CODING FOR PASSIVE RFID COMMUNICATION

CODING FOR PASSIVE RFID COMMUNICATION

Guang Yang

DISSERTATION FOR
THE DEGREE OF PHILOSOPHIAE DOCTOR

THE SELMER CENTER
DEPARTMENT OF INFORMATICS
UNIVERSITY OF BERGEN
NORWAY

MARCH 2012

谢谢爸妈

Abstract

This dissertation elaborates on channel coding for reliable communication in passive RFID systems. RFID applications have been developed and used widely. Since a passive RFID tag has no power requirements, passive RFID has received considerable attention for application to sensor networks, access management, etc.

A passive RFID system transfers energy together with information by means of inductive coupling. Coding schemes design for inductively coupled channels is the main task of this work. Due to the properties of inductive coupling, the communication over the inductively coupled channel has synchronization loss problems in addition to other classical channel errors. We therefore design codes that consider synchronization loss, energy transfer, transmission rate, and complexity of encoding and decoding. Because of the different properties of reader and tag, the coding schemes for a passive RFID system are designed differently for the two directions between a reader and a tag.

In this dissertation, electronic circuits as infrastructure for the physical layer of the system are described. Modulation codes, error control codes and constrained codes, including their algebraic and structural properties, related algorithms, and techniques are addressed. Code combination and code power spectra are also considered as important issues in the code design that improve the ability of error detection and correction.

Acknowledgements

The PhD study in Bergen has been challenging, but I have been very lucky receiving invaluable support and encouragement along the way. This dissertation would not have been possible without anyone. It is impossible to name all, but I wish to use this opportunity to thank a few.

I would like to express my gratitude to my supervisors: Øyvind Ytrehus for his knowledge, inspiration, and patience, Eirik Rosnes who always leads a losing girl out from mazes, and Ángela I. Barbero. I have learned a lot from all of you, not only the coding theory but also research attitude and spirit.

I am very thankful to Irina Naydenova and Matthew G. Parker for giving the beginning lectures on coding theory that opened the door for me to here. I appreciate Matthew for proofreading. Special thanks to Jinquan Luo, Chunlei Li, and Nian Li for all the help in algebra and probability theory, and Saerda Halifu for help with programs running on the clusters of BCCS.

I would also like to thank the administrative staff at the Dept. of Informatics: Marta Lopez, Tor Bastiansen, Liljan Myhr, Inger Nilsen, and Steinar Heldal for help with practical issues.

Contents

LIST OF FIGURES

Introduction

1 RFID Systems

In addition to the industrial setting, identification systems are budgeted for cheap applications. The barcode mechanism is a cheap solution, but with very low storage capacity, and it can not be reprogrammed. It is therefore found to be inadequate in an increasing number of scenarios [1]. Together with a contactless reader, a data carrying device having larger data storage capacity and with the ability to be reprogrammed is a more flexible solution.

Radio frequency identification (RFID) is a wireless communication technology that has been widely applied for identification and tracking. It is convenient to have RFID applications in our daily life. In logistics or transportation, RFID technology has been employed for cargo management; in supermarkets, RFID tags are used for tracking products; and in public transport, RFID tickets have replaced conventional tickets. Some credit card companies integrate special RFID cards into mobile phones, so that users can access their bank accounts and pay by phones.

1.1 Reader and Tag

Essential components of an RFID system are an interrogator (reader) and a transponder (tag). A tag stores information and is attached to an item, while a reader is a device that can recognize the presence of tags for the purpose of identification and tracking. Both the reader and the tag contain radio frequency modules that are integrated as semiconductor chips for signal processing, and antennas for transmitting and receiving signals.

The shape of a tag can be variable, such as a paper label, small plastic bar, etc. They must be attachable and able to communicate information over a radio frequency with a certain reader. The tag is structurally simpler than the reader with optional functions like writing once or many times, anti-collision, encryption, and security protocols. Some RFID readers apply additional middleware to deal with collected information. In some advanced systems, readers are able to communicate with networks of other devices through a variety of interfaces, like built-in wireless Ethernet, Bluetooth, and even ZigBee [2].

