Here, we discuss the extent to which our proposed system, FreeWave, achieves such requirements.

Unblockability: In order to use FreeWave, a client only needs to know the VoIP ID of the FreeWave server, i.e., `server-id`, but no other secret/public information like the server's IP address. `server-id` is distributed in a public manner to the users, so we assume that it is also known to censors. Considering the use of encrypted VoIP connections by FreeWave, this public knowledge of `server-id` does not allow censors to identify (and block) the VoIP connections to the FreeWave server. In addition, a censor will not be able to identify FreeWave's VoIP connections from their IP addresses since, as discussed in Section 4.4.2, the encrypted VoIP connections to the FreeWave server are relayed through oblivious, intermediate nodes (given the FreeWave server is set up appropriately). For instance, in Skype-based FreeWave the VoIP connections to the FreeWave server are relayed by oblivious Skype supernodes. Also, FreeWave server is not mapped to a particular set of supernodes, i.e., its VoIP connections are relayed through a varying set of super nodes. In all of the above arguments, we assume that the VoIP service provider used by FreeWave is not colluding with the censors; otherwise, the unobservability is lost. Such collusion could happen if a centralized VoIP service, e.g., Google Voice, informs censors of the clients calling FreeWave's Google Voice ID, or if the censors control the supernodes used by a FreeWave server.

Another point in making FreeWave unblockable is that it does not depend on a particular VoIP system, and can select from a wide range of VoIP providers. As a result, in order to block FreeWave, censors will need to block *all* VoIP services, which is very unlikely due to several political and economic considerations.

Note that unblockability is a serious challenge with many existing circumvention systems, as the very same information that they advertise for their connectivity can be used by censors to block them. For example, the Tor [118] system requires its clients to connect to a public set of IP addresses, which can be IP-filtered by censors. More recently, Tor has adopted the use of Tor *bridges* [125], which are volunteer proxies with semi-public IP addresses.

Unfortunately, there are different challenges [122, 126, 127, 130, 139] in distributing the IP addresses of Tor bridges only to real clients, but not to the censors.

Unobservability: The arguments made above for FreeWave's unblockability can also be used to justify its unobservability. As mentioned above, even though FreeWave server's VoIP ID (`server-id`) is assumed to be known to censors, the end-to-end encryption of VoIP connections prevents a censor from observing users making VoIP connections to `server-id`. In addition, VoIP relays sitting between FreeWave clients and a FreeWave server, e.g., Skype supernodes, foil the identification of FreeWave connections through IP address filtering.

Deployment feasibility: The real-world deployment of FreeWave does not rely on other entities. This is in contrast to some recent designs that need collaboration from third parties for their operation. For instance, Infranet [115] requires support from some web destinations that host the circumvention servers. As another example, several recent proposals [120, 121, 160] rely on the collaboration from friendly ISPs for their operation.

Quality of service: In Section 4.8, we discuss the connection performance provided by our prototype implementation of FreeWave. Our results show that FreeWave provides reliable connections that are good for normal web browsing.

4.6 Security Analysis

In this section, we discuss the security of FreeWave clients to the threats imposed by different entities.

4.6.1 Security Against Censors

The end-to-end encryption of VoIP connections protects the confidentiality of the data sent by FreeWave clients against a monitoring censor, even if the censor is able to identify VoIP connections targeted to FreeWave. Such end-to-end encryption also ensures the web browsing privacy of FreeWave clients. As mentioned in Section 4.4.2, Skype calls are encrypted end-to-end, and SIP-based VoIPs also provide end-to-end encryption using the SRTP protocol. In the case of centralized VoIP services, like the Google Voice, the encryptions are usually client-to-server; hence the FreeWave client should ensure that its VoIP provider is not colluding with the censors.

Even though FreeWave uses encrypted VoIP connections, a censor may still try to identify FreeWave-generated VoIP connections by performing traffic analysis, i.e., by analyzing communication patterns. The use of actual VoIP connections by FreeWave (instead of shaped connections as in [137, 138]) makes traffic analysis particularly hard. We show this in Section 4.8.3 by analyzing FreeWave's VoIP connections and comparing them with regular VoIP connections. As discussed in Section 4.8.3, the choice of the VoIP system affects the feasibility of traffic analysis. Please see Section 4.8.3 for more discussion on FreeWave traffic analysis.

4.6.2 Security Against FreeWave Servers

A FreeWave server only knows the VoIP IDs of its client, but not their IP addresses since the VoIP connections are being relayed through intermediate VoIP nodes. As a result, unless the VoIP service (e.g., the Google Voice server, or a Skype supernode owned by a FreeWave server) is colluding with the FreeWave server, the FreeWave server will not be able to link VoIP IDs to IP addresses, i.e., the client is anonymous to the server. Note that anonymity against circumvention systems is not demanded by typical censored users who are only willing to access non-sensitive censored information like the news, and in fact some popular circumvention mechanisms do not provide such anonymity, e.g., the single-proxy based systems such as the Anonymizer [119]. A FreeWave client can strengthen its anonymity against the FreeWave server in different ways. For instance, she can enforce its VoIP traffic to be relayed by additional intermediate VoIP relays, e.g., by the client's Skype supernodes.

In the basic design of FreeWave mentioned above, a FreeWave server can observe the traffic contents exchanged by a FreeWave client, since the tunneled traffic is not always encrypted. However, a client can easily ensure security and privacy from the server by using an extra layer of encryption. For instance, a client can use FreeWave to get connected to an anonymity system like Anonymizer [119], and then use the tunneled connection with this anonymity system to browse the Internet. This secures this client's traffic from the FreeWave server, as well as making it confidential. Note that considering the fact that FreeWave clients are anonymous to FreeWave servers, clients may opt not to use such an additional protection for low-sensitive activities like web browsing.

4.6.3 Security Against VoIP Providers

Except for the centralized VoIP services, the VoIP connections between FreeWave clients and servers are encrypted end-to-end using the keys shared through the VoIP protocol. In the case of a centralized VoIP service, like the Google Voice, FreeWave parties can exchange a key using a key sharing mechanism, like the Diffie-Hellman key exchange [161], over the established FreeWave VoIP. As a result, the VoIP provider will not be able to observe the data being communicated or the web destinations being browsed. However, the VoIP service provider might be able to identify VoIP IDs that have made VoIP calls to a FreeWave server. As a result, in order to ensure its unobservability FreeWave needs to use VoIP providers that are not colluding with the censors. Note that FreeWave does not rely on a particular VoIP system and any VoIP provider can be used for its operation.

4.7 FreeWave MoDem

The MoDem component is one of the main components of both FreeWave client and FreeWave server application, which translates Internet traffic into acoustic signals and vice versa. MoDem consists of a *modulator* and a *demodulator*. MoDem's modulator modulates data (IP bits) into acoustic signals, and MoDem's demodulator extracts the encoded data from a received acoustic signal. In the following, we describe the design of MoDem's modulator and demodulator.

4.7.1 Modulator Description

We design a *bit-interleaved coded modulation* (BICM) [149] for MoDem's modulator, which is shown in Figure 4.4. First, the modulator encodes the

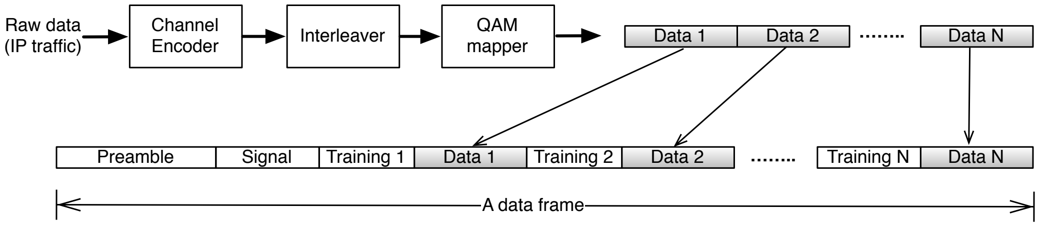

Figure 4.4: The modulator block of FreeWave's MoDem. The modulated data is wrapped by a wrapper protocol before being transformed into acoustic waveforms.

information bits, $\{a_i\}$, i.e., IP traffic, using a channel encoder with rate R_c. The encoded stream, $\{b_i\}$, is permuted using a random interleaver [149], and the interleaved sequence is then partitioned into subsequences $\boldsymbol{c}_n = \{c_n^1, \ldots, c_n^Q\}$ of length-Q (n is the partition index and Q is a parameter of our modulator). Finally, a QAM mapper [149] generates the modulated data by mapping each subsequence $\boldsymbol{c_n}$ to a 2^Q-ary quadrature amplitude modulation symbol.

We design a wrapper protocol to carry the modulated data. This wrapper performs three important tasks: 1) it allows a demodulator to synchronize itself with the modulator in order to correctly identify the starting points of the received data; 2) it lets the sender and receiver negotiate the modulation parameters; and, 3) it lets the demodulator adapt itself to the time-varying channel. Figure 4.4 shows the modulated data being wrapped by our wrapper protocol. As can be seen, the modulated bit stream is converted into data *frame*s that are sent over the VoIP channel. Each data frame starts with a known *preamble* block, which is needed for synchronization as well as for receiver initialization purposes. The frame preamble is followed by a *signal* block that is used to communicate the modulation and coding parameters used for this particular frame. The signal block is followed by N blocks of training and data symbols. The data symbols are the output of the QAM modulator. The training blocks are needed to adapt the demodulator to the time-varying channel.

The data frames, as generated above, are sent over the VoIP channel using acoustic signals. In particular, for x_n being the n-th symbol in a frame, the frame is mapped to a waveform $x(t) : \mathbb{R} \rightarrow \mathbb{C}$ as follows:

$$x(t) = \sum_{l} x_l p(t - lT) \tag{4.1}$$

where $p(t)$ is a basic pulse shifted by multiples of the symbol period T. This signal is then transformed to a passband [84] signal with the center frequency of f_C:

$$x_{PB}(t) = 2\,\mathrm{Re}\{x(t)e^{2\pi i f_C t}\} \tag{4.2}$$

which is then sent over the VoIP channel (by getting sent to the virtual sound card). Re{} returns the real component of a complex number, and i is the imaginary unit.

4.7.2 Demodulator Description

Figure 4.5 shows MoDem's demodulator, which is designed to effectively extract the data that the modulator embedded into an audio signal. For an audio waveform, $r(t)$, received from the virtual sound card, the demodulator shifts its spectrum by the center frequency f_C, passes it through a low-pass filter and then samples the resulting signal at symbol rate (equal to $1/T$). The synchronizer correlates the preamble block with the obtained samples, declares the point of maximum correlation as the starting point of the received frame, discards all samples before this point, and enumerates the remaining samples by $r_n (n = 1, 2, ...)$. We assume the voice channel to be linear and can hence write [84]:

$$r_n = \sum_{k=-K_f}^{K_p} h_{n,k} x_{n-k} + w_n \tag{4.3}$$

where n and k are time and delay indices, respectively. Also, w_n is a complex white Gaussian noise process, which models the noise added to the modulated data as a result of the noisy channel (e.g., due to VoIP codec's lossy compression). Moreover, $h_{n,k}$ is the channel gain [84], which may vary in time. The channel length is assumed to be at most $K_f + K_p + 1$, where K_f is

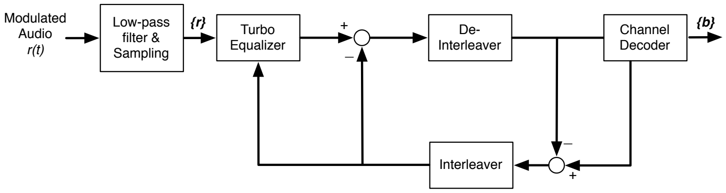

Figure 4.5: Block diagram of MoDem's demodulator.

the length of the precursor and K_p is the length of the postcursor response.

The demodulator passes the discrete stream of $\{r\}$ through a Turbo equalizer [84]. The goal of this equalizer is to obtain an estimation of $\{x\}$, i.e., the discrete modulated data. The estimated data is passed to a channel decoder, which is the equivalent decoder for the encoder used by MoDem's modulator. We also put an interleaver and a de-interleaver block between the Turbo equalizer and the channel decoder modules; this is to uniformly distribute burst bit errors, generated in the channel, across the stream in order to improve the decoding process. This is because our channel decoder performs well with distributed errors, but poorly with bursty errors.

4.8 Prototype and Evaluation

In this section, we describe our prototype implementation and discuss its connection performance.

4.8.1 Implementation Setup

We have built a prototype implementation of FreeWave over Skype. Our MoDem component uses Matlab's libraries for acoustic signal processing, and we use Virtual Audio Card [10] as our virtual sound card (VSC) software. We also use the free version of Skype client software[11] provided by Skype Inc. as our VoIP client component. Our MoDem software, as well as the Skype client, is set up to use the Virtual Audio Card as its audio interface. We have built our FreeWave client and FreeWave server using the components mentioned above. In order to emulate a real-world experience, i.e., a long distance between a FreeWave client and a FreeWave server, we connect our

[10] `http://software.muzychenko.net/eng/vac.htm`

[11] `http://www.skype.com/intl/en-us/get-skype/`

FreeWave client to the Internet though a VPN connection. In particular, we use the SecurityKISS[12] VPN solution that allows us to pick VPN servers located in different geographical locations around the world. Note that this identifies the location of our FreeWave clients; our FreeWave server is located in Champaign, IL, USA.

MoDem specifications: Our evaluations show that the data rates that can be achieved with our system clearly depend on the bandwidth of the Internet connection and the distance between the client and server. The minimum bandwidth required for a voice call is 6 kbps for both upload and download speeds, according to Skype. For the pulse function of MoDem's modulator, $p(t)$ (Section 4.7), we use a square-root raised cosine filter with a roll-off factor 0.2 and a bandwidth of $1/T$. The carrier frequency f_C is chosen such that the spectrum of the voiceband is always covered. At the demodulator, the same square-root raised cosine filter is used for low-pass filtering. Our communication system automatically adjusts the symbol constellation size Q, the channel coding rate R_c, and the symbol period T such that the best possible data rate is achieved. The receiver knows how well the training symbols were received, and based on this feedback the modulator can optimize the data rate. The relationship between the data rate R and the above parameters is $R = (QR_c)/T$. Our designed demodulator is iterative [84]. The number of iterations needed for convergence depends on the channel condition, which is typically measured by means of the signal to noise power ratio, the SNR.

4.8.2 Connection Performance

Connection data rates: Table 4.1 shows the bit rates achieved by FreeWave clients connecting from different geographic locations to our FreeWave server, located in Champaign, IL, USA. At the beginning of each FreeWave connection, our client runs an assessment subprotocol to identify the best codecs and the reliable data rate. The table lists the best compromise between data rates and packet drop rates, for different clients. As can be seen, clients in different parts of Europe are reliably able to get connection bit rates of 16kbps by using FreeWave over Skype. Users within the US are

[12] `http://www.securitykiss.com/`

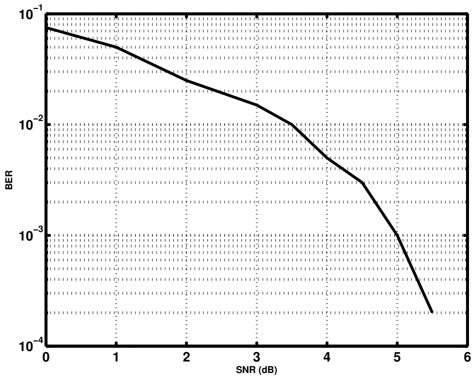

Figure 4.6: BER versus SNR for FreeWave.

Table 4.1: Evaluation results of FreeWave.

Client location	MoDem parameters			Data rate	Packet drop rate
	Q	$1/T$	R_C		
Berlin, Germany	4	8 kHz	0.5	16000 bps	0
Frankfurt, Germany	4	8 kHz	0.5	16000 bps	0
Paris, France	4	8 kHz	0.5	16000 bps	0
Maidenhead, UK	4	8 kHz	0.5	16000 bps	0
Manchester, UK	4	8 kHz	0.5	16000 bps	0
Lodz, Poland	4	8 kHz	0.5	16000 bps	0.06
Chicago, IL	4	9.6 kHz	0.5	19200 bps	0.01
San Diego, CA	4	9.6 kHz	0.469	18000 bps	0

able to achieve higher data rates, e.g., 19.2kbps for a client in Chicago, IL. Note that the distance between a FreeWave client and the FreeWave server slightly affects the achievable data rates. To illustrate this, Figure 4.6 shows the bit error rate (BER) performance of our designed demodulator for different SNRs in the log-scale for a 19kbps FreeWave connection. As can be seen, for SNRs larger than 5.4dB the BER tends to zero (the zero value cannot be shown in the log-scale figure). A distributed deployment of FreeWave can provide users from many different geographic locations with the same reliable data rate speeds; for instance, FreeWave servers running in Europe can assist FreeWave users from the Middle East better than the FreeWave servers that are located in the US.

Maximum achievable data rates: As illustrated above, our FreeWave prototype is able to reliably achieve bit rates of up to 19kbps, using the MoDem component designed in this chapter. It is possible to design more complicated MoDems that can achieve higher bit rates; however, a MoDem will not be able to achieve arbitrarily large data rates. This is due to the fact that each VoIP codec samples speech at a particular rate (or at a given range of rates) [162] and FreeWave cannot achieve data rates higher than a codec's bit-rate. For instance, Skype generates a bit-rate between 6 and 40kbps [162] (depending on the distance between the end-hosts, Internet bandwidth and few other factors), resulting in a "maximum" achievable rate of 40kbps for FreeWave (the actual rate achieved depends on the efficiency of MoDem). The "L16" codec generates a 128kbps data rate, resulting in a maximum FreeWave bit-rate of up to 128kbps. As another instance, the widely used codec of "G.711" produces a 64kbps data rate [162], leading to a maximum FreeWave bit rate of 64kbps.

We believe that the bit rates achievable by the current design of FreeWave are enough for normal web browsing, especially for a user under a repressive regime who aims to do normal web browsing. On the other hand, a trivial approach to achieve much higher rates is to encode Internet traffic into the *video* signals carried over VoIP connections. This requires designing efficient modulator/demodulators for encoding data into video, which we leave for future research.

4.8.3 Traffic Analysis

In order to resist traffic analysis, FreeWave VoIP connections should have communication patterns similar to that of regular VoIP connections. Note that FreeWave uses encrypted VoIP connections, so a censor will not be able to analyze packet contents (popular VoIP providers like Skype provide/mandate encrypted VoIP connections). The two traffic patterns that may be used for traffic analysis in this case are *packet rates* and *packet sizes.* Most of the standard VoIP codecs, like the widely used G.7 series [162], use fixed bit rates and fixed packet sizes during a given connections, or even across all connections [162]. This prevents any kind of traffic analysis against FreeWave connections that use these codecs. In fact, these codecs are widely used by different VoIP providers, e.g., the Google Voice service [163]. On the other hand, several VoIP codecs use variable bit-rates, most notably Skype's proprietary SILK [164] codec. When FreeWave uses a VoIP service that uses variable-bit-rate codecs, special care needs to be taken to prevent traffic analysis. We have analyzed the FreeWave traffic sent over Skype in our prototype implementation, and have compared its traffic patterns with regular Skype traffic. We observe that there are two states in a regular Skype call: “Skype-Speak”, in which the callee is speaking over Skype, and “Skype-Silence”, in which the callee is silent (e.g., she is listening to the person on the other side of the line).

Table 4.2 shows the average communication statistics for the three different types of Skype traffic, i.e., Skype in the Skype-Speak state, Skype in the Skype-Silent, and Skype tunneling FreeWave. All the analysis is done for the same pair of Skype peers. As can be seen from the table, FreeWave over Skype generates communication patterns very similar to regular Skype in the Skype-Speak state, while the Skype-Silent state generate lower packet rates and smaller packet sizes. This is because in order to conserve bandwidth Skype’s SILK [164] codec reduces its packet rate and uses smaller packets when the audio signal captured by the Skype client is weak. We observe that, based on this analysis, a FreeWave over Skype call makes communication patterns very similar to a typical Skype call: In a typical Skype call, when one side of the connection is in the Skype-Speak state, the other side is usually in the Skype-Silent state (i.e., listening to the other side). In a FreeWave over Skype call, also, when one side of the connection is sending data the other side is usually idle, e.g., a web traffic is a serious of HTTP GET and HTTP RESPONSE messages that appear in a sequence. Furthermore, simple modifications can be made to FreeWave client and server software in order to better hide its traffic pattern; for instance, one side can stop sending data if the other side is sending data, or a dummy audio can be sent if both sides have been silent for a long time. Once again, note that this is only required if FreeWave is deployed on a VoIP system that uses a variable-length audio codec.