1.2 Power Supply

An RFID system can be classified by the power supply of the tag. When tags have their own power source, i.e., batteries that supply energy for signal processing and radio wave broadcasting, then the tag/reader pair is referred to as an active RFID system. Conversely, when tags do not have their own independent power source, then it is a passive RFID system, in which tags have to obtain power from the reader. There are also semi-passive RFID systems, in which tags have their own power source for signal processing but have to take power from readers for radio wave broadcasting. Passive RFID systems employ inductive coupling to transfer energy from reader to tag via the electromagnetic field generated by their inductively coupled circuits. See Section 2 for details.

1.3 RFID Networks

RFID system networks are built as "master-slave" architectures. Depending on its purpose, an RFID system can be organized as a single-tag-single-reader (STSR), single-tag-multiple-readers (STMR), multiple-tags-single-reader (MTSR), or multiple-tags-multiple-readers (MTMR) network. There are fully developed RFID networks such as *real-time locating systems* and *RFID sensor systems* [3]. The most important RFID network is the EPCglobal network that provides real-time data about individual items as they move through a global supply chain. The EPCglobal network was proposed by EPCglobal that released worldwide standards for RFID and specified usage of the Internet for data sharing within the EPCglobal network, and also provided security features such as authentication and authorization of the EPCglobal network [2].

2 Inductive Coupling

This section explains how a passive RFID tag gets power from a reader and powers up its semiconductor chip for signal processing and sending of identity information back to the reader by means of inductive coupling. Due to the properties of the inductively coupled channel, signal synchronization is a critical problem that we consider. Thus, bit-shift channels that model synchronization loss are introduced in Section 4. The code design for bit-shift channels is discussed in Section 6.

2.1 Electromagnetic Field

As introduced in [4], within a closed circuit, a steady current I_1 produces a magnetic field $\mathbf{B}_1$. Fig. 1 illustrates this phenomenon. This magnetic field is measured by the *magnetic flux density*, which is proportional to the current I_1, as the *Biot-Savart law* describes

$$\mathbf{B}_1 = \frac{\mu_0}{4\pi} I_1 \int \frac{\mathrm{d}\mathbf{s} \times \mathbf{r}}{r^3} \tag{1}$$

where μ_0 is the *magnetic constant*, $\mathrm{d}\mathbf{s}$ is the short segment of the wire, i.e., the *differential* element of the wire in the direction of the conventional current, and $\mathbf{r}$ is a vector that points from the short segment of current to the observation point where we are to compute the magnetic field. The magnetic field at the center of Circuit 1 is: $\mathbf{B}_1 = \frac{\mu_0 I_1}{2R}$, where R is the radius of Circuit 1. Additionally, the direction of $\mathbf{B}_1$ is given by the *right-hand rule*: by wrapping the right hand around a solenoid with fingers in the direction of the conventional current, the thumb points in the direction of the magnetic field that is generated by the current.

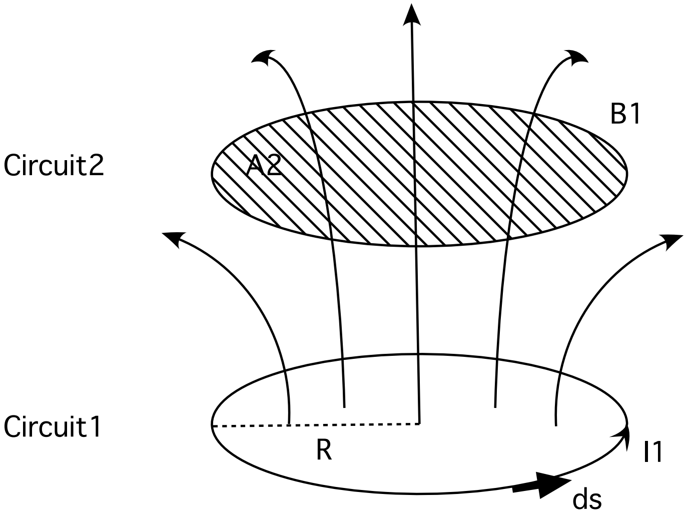

Fig. 1: *Magnetic field of inductive coupling circuits*

Any magnetic field is artificially visualized by magnetic "needles" that determine the direction and strength of the field. Thus, the "number" of magnetic needles that pass a certain surface indicates the magnetic flux Φ, as shown in Fig. 1. The flux through Circuit 2 is calculated by