4.9 Comparison with Similar Systems

Recently, there have been two proposals for censorship circumvention that, similar to FreeWave, use the openness of VoIP to evade censorship. Due to their similarity with FreeWave we describe the advantages of FreeWave over them in this section.

Table 4.2: Comparing communication patterns of regular Skype with FreeWave-over-Skype.

Pattern	FreeWave over Skype	Skype-Speak	Skype-Silent
Average packet rate (pps)	49.91	50.31	49.57
Average packet size	148.64	146.50	103.97
Minimum packet size	64	64	64
Maximum packet size	175	171	133

4.9.1 SkypeMorph

SkypeMorph [138] is a pluggable transport [134] for Tor. SkypeMorph is designed to obfuscate the connections between Tor [118] users and Tor bridges [125] so that they look like legitimate Skype traffic. The main goal of SkypeMorph is to make it hard for a censor to distinguish between obfuscated Tor bridge connections and actual Skype calls using deep-packet inspection and statistical traffic analysis. A big implementation-wise difference with our proposal is that SkypeMorph does not completely run, but mimics, Skype, whereas FreeWave runs the target VoIP protocol in its entirety. FreeWave has the following main advantages over SkypeMorph:

Server obfuscation: Similar to the most of existing obfuscation-based techniques, SkypeMorph only provides traffic obfuscation, but it does not provide server obfuscation. A censor may not be able to identify SkypeMorph traffic through statistical analysis, since SkypeMorph shapes it to look like a regular Skype traffic. However, if a censor discovers the IP address of a SkypeMorph Tor bridge, e.g., through bridge enumeration [126,127], SkypeMorph's obfuscations do not provide any protection since the censor can easily block its traffic by IP addresses matching. As an indication to the severity of this problem, the Chinese censors were able to enumerate *all* bridges in under a month [140]. Once a Tor bridge is known to a censor, SkypeMorph is not able to provide *any* protection.

On the other hand, FreeWave provides server obfuscation in addition to traffic obfuscation. Instead of morphing the traffic into VoIP, FreeWave uses the overlay network of VoIP systems to route the connections among users and servers. As a result, FreeWave's VoIP traffic gets relayed by "oblivious" VoIP nodes, hiding the identity (e.g., the IP address) of the FreeWave server. Even a censor who knows the IP address of a FreeWave server will not be able to identify and/or block client connections to that server, since these connections do not go directly to the server. For instance, if Skype is used by FreeWave, the FreeWave connections get relayed by *Skype supernodes*, which are oblivious Skype users residing “outside” the censoring ISP (please see Section 4.4.2 for further discussion). Note that there is not a one-to-one correspondence between supernodes and FreeWave servers, i.e., various supernodes relay traffic to a particular FreeWave server for different connections. As another example, if Google Voice is used by FreeWave, all the FreeWave connections get relayed by Google servers, hiding FreeWave servers' IP addresses. Note that we assume that VoIP connections are also encrypted.

Comprehensive traffic obfuscation SkypeMorph shapes Tor traffic into Skype calls, but it does not run the actual Skype protocol (except for the Skype login process) [138]. This can enable sophisticated attacks that can discriminate SkypeMorph from Skype by finding protocol details that are not properly imitated by SkypeMorph. For instance, SkypeMorph fails to mimic Skype's TCP handshake [165], which is essential to every genuine Skype call. Also, Skype protocol may evolve over time and SkypeMorph would need to follow the evolution. FreeWave, on the other hand, runs the actual VoIP protocol in its entirety, providing a more comprehensive traffic obfuscation.

No need to pre-share secret information: SkypeMorph needs to secretly share its Skype ID with its clients, as well as its IP address and port number (this can be done using Tor's BridgeDB [166] as suggested by the authors). Once this secret information is disclosed to a censor (e.g., through bridge enumeration) the identified Tor bridge will need to change both its IP address and its Skype ID, as suggested in [138], to reclaim its accessibility by clients. FreeWave, however, does not need to share *any* information with its clients: even the VoIP IDs of the FreeWave servers are publicly advertised without compromising the provided unobservability.

Obfuscation diversity: SkypeMorph is designed to morph traffic only into Skype. As a result, if a censor decides to block Skype entirely, SkypeMorph will be blocked as well. FreeWave, on the other hand, is a general infrastructure and can be realized using a wide selection of VoIP services. Needless to say, SkypeMorph may also be modified to mimic other popular VoIP services, but it requires substantial effort in understanding and analyzing the candidate VoIP system. FreeWave, however, can be used with *any* VoIP service without the need for substantial modifications.

4.9.2 CensorSpoofer

A key goal in the design of CensorSpoofer [141] is to provide unobservability, as is the case in FreeWave. CensorSpoofer decouples upstream and downstream flows of a connection; the upstream flow, which is supposed to be low-volume, is steganographically hidden inside instant messages (IM) or email messages that are sent towards the secret IM or email addresses of the CensorSpoofer server. The IM IDs or the email addresses of the CensorSpoofer server need to be shared securely with clients through out-of-band channels. The CensorSpoofer server sends the downstream flow of a connection by spoofing a randomly chosen IP address, in order to obfuscate its own IP address. This spoofed flow is morphed into an encrypted VoIP protocol to obfuscate traffic patterns as well. A CensorSpoofer client also needs to generate "dummy" packets towards the spoofed IP address to make the connection look bidirectional. FreeWave makes the following contributions over CensorSpoofer:

No invitation-based bootstrapping: A new CensorSpoofer client needs to know a *trusted* CensorSpoofer client in order to bootstrap [141]. The trusted client helps the new client to send her personalized upstream ID and SIP ID to the CensorSpoofer server. Finding an existing, trusted CensorSpoofer client might be challenging for many new clients unless CensorSpoofer is widely deployed. Also note that even an existing CensorSpoofer client needs to re-bootstrap its CensorSpoofer connectivity if her personalized CensorSpoofer IDs are discovered by the censors. FreeWave, on the other hand, does not require an invitation-based bootstrapping.

Comprehensive traffic obfuscation Unlike FreeWave and similar to SkypeMorph, CensorSpoofer does not entirely run the VoIP protocol. This can enable sophisticated attacks that are able to find protocol discrepancies between CensorSpoofer and genuine VoIP traffic. Also, the use of IP spoofing by CensorSpoofer may enable active traffic analysis attacks that manipulate its downstream VoIP connection and watch the server's reaction.

Bidirectional circumvention: In CensorSpoofer VoIP connections only carry the downstream part of a circumvented connection. The upstream data are sent through *low-capacity* steganographic channels inside email or instant messages [141]. FreeWave, however, provides a high-capacity channel for both directions of a circumvented connection.

4.10 Related Work

Censorship circumvention systems have been evolving continuously to keep up with the advances in censorship technologies. Early circumventions systems simply used network proxies [167] residing outside censorship territories, trying to evade the simple IP address blocking and DNS hijacking techniques enforced by pioneer censorship systems. Examples of such proxy-based circumvention tools are DynaWeb [116], Anonymizer [119], and Freenet [168].

Proxy-based circumvention tools lost their effectiveness with the advent of more sophisticated censorship technologies such as deep-packet inspection [112, 113]. Deep-packet inspection analyzes packet contents and statistics looking for deviations from the censor's regulations. This has led to correspondingly more sophisticated circumvention tools that remain accessible to their users. Many circumvention designs seek availability by sharing some *secret* information with their users so that their utilization is unobservable to the censors agnostic to this secret information. In Infranet [115], for instance, a user needs to make a special, secret sequence of HTTP requests to an Infranet server to request censored web contents, which are then sent to him using image steganography. Collage [117] similarly bases its unobservability on sharing secrets with its clients. A Collage client and the Collage server secretly agree on some user-generated content sharing websites, e.g., flickr.com, and use image steganography to communicate through these websites. The main challenge for these systems, which rely on pre-sharing secret information, is to be able to share secret information with a large set of actual users while keeping them secret from censors; this is a big challenge to solve as indicated in several researches [128–130]. Sharing secret information with users has also been adopted by the popular Tor [118] anonymity network. The secret pieces of information here are the IP addresses of volunteer Tor relays, known as Tor bridges [125], that proxy the connections of Tor clients to the Tor network. This suffers from the same limitation as censors can pretend to be real Tor users and gradually identify a large fraction of Tor bridges [126, 127, 139].

More recently, several researches propose to build circumvention into the Internet infrastructure [120,121,160]. Being built into the Internet infrastructure makes such circumvention highly unobservable: a client's covert communication with a censored destination appears to the censor to be a benign connection to a non-prohibited destination. Telex [120], Cirripede [121] and Decoy Routing [160] are example designs using such infrastructure-embedded approach. Decoy Routing needs to share secrets with its clients using out-of-band channels, whereas Telex and Cirripede share the secret information needed to initialize their connections using covert channels inside Internet traffic. Cirripede uses an additional client registration stage performed steganographically, distinguishing it from the other designs. Even though these systems are a large step forward in providing unobservable censorship circumvention, their practical deployment is not trivial as they need to be deployed by a number of real-world ISPs that will make software/hardware modifications to their network infrastructures, posing a substantial deployment challenge.

Another research trend uses traffic obfuscation to make circumvented traffic unobservable. Appelbaum et al. propose a platform that allows one to build protocol-level obfuscation plugins for Tor, called *pluggable transports* [134]. These plugins obfuscate a Tor client's traffic to Tor bridges by trying to remove any statistical/content pattern that identifies Tor's traffic. Obfsproxy [135], the pioneer pluggable transport, removes all content identifiers by passing a Tor client's traffic through an additional layer of stream cipher encryption. Obfsproxy, however, does not disguise the statistical patterns of Tor's traffic. SkypeMorph [138] and StegoTorus [137] attempt to remove Tor's statistical patterns as well by morphing it into popular, uncensored Internet protocols such as Skype and HTTP. Flashproxy [169] is another recently designed pluggable transport that separates a Tor client's traffic into multiple connections, which are proxied by web browsers rendering volunteer websites.

CensorSpoofer [141] is another recent proposal that, similar to SkypeMorph [138], shapes Tor traffic into VoIP protocols. CensorSpoofer is unique in separating the upstream and downstream flows of a circumvented connection, and in using IP spoofing to obfuscate its server's identity. A security concern with morphing approaches [137,138,141,170] is that they do not provide a provable indistinguishability; censors may be able to devise advanced statistical classifiers and/or protocol identifiers to find discrepancies between a morphed traffic and genuine connections. Another approach that similarly uses VoIP traffic is TranSteg [171]; it re-encodes a VoIP call packets using a different, lower-rate codec in order to free a portion of VoIP packet payloads, which are then used to send a low-bandwidth hidden traffic.

4.11 Limitations and Recommendations

Server location In order to achieve server obfuscation, special care needs to be taken in setting up a FreeWave server. In the case of Skype, for instance, the FreeWave server should be completely firewalled such that its Skype traffic is completely handled by Skype supernodes. Also, a FreeWave server should use a large, dynamic set of supernodes (i.e., by flushing its supernode cache [144, 145]) so that one cannot map a FreeWave server to its supernodes. A corrupt supernode (e.g., controlled by the censors) used by a FreeWave server can identify the clients that used FreeWave through that supernode. The mechanisms to protect server obfuscation vary depending on the utilized VoIP system.

Traffic analysis If the VoIP service deployed by FreeWave uses a variable-length audio codec, like SILK [164], FreeWave's traffic might be subject to traffic analysis. In Section 4.8.3, we showed that the current deployment of FreeWave over Skype performs well against simple traffic analysis, yet more sophisticated traffic analysis [172] may be able to distinguish FreeWave's current prototype from Skype. A trivial countermeasure is to add some prerecorded human speech to FreeWave's audio, which would further reduce FreeWave's data rate. A better approach is to encode FreeWave's traffic into video, instead of audio, which is more robust to traffic analysis and provides much higher throughputs.

Trusting the VoIP provider A VoIP provider colluding with censors can significantly degrade FreeWave's obfuscation promises if FreeWave deploys it. On the bright side, FreeWave can choose from a wide range of VoIP providers.

In the case of Skype, in particular, Chinese Skype users get provided with a special implementation of Skype, TOM-Skype, which is suspected [173] to have built-in surveillance functionalities such as text message filtering [174–177].

Denial of service Since FreeWave's VoIP IDs are public, censors can exhaust FreeWave servers by making many FreeWave connections. Different approaches can be taken to limit the effect of such attempts, such as the existing sybil defense mechanisms [178], as well as usage limitation enforcement per VoIP caller.

4.12 Conclusions

In this chapter, we presented FreeWave, a censorship circumvention system that is highly unblockable by censors. FreeWave works by modulating a client's Internet traffic inside the acoustic signals that are carried over VoIP connections. Being modulated into acoustic signals, as well as the use of encryption, makes FreeWave's VoIP connections unobservable by a censor. By building a prototype implementation of FreeWave we show that FreeWave can be used to achieve connection bit rates that are suitable for normal web browsing.

APPENDIX A

PROOFS

A.1 Proof of Lemma 2

Proof. Assume N is the maximal size of an ϵ-Code such that $\boldsymbol{D}^{(i)} \subset F(\boldsymbol{x}^{(i)})$, where

$$F(\boldsymbol{x}) = \{\boldsymbol{y} : i(\boldsymbol{x}, \boldsymbol{y}) > \theta\} \tag{A.1}$$

Then we have

$$P(\boldsymbol{D}^{(i)}) = \int_{\boldsymbol{D}^{(i)}} P(d\boldsymbol{y}) < \int_{\boldsymbol{D}^{(i)}} e^{-\theta} P(d\boldsymbol{y}|\boldsymbol{x}) \leq e^{-\theta} \tag{A.2}$$

and

$$P(\cup_i \boldsymbol{D}^{(i)}) \leq \sum_i P(\boldsymbol{D}^{(i)}) \leq N e^{-\theta} \tag{A.3}$$

Let $\boldsymbol{D} = \cup_i \boldsymbol{D}^{(i)}$. By the maximality of N it follows that

$$P(\boldsymbol{D}^c \cap F(\boldsymbol{x})|\boldsymbol{x}) < 1 - \epsilon \tag{A.4}$$

Or equivalently

$$\epsilon < P(\boldsymbol{D} \cup F^c(\boldsymbol{x})|\boldsymbol{x}) \leq P(\boldsymbol{D}|\boldsymbol{x}) + P(F^c(\boldsymbol{x})|\boldsymbol{x}) \tag{A.5}$$

Multiplying this inequality with $P(d\boldsymbol{y})$ and integrating it over $\boldsymbol{x}$ then yields

$$\epsilon \leq P(\boldsymbol{D}) + P(i(\boldsymbol{x}, \boldsymbol{y}) \leq \theta) \tag{A.6}$$

Putting everything together we obtain the result

$$\epsilon - P(i(\boldsymbol{x},\boldsymbol{y}) \le \theta) \le P(\boldsymbol{D}) \le Ne^{-\theta} \tag{A.7}$$

□

A.2 Proof of Theorem 9

The Markov chain $\varPsi$ is aperiodic and irreducible. The state space of $\varPsi_i$ can be chosen to be $\mathbb{X} = \mathbf{2} \times \mathbb{N} \cup \{(0,0)\}$. First we verify that Foster's criterion holds

Lemma 3. *There exists a Lyapunov function $V : \mathbb{X} \to (0,\infty]$, finite at some $\psi_0 \in \mathbb{X}$, a finite set $\mathbb{S} \subset \mathbb{X}$, and $b < \infty$ such that*

$$\mathbb{E}[V(\varPsi_{i+1}) - V(\varPsi_i)|\varPsi_i = \psi] \le -1 + b\mathbf{1}_{\mathbb{S}}(\psi), \quad \psi \in \mathbb{X} \tag{A.8}$$

Further this function V is Lipschitz, i.e., for some $\alpha > 0$

$$|V(y) - V(x)| \le \alpha||y - x|| \quad \forall y, x \in \mathbb{X} \tag{A.9}$$

and for some $\beta > 0$ and

$$\sup_{x\in\mathbb{X}} \mathbb{E}[e^{\beta||\varPsi_{i+1}-\varPsi_i||}|\varPsi_i = x] < \infty \tag{A.10}$$

Proof. We need to find a function V such that $\mathbb{E}[V(\varPsi_{i+1}) - V(\varPsi_i)|\varPsi_i = \psi] \le -1$ for all but a finite number of $\psi \in \mathbb{X}$. If we simply choose $V(\tilde{y}, q) = cq$ for some sufficiently large constant $c > 0$, then the requirement is clearly satisfied for all $\psi \in \mathbb{X}$ such that $\tilde{y} = 1$, but it fails to hold otherwise. To fix this shortcoming we reward the transitions to a state with $\tilde{y} = 1$ by a decreasing difference $V(\varPsi_{i+1}) - V(\varPsi_i)$. In particular we choose $V(\tilde{y}, q) = (q - \tilde{y})/(\mu - \lambda)$. Standard calculations reveal that for that choice $\mathbb{E}[V(\varPsi_{i+1}) - V(\varPsi_i)|\varPsi_i = \psi] = -1$ for all $\psi \in \mathbb{X}$ with $q > 1$. Linear functions are always Lipschitz and $||\varPsi_{i+1} - \varPsi_i||$ is bounded almost surely. □

By the results in [105] or Proposition A.5.7 in [106], the chain $\varPsi$ is then geometrically ergodic.

By Theorem A.5.8 in [106], the asymptotic variance σ^2 is well defined, non-negative and finite, and

$$\sigma^2 = \mathrm{Var}(f(\Psi_0)) + 2\sum_{i=1}^{\infty} \mathrm{Cov}(f(\Psi_0), f(\Psi_i)) \tag{A.11}$$

Finally, $f(\tilde{y}, q)$ is a bounded, nonlattice, real-valued functional on the state space $\mathbb{X}$ and hence

$$P_{\psi_0}\left(\frac{\sum_{i=0}^{n-1} f(\tilde{y}_i, q_i) - n\pi_\Psi(f)}{\sigma\sqrt{n}} \leq \xi\right) - \Phi(\xi) \tag{A.12}$$

$$= \frac{p_\Phi(\xi)}{\sigma\sqrt{n}}\left[\frac{\eta}{6\sigma^2}(1-\xi^2) - \hat{f}(\psi_0)\right] + o(n^{-1/2}) \tag{A.13}$$

where $p_\Phi(\xi)$ denotes the density of the standard Normal distribution Φ, $\hat{f}$ is the solution to Poissons equation and η is a constant [107]. The solution $\hat{f}$ can be chosen such that $\pi_\Psi(\hat{f}) = 0$ and the claim follows by averaging out ψ_0.

A.3 Proof of Theorem 10

Using the representation of σ^2 in Equation 3.16, it remains to find explicit expressions for $\mathrm{Var}(f(\Psi_0))$ and the sum $\sum_{i=0}^{\infty} \mathrm{Cov}(f(\Psi_0), f(\Psi_i))$.

The term $\mathrm{Var}(f(\Psi_0))$ is easy to compute

$$\begin{aligned}\mathrm{Var}(f(\Psi_0)) = & \log^2(\frac{1}{\lambda})\pi_Q(0) + \log^2(\frac{\mu}{\lambda})\mu\overline{\pi_Q(0)} \\ & + \log^2(\frac{\bar{\mu}}{\lambda})\bar{\mu}\overline{\pi_Q(0)} - C^2\end{aligned} \tag{A.14}$$

Equation 3.21 holds true.

Proof. For the computation of the sum $\sum_{i=0}^{\infty} \mathrm{Cov}(f(\Psi_0), f(\Psi_i))$ we will set up and solve a recursion. Grassmann proposed this approach in [109] to obtain the asymptotic variance of a continuous time finite state birth-death process.

We define

$$r(\psi, i) = \sum_{\psi' \in \mathbb{X}} (f(\psi') - C)\pi_\Psi(\psi')p_{\Psi_i|\Psi_0}(\psi|\psi') \tag{A.15}$$

Clearly

$$r(\psi, 0) = (f(\psi) - C)\pi_{\Psi}(\psi) \tag{A.16}$$

and

$$\mathrm{Cov}(f(\Psi_0), f(\Psi_i)) = \sum_{\psi \in \mathbb{X}} (f(\psi) - C) r(\psi, i) = \sum_{\psi \in \mathbb{X}} f(\psi) r(\psi, i) \tag{A.17}$$

Note however that for the computation of the asymptotic variance we actually do not even need to know this covariance for each i. It is sufficient to know its sum. So we define

$$R(\psi) = \sum_{i=0}^{\infty} r(\psi, i) \tag{A.18}$$

write

$$\sum_{i=0}^{\infty} \mathrm{Cov}(f(\Psi_0), f(\Psi_i)) = \sum_{\psi \in \mathbb{X}} f(\psi) R(\psi) \tag{A.19}$$

and derive a recursion for $R(\psi)$.

For mean ergodic Markov processes $p_{\Psi_i|\Psi_0}(\psi|\psi') \to \pi_{\Psi}(\psi)$ as $i \to \infty$ and hence

$$\lim_{i \to \infty} r(\psi, i) = 0 \tag{A.20}$$

Summing $r(\psi, i+1) - r(\psi, i)$ in i from zero to infinity then clearly yields $(C - f(\psi))\pi_{\Psi}(\psi)$. By the Chapman-Kolmogorov equations

$$\begin{aligned} &p_{\Psi_{i+1}|\Psi_0}(\psi|\psi') - p_{\Psi_i|\Psi_0}(\psi|\psi') \\ &\quad = \sum_{\psi'' \in \mathbb{X}} p_{\Psi_i|\Psi_0}(\psi''|\psi') \left\{ p_{\Psi_{i+1}|\Psi_i}(\psi|\psi'') - \delta_{\psi,\psi''} \right\} \end{aligned} \tag{A.21}$$

and thus

$$\begin{aligned}
&r(\psi, i+1) - r(\psi, i) \\
&= \sum_{\psi' \in \mathbb{X}} (f(\psi') - C)\pi_\Psi(\psi') \left\{ p_{\Psi_{i+1}|\Psi_0}(\psi|\psi') - p_{\Psi_i|\Psi_0}(\psi|\psi') \right\} \\
&= \sum_{\psi'' \in \mathbb{X}} \left\{ p_{\Psi_{i+1}|\Psi_i}(\psi|\psi'') - \delta_{\psi,\psi''} \right\} r(\psi'', i) \qquad \text{(A.22)}
\end{aligned}$$

If this expression for $r(\psi, i+1) - r(\psi, i)$ is also summed in i from zero to infinity and then compared to the above result of the same sum, we obtain

$$(C - f(\psi))\pi_\Psi(\psi) = \sum_{\psi'' \in \mathbb{X}} \left\{ p_{\Psi_{i+1}|\Psi_i}(\psi|\psi'') - \delta_{\psi,\psi''} \right\} R(\psi'') \qquad \text{(A.23)}$$

For notational convenience we abbreviate the right-hand side of Equation A.23 by $D(\psi)$.

For ψ with $q > 0$ and $\tilde{y} = 0$

$$\begin{aligned}
D(\psi) =&(\bar{\lambda}\bar{\mu} - 1)R(q, 0) + \bar{\lambda}\bar{\mu}R(q+1, 1) + \lambda\bar{\mu}R(q, 1) \\
&+ \lambda\bar{\mu}R(q-1, 0) \qquad \text{(A.24)}
\end{aligned}$$

For ψ with $q > 0$ and $\tilde{y} = 1$

$$\begin{aligned}
D(\psi) =&(\lambda\mu - 1)R(q, 1) + \bar{\lambda}\mu R(q+1, 1) + \bar{\lambda}\mu R(q, 0) \\
&+ \lambda\mu R(q-1, 0) \qquad \text{(A.25)}
\end{aligned}$$

And for ψ with $q = 0$ and $\tilde{y} = 0$

$$D(\psi) = -\lambda R(0, 0) + \bar{\lambda}R(1, 1) \qquad \text{(A.26)}$$

Adding Equations A.24 and A.25 yields

$$\bar{\lambda}R(q+1, 1) - \lambda R(q, 0) - \bar{\lambda}R(q, 1) + \lambda R(q-1, 0) \qquad \text{(A.27)}$$

We now sum Equation A.23 in two ways:

$$\sum_{\psi' \in \mathbb{X}: q' \leq q} D(\psi) = \bar{\lambda}R(q+1, 1) - \lambda R(q, 0) = M(q, 0) \qquad \text{(A.28)}$$

for $q \geq 0$ where we defined

$$M(q,0) = \sum_{\psi' \in \mathbb{X}: q' \leq q} (C - f(\psi))\pi_{\mathit{\Psi}}(\psi) \tag{A.29}$$

and

$$\begin{aligned} \sum_{\psi' \in \mathbb{X}: q' < q || q' = q, \tilde{y} = 1} D(\psi) = -\lambda\bar{\mu}R(q-1,0) - \lambda\bar{\mu}R(q,1) \\ +\bar{\lambda}\mu R(q,0) + \bar{\lambda}\mu R(q+1,1) = M(q,1) \end{aligned} \tag{A.30}$$

for $q \geq 1$ where we defined

$$M(q,1) = \sum_{\psi \in \mathbb{X}: q' < q || q' = q, \tilde{y} = 1} (C - f(\psi))\pi_{\mathit{\Psi}}(\psi) \tag{A.31}$$

We can combine Equations A.28 and A.30 to obtain the first-order recurrence

$$R(q+1,0) = \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu} R(q,0) + \tilde{M}(q) \tag{A.32}$$

for $q \geq 0$ where

$$\tilde{M}(q) = \frac{1}{\mu} M(q+1,1) - M(q+1,0) + \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu} M(q,0) \tag{A.33}$$

Note that

$$\begin{aligned} M(q,0) &= C\left(1 - \frac{\pi_Q(0)}{\bar{\mu}} \frac{\left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^{q+1}}{1 - \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}}\right) - \sum_{\psi \in \mathbb{X}: q' \leq q} f(\psi)\pi_{\mathit{\Psi}}(\psi) \\ &= \frac{\pi_Q(0)}{\bar{\mu}} \frac{\left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^{q+1}}{1 - \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}} \left(\mu \log(\frac{\mu}{\lambda}) + \bar{\mu}\log(\frac{\bar{\mu}}{\bar{\lambda}}) - C\right) \\ &= \frac{\bar{\lambda}}{\bar{\mu}} \left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^{q+1} \left(\mu \log(\frac{\mu}{\lambda}) + \bar{\mu}\log(\frac{\bar{\mu}}{\bar{\lambda}}) - C\right) \\ &= c_{M0} \left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^{q+1} \end{aligned} \tag{A.34}$$

and with this Equation A.33 becomes

$$\tilde{M}(q) = \frac{1}{\mu}M(q+1,1) \tag{A.35}$$

But

$$\begin{aligned} M(q+1,1) &= M(q,0) + (C - \log\frac{\mu}{\lambda})\pi_Q(0)\frac{\mu}{\bar{\mu}}\left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^{q+1} \\ &= \left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^{q+1}\left\{c_{M0} + (C - \log\frac{\mu}{\lambda})\pi_Q(0)\frac{\mu}{\bar{\mu}}\right\} \end{aligned}$$

So we obtain

$$\begin{aligned} \tilde{M}(q) &= \left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^{q+1}\left\{\frac{c_{M0}}{\mu} + (C - \log\frac{\mu}{\lambda})\frac{\pi_Q(0)}{\bar{\mu}}\right\} \\ &= c_{\tilde{M}}\left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^{q+1} \end{aligned} \tag{A.36}$$

and the generating function

$$\tilde{M}(z) = \sum_{q\geq 0}\tilde{M}(q)z^q = \frac{c_{\tilde{M}}\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}}{1 - z\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}} \tag{A.37}$$

We now define two new sequences a_q and b_q such that

$$R(q,0) = a_q + b_q R(0,0) \tag{A.38}$$

Clearly, $a_0 = 0$ and $b_0 = 1$. By substituting Equation A.38 into Equation A.32 we find that

$$a_{q+1} = \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}a_q + \tilde{M}(q) \tag{A.39}$$

and

$$b_{q+1} = \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}b_q \tag{A.40}$$

The solution to the recurrence b_q is obvious:

$$b_q = \left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^q \tag{A.41}$$

In order to obtain the solution to the recurrence b_q we employ the generating function method

$$\sum_{q\geq 0} a_{q+1} z^q = \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu} \sum_{q\geq 0} a_q z^q + \sum_{q\geq 0} \tilde{M}(q) z^q \tag{A.42}$$

and therefore

$$z^{-1} A(z) = \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu} A(z) + \tilde{M}(z) \tag{A.43}$$

We can now solve this equation for $A(z)$ to obtain

$$A(z) = \frac{\tilde{M}(z)}{z^{-1} - \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}} = c_{\tilde{M}} \frac{\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu} z}{\left(1 - \frac{\lambda\bar{\mu}}{\bar{\lambda}\mu} z\right)^2} \tag{A.44}$$

and the corresponding sequence

$$a_q = c_{\tilde{M}} q \left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^q \tag{A.45}$$

Finally

$$R(q,0) = a_q + b_q R(0,0) = c_{\tilde{M}} q \left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^q + \left(\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^q R(0,0) \tag{A.46}$$

Further

$$\sum_{\psi\in\mathbb{X}} r(\psi, i) = \sum_{\psi'\in\mathbb{X}} (f(\psi') - C) \pi_{\Psi}(\psi') \sum_{\psi\in\mathbb{X}} p_{\Psi_i|\Psi_0}(\psi|\psi') = 0 \tag{A.47}$$

and summing this equation in i yields

$$\sum_{\psi\in\mathbb{X}} R(\psi) = 0 \tag{A.48}$$

This result now allows us to compute $R(0,0)$: Using Equation A.28 we can write

$$\begin{aligned}\sum_{\psi\in\mathbb{X}} R(\psi) &= \sum_{q\geq 0} R(q,0) + \sum_{q\geq 0}\left(\frac{\lambda}{\bar{\lambda}}R(q,0) + \frac{1}{\bar{\lambda}}M(q,0)\right)\\ &= \frac{1}{\bar{\lambda}}\sum_{q\geq 0}\left(R(q,0) + M(q,0)\right) \qquad \text{(A.49)}\end{aligned}$$

Combining Equations A.38, A.48 and A.49 then yields

$$\begin{aligned}R(0,0) &= -\frac{\sum_{q\geq 0} a_q + \sum_{q\geq 0} M(q,0)}{\sum_{q\geq 0} b_q}\\ &= -c_{\tilde{M}}\frac{\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}}{1-\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}} - c_{M0}\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu} \qquad \text{(A.50)}\end{aligned}$$

Using the expressions for $R(j+1,1)$, $R(j,0)$ and $M(j,0)$ from Equations A.28, A.46 and A.34, respectively, we are eventually in a position to simplify the expression in Equation A.19:

$$\begin{aligned}&\sum_{i=0}^{\infty}\operatorname{Cov}(f(\mathit{\Psi}_0), f(\mathit{\Psi}_i)) = \log\frac{1}{\bar{\lambda}}R(0,0)\\ &+ \log\frac{\mu}{\lambda}\left(\frac{\lambda}{\bar{\lambda}}\sum_{q\geq 0}R(q,0) + \frac{1}{\bar{\lambda}}\sum_{q\geq 0}M(q,0)\right) + \log\frac{\bar{\mu}}{\bar{\lambda}}\sum_{q>0}R(q,0) \qquad \text{(A.51)}\end{aligned}$$

where

$$\begin{aligned}\sum_{q\geq 0} R(q,0) &= c_{\tilde{M}}\frac{\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}}{\left(1-\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}\right)^2} + \frac{R(0,0)}{1-\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}}\\ &= -c_{M0}\frac{\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}}{1-\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}} \qquad \text{(A.52)}\end{aligned}$$

and

$$\sum_{q\geq 0} M(q,0) = c_{M0}\frac{\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}}{1-\frac{\lambda\bar{\mu}}{\bar{\lambda}\mu}} \qquad \text{(A.53)}$$

The claim then follows. □

REFERENCES

[1] G. Perkin, "Expert report - Macondo," Engineering Partners International, Tech. Rep., August 2011.

[2] Oceaneering International, Inc., "ROV Services Rate Schedule Gulf of Mexico," January 2012. [Online]. Available: http://www.oceaneering.com/oceandocuments/rates/ROV-AmericasRegion.pdf

[3] T. Riedl, personal communication with more than 200 contacts in the oil and gas industry, March 2013.

[4] Monterey Bay Aquarium Research Institute, "2014 Vessel and Vehicle Rates," 2014. [Online]. Available: http://www.mbari.org/dmo/ship_rates.htm

[5] A. Reid, "ROV Market Prospects," in *ROV 2013 Conference.* Aberdeen, Scotland: Subsea UK, September 2013.

[6] Oceaneering International, Inc., "Annual report," March 2013.

[7] P. Newman and J. Westwood, "AUVs and ROVs - Global Market Prospects," in *UUV OI*, London, UK, March 2013.

[8] Department of the Navy, United States of America, "The Navy Unmanned Undersea Vehicle (UUV) Master Plan," November 2004.

[9] M. Burrows, *Elf Communications Antennas (IEE Electromagnetic Waves).* Institution of Engineering and Technology, 6 1978. [Online]. Available: http://amazon.com/o/ASIN/0906048001/

[10] E. H. Grant, T. J. Buchanan, and H. F. Cook, "Dielectric behavior of water at microwave frequencies," *The Journal of Chemical Physics*, vol. 26, no. 1, 1957.

[11] A. Defant, *Physical oceanography.* Pergamon Press, New York, 1961, vol. 1.

[12] L. N. Lieberman, *Transmission of energy within the sea, other electromagnetic radiation*, ser. The Sea. Interscience, New York, 1962, vol. 1.

[13] J. Levine and E. J. MacNichol, "Color vision in fishes," *Scie. Am.*, vol. 246, pp. 108–117, 1982.

[14] X. Che, I. Wells, G. Dickers, P. Kear, and X. Gong, "Re-evaluation of rf electromagnetic communication in underwater sensor networks," *Communications Magazine, IEEE*, vol. 48, no. 12, pp. 143–151, 2010.

[15] N. Farr, A. Bowen, J. Ware, C. Pontbriand, and M. Tivey, "An integrated, underwater optical/acoustic communications system," in *OCEANS 2010 IEEE-Sydney.* IEEE, 2010, pp. 1–6.

[16] B. Cochenour, L. Mullen, and A. Laux, "Spatial and temporal dispersion in high bandwidth underwater laser communication links," in *Military Communications Conference, 2008. MILCOM 2008. IEEE.* IEEE, 2008, pp. 1–7.

[17] C. D. Mobley, *Radiative transfer in the ocean*, ser. Encyclopedia of ocean sciences. Academic Press: Oxford, UK, 2001, vol. 4.

[18] A. Beer, "Bestimmung der Absorption des rothen Lichts in farbigen Flüssigkeiten," *Annalen der Physik*, vol. 162, no. 5, pp. 78–88, 1852.

[19] C. T. Roman, N. Jaworski, F. T. Short, S. Findlay, and R. S. Warren, "Estuaries of the northeastern united states: habitat and land use signatures," *Estuaries*, vol. 23, no. 6, pp. 743–764, 2000.

[20] S. Richards, A. Heathershaw, and P. Thorne, "The effect of suspended particulate matter on sound attenuation in seawater," *The Journal of the Acoustical Society of America*, vol. 100, no. 3, pp. 1447–1450, 1996.

[21] P. Thome, R. Soulsby, and P. Hardcastle, "Acoustic measurements of suspended sediment over sandwaves," *Coastal and Estuarine Studies*, vol. 40, pp. 335–349, 1992.

[22] N. R. Brown, T. G. Leighton, S. D. Richards, and A. D. Heathershaw, "Measurement of viscous sound absorption at 50–150 khz in a model turbid environment," *The Journal of the Acoustical Society of America*, vol. 104, no. 4, pp. 2114–2120, 1998.

[23] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 84–89, 2009.

[24] DOF Subsea, "Schilling Robotics UHD - Working Class ROV," May 2013. [Online]. Available: dof.no/Files/Files/Assets/ROVs/Schilling_UHD.pdf

[25] Kongsberg Maritime AS, "HUGIN AUV - Technical specifications," April 2013. [Online]. Available: http://www.km.kongsberg.com/ks/web/nokbg0397.nsf/AllWeb/76ABD1760DA9C064C1257B470029C7A5/$file/382309_hugin_product_specification.pdf?OpenElement

[26] N. Polmar, *The Naval Institute guide to the ships and aircraft of the US fleet.* Naval Institute Press, 2005.

[27] N. Polmar and K. J. Moore, *Cold War submarines: the design and construction of US and Soviet submarines.* Potomac Books, Inc., 2004.

[28] R. Takagi, "High speed railways: the last 10 years," *Japan Railway and Transport Review*, vol. 40, pp. 4–7, 2005.

[29] M. Johnson, L. Freitag, and M. Stojanovic, "Improved doppler tracking and correction for underwater acoustic communications," in *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, vol. 1, 1997, pp. 575–.

[30] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The whoi micro-modem: an acoustic communications and navigation system for multiple platforms," in *OCEANS, 2005. Proceedings of MTS/IEEE.* IEEE, 2005, pp. 1086–1092.

[31] M. Ewing and J. Worzel, "Propagation of sound in the ocean," *Memoir*, vol. 27, 1948.

[32] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *Oceanic Engineering, IEEE Journal of*, vol. 25, no. 1, pp. 4–27, 2000.

[33] M. Stojanovic, J. A. Catipovic, and J. G. Proakis, "Phase-coherent digital communications for underwater acoustic channels," *Oceanic Engineering, IEEE Journal of*, vol. 19, no. 1, pp. 100–111, 1994.

[34] I. Proakis, *Digital Communications.* McGraw-Hill, New York, 1989.

[35] S. Singh, S. E. Webster, L. Freitag, L. L. Whitcomb, K. Ball, J. Bailey, and C. Taylor, "Acoustic communication performance of the whoi micro-modem in sea trials of the nereus vehicle to 11,000 m depth," in *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges.* IEEE, 2009, pp. 1–6.

[36] S. D. Roy, T. McDonald, and J. V. Proakis, "High-rate communication for underwater acoustic channels using multiple transmitters and spacetime coding: Receiver structures and experimental results," *IEEE J. Oceanic Eng.*, vol. 32, pp. 663–688, July 2007.

[37] C. Pelekanakis, M. Stojanovic, and L. Freitag, "High rate acoustic link for underwater video transmission," in *OCEANS 2003. Proceedings*, vol. 2. IEEE, 2003, pp. 1091–1097.

[38] M. Stojanovic, "Low complexity OFDM detector for underwater acoustic channels," in *OCEANS 2006*. IEEE, 2006, pp. 1–6.

[39] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Multicarrier communication over underwater acoustic channels with nonuniform doppler shifts," *Oceanic Engineering, IEEE Journal of*, vol. 33, no. 2, pp. 198–209, 2008.

[40] B. Li, J. Huang, S. Zhou, K. Ball, M. Stojanovic, L. Freitag, and P. Willett, "Mimo-ofdm for high-rate underwater acoustic communications," *Oceanic Engineering, IEEE Journal of*, vol. 34, no. 4, pp. 634–644, 2009.

[41] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Non-uniform doppler compensation for zero-padded ofdm over fast-varying underwater acoustic channels," in *OCEANS 2007-Europe*. IEEE, 2007, pp. 1–6.

[42] B. Li, S. Zhou, M. Stojanovic, and L. Freitag, "Pilot-tone based zp-ofdm demodulation for an underwater acoustic channel," in *OCEANS 2006*. IEEE, 2006, pp. 1–5.

[43] A. C. Singer, J. K. Nelson, and S. S. Kozat, "Signal processing for underwater acoustic communications," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 90–96, 2009.

[44] R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman Lectures on Physics, boxed set: The New Millennium Edition*, slp ed. Basic Books, 1 2011. [Online]. Available: http://amazon.com/o/ASIN/0465023827/

[45] S. Rienstra and A. Hirschberg, "An introduction to acoustics," *Report IWDE*, pp. 92–06, 2001.

[46] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," *Fund. Math*, vol. 3, no. 1, pp. 133–181, 1922.

[47] M. Schulkin and H. W. Marsh, "Sound absorption in sea water," *The Journal of the Acoustical Society of America*, vol. 34, no. 6, pp. 864–865, 1962. [Online]. Available: http://link.aip.org/link/?JAS/34/864/1

[48] M. A. Ainslie and J. G. McColm, "A simplified formula for viscous and chemical absorption in sea water," *The Journal of the Acoustical Society of America*, vol. 103, no. 3, pp. 1671–1672, 1998.

[49] R. Francois and G. Garrison, "Sound absorption based on ocean measurements: Part i: Pure water and magnesium sulfate contributions," *The Journal of the Acoustical Society of America*, vol. 72, no. 3, pp. 896–907, 1982.

[50] R. Francois and G. Garrison, "Sound absorption based on ocean measurements. part ii: Boric acid contribution and equation for total absorption," *The Journal of the Acoustical Society of America*, vol. 72, no. 6, pp. 1879–1890, 1982.

[51] F. Fisher and V. Simmons, "Sound absorption in sea water," *The Journal of the Acoustical Society of America*, vol. 62, no. 3, pp. 558–564, 1977.

[52] W. H. Thorp, "Deep-ocean sound attenuation in the sub-and low-kilocycle-per-second region," *The Journal of the Acoustical Society of America*, vol. 38, no. 4, pp. 648–654, 2005.

[53] L. Berkhovskikh and Y. Lysanov, *Fundamentals of Ocean Acoustics.* New York, NY, USA: Springer, 1982.

[54] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: propagation models and statistical characterization," *IEEE Communications Magazine*, vol. 47, pp. 84–89, January 2009.

[55] G. M. Wenz, "Acoustic ambient noise in the ocean: spectra and sources," *The Journal of the Acoustical Society of America*, vol. 34, no. 12, pp. 1936–1956, 1962.

[56] R. F. Coates, *Underwater acoustic systems.* J. Wiley, 1989.

[57] R. J. Urick, "Ambient noise in the sea," DTIC Document, Tech. Rep., 1984.

[58] D. Chapman, "Surface-generated noise in shallow water: A model," *Proc. IOA*, vol. 9, no. Part 4, 1987.

[59] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 34–43, 2007.

[60] RESON A/S, "Hydrophone TC4014 - Datasheet," 2005. [Online]. Available: www.teledyne-reson.com/wp-content/uploads/2010/12/TC4014.pdf

[61] X. R. Lie and V. P. Jilkov, "A survey of maneuvering target tracking: Dynamic models," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, Orlando, FL, USA, 2000.

[62] C. Shalizi, “Stochastic processes (advanced probability ii),” lecture notes, 2007. [Online]. Available: http://www.stat.cmu.edu/~cshalizi/754/notes/

[63] H. J. McKean et al., “A winding problem for a resonator driven by a white noise,” *Journal of Mathematics of Kyoto University*, vol. 2, no. 2, pp. 227–235, 1962.

[64] J. Touboul and O. Faugeras, “A characterization of the first hitting time of double integral processes to curved boundaries,” *Advances in Applied Probability*, pp. 501–528, 2008.

[65] M. Goldman, “On the first passage of the integrated wiener process,” *The Annals of Mathematical Statistics*, pp. 2150–2155, 1971.

[66] A. Lachal, “Sur le premier instant de passage de l’intégrale du mouvement brownien,” in *Annales de l’institut Henri Poincaré (B) Probabilités et Statistiques*, vol. 27, no. 3. Gauthier-Villars, 1991, pp. 385–405.

[67] J. England, A. Hewitt, E. McHenry, and C. Vaillancourt, “Autonomous underwater vehicle final report,” Tech. Rep., May 2006. [Online]. Available: http://mickpeterson.org/Classes/Design/2005_6/Projects/AUV/Media/Final%20Report.pdf

[68] M. Fréchet, “Généralisation du théorème des probabilités totales,” *Fundamenta mathematicae*, vol. 25, no. 1, pp. 379–387, 1935.

[69] M. Fréchet, “Sur les tableaux de corrélation dont les marges sont données,” *Ann. Univ. Lyon Sect. A*, vol. 9, pp. 53–77, 1951.

[70] F. W. Olver, *NIST handbook of mathematical functions.* Cambridge University Press, 2010. [Online]. Available: http://dlmf.nist.gov

[71] H. Crabtree and M. Schuler, *The Anschutz Gyro-Compass and Gyroscope Engineering.* Watchmaker Publishing, 2003.

[72] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” in *Proceedings of the IEEE*, vol. 92, 2004, pp. 40–422.

[73] S. J. Julier and J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, vol. 3, 1997.

[74] R. V. Merwe, “Sigma-point kalman filters for probabilistic inference in dynamic state-space models,” Ph.D. dissertation, OGI School of Science & Engineering at Oregon Health & Science University, Portland, OR, 2004.

[75] H. W. Sorenson, “Least-squares estimation: From Gauss to Kalman,” *Spectrum, IEEE*, vol. 7, no. 7, pp. 63–68, 1970.

[76] T. Richardson and R. L. Urbanke, *Modern coding theory.* Cambridge University Press, 2008.

[77] D. J. Costello, J. Hagenauer, H. Imai, S. B. Wicker et al., “Error control coding,” in *Fundamentals and Applications, Printice Hall, Upper Saddle River, NJ.* Citeseer, 2004.

[78] J. Hagenauer, “The turbo principle: Tutorial introduction and state of the art,” in *Proc. International Symposium on Turbo Codes and Related Topics*, 1997, pp. 1–11.

[79] C. Douillard, M. Jézéquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, “Iterative correction of intersymbol interference: Turbo-equalization,” *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.

[80] M. Tüchler, R. Kötter, and A. C. Singer, “Turbo equalization: principles and new results,” *Communications, IEEE Transactions on*, vol. 50, no. 5, pp. 754–767, 2002.

[81] A. Glavieux, C. Laot, and J. Labat, “Turbo equalization over a frequency selective channel,” in *Proc. Int. Symp. Turbo Codes*, 1997, pp. 96–102.

[82] C. Laot, A. Glavieux, and J. Labat, “Turbo equalization: Adaptive equalization and channel decoding jointly optimized,” *IEEE Journal on Selected Areas in Communications*, pp. 229–235, 2001.

[83] M. Tüchler and A. C. Singer, “Turbo equalization: An overview,” *Information Theory, IEEE Transactions on*, vol. 57, no. 2, pp. 920–952, 2011.

[84] R. Kötter, A. C. Singer, and M. Tüchler, “Turbo Equalization,” *IEEE Signal Processing Mag*, vol. 21, pp. 67–80, 2004.

[85] J. W. Choi, T. Riedl, K. Kim, A. Singer, and J. Preisig, “Adaptive linear turbo equalization over doubly selective channels,” *Oceanic Engineering, IEEE Journal of*, vol. 36, no. 4, pp. 473 –489, oct. 2011.

[86] W. Murphy and W. Hereman, “Determination of a position in three dimensions using trilateration and approximate distances,” Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, MCS-95-07, Tech. Rep., 1995.

[87] S. Ikeda and J. H. Manton, “Capacity of a single spiking neuron channel,” *Neural Computation*, vol. 21, pp. 1714–1748, 2009.

[88] V. Anantharam and S. Verdu, “Bits through queues,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 4–18, 1996.

[89] R. Sundaresan and S. Verdu, “Capacity of queues via point-process channels,” *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2697–2709, 2006.

[90] T. P. Coleman, “A simple memoryless proof of the capacity of the exponential server timing channel,” in *Proceedings of the IEEE Information Theory Workshop (ITW)*, Volos, Greece, 2009.

[91] A. S. Bedekar and M. Azizoglu, “On the information-theoretic capacity of discrete-time queues,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 446–461, 1998.

[92] L. Weiss, “On the strong converse of the coding theorem for symmetric channels without memory,” *Quart. Appl. Math*, vol. 18, no. 3, 1960.

[93] R. L. Dobruschin, “Mathematical problems in the Shannon theory of optimal coding of information,” *Fourth Berkeley Symposium*, 1961.

[94] V. Strassen, “Asymptotische Abschätzungen in Shannon’s Informationstheorie,” *Trans. Third Prague Conf. Information Theory*, pp. 689–723, 1962.

[95] J. Wolfowitz, “The coding of messages subject to chance errors,” *Illinois J. Math.*, vol. 1, pp. 591–606, 1957.

[96] A. Feinstein, “A new basic theorem of information theory,” *IRE Trans. Inform. Theory*, vol. PGIT-4, pp. 2–22, 1954.

[97] W. Feller, “Generalization of a probability limit theorem of cramer,” *Transactions of The American Mathematical Society*, vol. 54, pp. 361–361, 1943.

[98] C. G. Esseen, “Fourier analysis of distribution functions,” *Acta Math.*, vol. 77, pp. 1–125, 1945.

[99] R. G. Gallager, “A simple derivation of the coding theorem and some applications,” *IEEE Transactions on Information Theory*, vol. 11, pp. 3–18, 1965.

[100] D. Baron, M. A. Khojastepour, and R. G. Baraniuk, “How Quickly Can We Approach Channel Capacity?” in *Proceedings of the 38th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 2004.

[101] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, pp. 2307–2359, 2010.

[102] W. Whitt, "Asymptotic formulas for markov processes with applications to simulation," *Operations Research*, vol. 40, pp. 279–291, 1992.

[103] J. G. Kemeny and J. L. Snell, *Finite Markov Chains.* Princeton, N.J., USA: Van-Nostrand, 1960.

[104] H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation.* Amsterdam, The Netherlands: North Holland, 1993, vol. 3.

[105] F. M. Spieksma and R. L. Tweedie, "Strengthening ergodicity to geometric ergodicity for Markov chains," *Stochastic Models*, vol. 10, pp. 45–75, 1994.

[106] S. Meyn, *Control Techniques for Complex Networks*, 1st ed. New York, NY, USA: Cambridge University Press, 2007.

[107] I. Kontoyiannis and S. Meyn, "Spectral theory and limit theorems for geometrically ergodic markov processes," in *the 2001 INFORMS Applied Probability Conference*, 2001, pp. 304–362.

[108] A. N. Tikhomirov, "On the convergence rate in the central limit theorem for weakly dependent random variables," *Theory of Probability and Its Applications*, vol. XXV, no. 4, 1980.

[109] W. Grassmann, "The asymptotic variance of a time average in a birth-death process," *Annals of Operations Research*, vol. 8, pp. 165–174, 1987.

[110] T. J. Riedl, "Finite block-length achievable rates for queuing timing channels," Tech. Rep., 2011, draft.

[111] "Arab spring: an interactive timeline of Middle East protests," http://www.guardian.co.uk/world/interactive/2011/mar/22/middle-east-protest-interactive-timeline.

[112] "Defeat Internet Censorship: Overview of Advanced Technologies and Products," http://www.internetfreedom.org/archive/Defeat_Internet_Censorship_White_Paper.pdf, Nov. 2007.

[113] C. S. Leberknight, M. Chiang, H. V. Poor, and F. Wong, "A taxonomy of Internet censorship and anti-censorship," http://www.princeton.edu/~chiangm/anticensorship.pdf, 2010.

[114] "Iran: Blogger died as a result of 'shock'," http://times247.com/articles/iran-blogger-died-as-a-result-of-shock, Nov. 2012.

[115] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger, “Infranet: Circumventing Web Censorship and Surveillance,” in *11th USENIX Security Symposium*, 2002, pp. 247–262.

[116] “DynaWeb,” http://www.dongtaiwang.com/home_en.php.

[117] S. Burnett, N. Feamster, and S. Vempala, “Chipping Away at Censorship Firewalls with User-Generated Content,” in *USENIX Security Symposium*, 2010, pp. 463–468.

[118] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The Second-Generation Onion Router,” in *USENIX Security Symposium*, 2004.

[119] J. Boyan, “The Anonymizer: Protecting User Privacy on the Web,” *Computer-Mediated Communication Magazine*, vol. 4, no. 9, Sep. 1997.

[120] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman, “Telex: Anticensorship in the Network Infrastructure,” in *20th Usenix Security Symposium*, 2011.

[121] A. Houmansadr, G. T. K. Nguyen, M. Caesar, and N. Borisov, “Cirripede: Circumvention Infrastructure using Router Redirection with Plausible Deniability,” in *ACM Conference on Computer and Communications Security*, 2011.

[122] P. Winter and S. Lindskog, “How China Is Blocking Tor,” http://arxiv.org/abs/1204.0447, Tech. Rep. Arxiv preprint, arXiv:1204.0447, 2012.

[123] “Italy Censors Proxy That Bypasses BTjunkie and Pirate Bay Block,” http://torrentfreak.com/italy-censors-proxy-that-bypasses-btjunkie-and-pirate-bay-block-110716.

[124] “Tor partially blocked in China,” https://blog.torproject.org/blog/tor-partially-blocked-china, Sep. 2007.

[125] R. Dingledine and N. Mathewson, “Design of a blocking-resistant anonymity system,” The Tor Project, Tech. Rep., Nov. 2006.

[126] D. McCoy, J. A. Morales, and K. Levchenko, “Proximax: A Measurement Based System for Proxies Dissemination,” in *Financial Cryptography and Data Security*, 2011.

[127] J. McLachlan and N. Hopper, “On the risks of serving whenever you surf: vulnerabilities in Tor’s blocking resistance design,” in *the 8th ACM workshop on Privacy in the electronic society*, 2009.

[128] N. Feamster, M. Balazinska, W. Wang, H. Balakrishnan, and D. Karger, “Thwarting Web Censorship with Untrusted Messenger Discovery,” in *the 3rd International Workshop on Privacy Enhancing Technologies*, 2003.

[129] M. Mahdian, “Fighting Censorship with Algorithms,” in *Fun with Algorithms*, ser. Lecture Notes in Computer Science, P. Boldi and L. Gargano, Eds. Springer, 2010, vol. 6099, pp. 296–306.

[130] Y. Sovran, A. Libonati, and J. Li, “Pass it on: Social networks stymie censors,” in *the 7th International Conference on Peer-to-peer Systems*, Feb. 2008.

[131] “Ultrasurf,” http://www.ultrareach.com.

[132] “Psiphon,” http://psiphon.ca/.

[133] J. Appelbaum, “Technical analysis of the Ultrasurf proxying software,” https://media.torproject.org/misc/2012-04-16-ultrasurf-analysis.pdf, The Tor Project, Tech. Rep., 2012.

[134] J. Appelbaum and N. Mathewson, “Pluggable transports for circumvention,” https://gitweb.torproject.org/torspec.git/blob/HEAD:/proposals/180-pluggable-transport.txt.

[135] N. Mathewson, “A simple obfuscating proxy,” https://www.torproject.org/projects/obfsproxy.html.en.

[136] “Pluggable Transports Roadmap,” https://www.cl.cam.ac.uk/~sjm217/papers/tor12pluggableroadmap.pdf.

[137] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh, “StegoTorus : A Camouflage Proxy for the Tor Anonymity System,” in *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[138] H. M. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, “SkypeMorph : Protocol Obfuscation for Tor Bridges,” in *ACM Conference on Computer and Communications Security*, 2012.

[139] T. Wilde, “Knock Knock Knockin on Bridges Doors,” https://blog.torproject.org/blog/knock-knock-knockin-bridges-doors, 2012.

[140] “Ten ways to discover Tor bridges,” https://blog.torproject.org/blog/research-problems-ten-ways-discover-tor-bridges.

[141] Q. Wang, X. Gong, G. T. K. Nguyen, A. Houmansadr, and N. Borisov, "CensorSpoofer: Asymmetric Communication with IP Spoofing for Censorship-ResistantWeb Browsing," in *ACM Conference on Computer and Communications Security*, 2012.

[142] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper, "Routing Around Decoys," in *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[143] S. Baset and H. Schulzrinne, "Skype relay calls: Measurements and Experiments," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2008.

[144] S. A. Baset and H. G. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," in *the 25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006, pp. 1–11.

[145] S. Guha, N. Daswani, and R. Jain, "An Experimental Study of the Skype Peer-to-Peer VoIP System," in *the 5th International workshop on Peer-To-Peer Systems*, 2006.

[146] "VoIP growing statistics," http://www.sipnology.com/company/20-voip-growing-statistics.

[147] "VoIP Penetration Forecast to Reach 79% of US Businesses by 2013," http://www.fiercewireless.com/press-releases/voip-penetration-forecast-reach-79-us-businesses-2013, 2010.

[148] "The Importance of VoIP," http://ezinearticles.com/?The-Importance-of-VoIP&id=4278231.

[149] A. Banerjee, D. J. Costello, Jr., T. E. Fuja, and P. C. Massey, "Bit Interleaved Coded Modulation Using Multiple Turbo Codes," in *IEEE International Symposium on Information Theory*, 2002, p. 443.

[150] C. Jones, "The Importance Of VoIP And A Business Continuity Plan For Business Survival," http://www.tech2date.com/the-importance-of-voip-and-a-business-continuity-plan-for-business-survival.html, 2011.

[151] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, June 1999.

[152] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones, "RFC 1928: SOCKS Protocol Version 5," Apr. 1996.

[153] "Skype grows FY revenues 20%, reaches 663 mln users," http://www.telecompaper.com/news/skype-grows-fy-revenues-20-reaches-663-mln-users, 2011.

[154] H. Xie and Y. R. Yang, "A Measurement-based Study of the Skype Peer-to-Peer VoIP Performance," in *6th International workshop on Peer-To-Peer Systems*, 2007.

[155] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johanston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261: SIP Session Initiation Protocol," June 2002.

[156] H. Schulzrinne, S. L. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," Internet RFC 3550, July 2003.

[157] M. Baugher, E. Carrara, D. A. McGrew, M. Naslund, and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," Internet RFC 3711, Mar. 2004.

[158] J. Hautakorpi, G. Camarillo, R. Penfield, A. Hawrylyshen, and M. Bhatia, "Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments," Internet RFC 5853, Apr. 2010.

[159] "How to Configure SIP and NAT," http://www.linuxjournal.com/article/9399.

[160] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer, "Decoy Routing : Toward Unblockable Internet Communication," in *1st USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2011.

[161] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Groups," in *the 3rd ACM Conference on Computer and Communications Security*. ACM, Mar. 1996, pp. 31–37.

[162] "VoIP codecs," http://www.en.voipforo.com/codec/codecs.php.

[163] "Google talk call signaling," https://developers.google.com/talk/call_signaling#Supported_Media_Types.

[164] S. Jensen, K. Vos, and K. Soerensen, "SILK speech codec," Working Draft, IETF Secretariat, Fremont, CA, USA, Tech. Rep. draft-vos-silk-02.txt, Sep. 2010.

[165] D. Adami, C. Callegari, and S. Giordano, "A Real-Time Algorithm for Skype Traffic Detection and Classification," in *the 9th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking*, 2009.

[166] "Tor BridgeDB," https://gitweb.torproject.org/bridgedb.git/tree.

[167] I. Cooper and J. Dilley, "Known HTTP Proxy/Caching Problems," Internet RFC 3143, June 2001.

[168] I. Clarke, T. W. Hong, S. G. Miller, O. Sandberg, and B. Wiley, "Protecting Free Expression Online with Freenet," *IEEE Internet Computing*, vol. 6, no. 1, pp. 40–49, 2002.

[169] D. Fifield, N. Hardison, J. Ellithrope, E. Stark, R. Dingledine, D. Boneh, and P. Porras, "Evading Censorship with Browser-Based Proxies," in *Privacy Enhancing Technologies Symposium (PETS)*, 2012.

[170] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis," in *Network and Distributed System Security Symposium.* The Internet Society, Feb. 2009.

[171] W. Mazurczyk, P. Szaga, and K. Szczypiorski, "Using Transcoding for Hidden Communication in IP Telephony," Arxiv preprint, arXiv:1111.1250, Tech. Rep., 2011.

[172] A. White, A. Matthews, K. Snow, and F. Monrose, "Phonotactic reconstruction of encrypted VoIP conversations: Hookt on fon-iks," in *IEEE Symposium on Security and Privacy*, 2011, pp. 3–18.

[173] J. Knockel, J. R. Crandall, and J. Saia, "Three Researchers, Five Conjectures: An Empirical Analysis of TOM-Skype Censorship and Surveillance," in *the 1st USENIX Workshop on Free and Open Communications on the Internet*, 2011.

[174] "Dynamic Internet Technology Inc. Alleges Skype Redirects Users in China to Censorware Version- Ten Days After Users Are Able To Download Freegate Software Through Skype," http://www.businesswire.com/news/home/20070924006377/en/Dynamic-Internet-Technology-Alleges-Skype-Redirects-Users, 2007.

[175] "Surveillance of Skype messages found in China," International Herald Tribune, Tech. Rep., 2008.

[176] T. Claburn, "Skype Defends VoIP IM Monitoring In China," http://www.informationweek.com/news/210605439, 2008.

[177] "Skype says texts are censored by China," http://www.ft.com/cms/s/2/875630d4-cef9-11da-925d-0000779e2340.html#axzz1tpnzbzkm, Mar. 2009.

[178] N. Borisov, "Computational puzzles as sybil defenses," in *IEEE International Conference on Peer-to-Peer Computing*, 2006, pp. 171–176.