$$\Phi_2 = \int \mathbf{B}_1 \cdot \mathrm{d}A_2 \tag{2}$$

where A_2 is the area of Circuit 2. From Eq. (1) and Eq. (2), it is clear that Φ_2 is a function of I_1. As stated by *Faraday's law*, the *change* of I_1 induces an electromotive force (EMF) in the nearby Circuit 2 as

$$\varepsilon = -\frac{\mathrm{d}\Phi_2}{\mathrm{d}t}. \tag{3}$$

The negative sign in Eq. (3) indicates that the induced EMF is the result of balancing the change of flux Φ_2. It is also stated by *Lenz's law* that the induced current is always in a direction resisting this change.

Since Φ_2 depends on I_1, this can be alternately expressed as

$$\Phi_2 = M_{21} I_1. \tag{4}$$

M_{21} is known as *mutual inductance*, which depends on the geometrical placement of the two circuits. Likewise, the induced current I_2 also produces a magnetic field $\mathbf{B}_2$, whose flux Φ_1 passing through Circuit 1 is proportional to the current I_2, as $\Phi_1 = M_{12} I_2$. Additionally, the flux changes within each circuit also induce an EMF in the circuit itself, which is called *self-inductance*.

2.2 Inductive Coupling Circuits

A passive RFID system is a wireless communication system that employs a near-field coupling technique. The inductive coupling mechanism makes possible the transfer of energy from one circuit to another circuit in the vicinity without a wired connection.

A simplified passive RFID system contains one reader and one tag, and a coupling of conductors in circuits as Fig. 2 shows [5]. Two inductors L_r and L_t are two coils winded with conducting wires. One single loop of the coil can be regarded as one circuit in Fig. 1. Therefore, the number of loops and the cross-section area of the coil indicate the inductance ability of the inductor. The alternating current (AC) of the reader's circuit induces a magnetic field, which causes magnetic flux to change in the coils of both the reader and the tag's inductors. This change in magnetic flux generates EMF for both circuits, which is the power source for the tag. Thus, the signal transmission from the

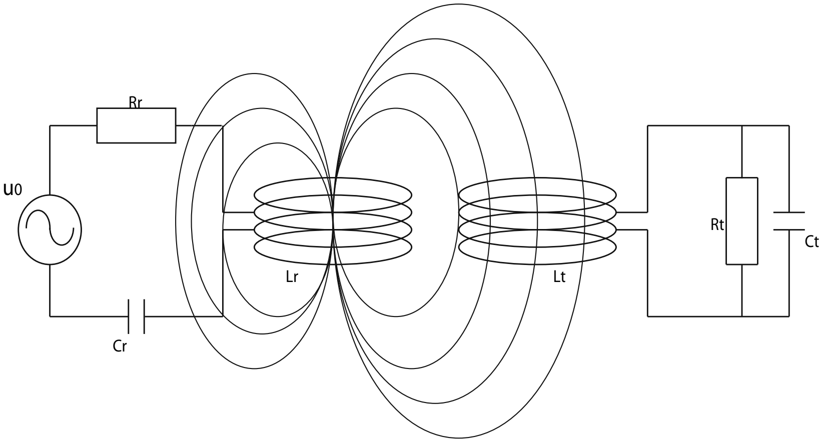

Fig. 2: *Inductive coupling circuit*

reader to the tag also transmits energy. In the same way, current in the tag circuits also produces a magnetic field which causes changes of magnetic flux in the coils of both the tag and the reader's inductors. *Mutual inductance* can be calculated as

$$M_{rt} = \frac{\mu_0 \cdot N_r \cdot R_r^2 \cdot N_t \cdot R_t^2 \cdot \pi}{2\sqrt{\left(R_r^2 + x^2\right)^3}} \tag{5}$$

and

$$M_{tr} = \frac{\mu_0 \cdot N_t \cdot R_t^2 \cdot N_r \cdot R_r^2 \cdot \pi}{2\sqrt{\left(R_t^2 + x^2\right)^3}} \tag{6}$$

where μ_0 is the magnetic field constant representing magnetic conductivity in a vacuum; N_r, N_t are the number of windings of the coil of the reader and the tag; and R_r, R_t are the coil radius of the reader and the tag, respectively. Assuming that the coil is in the x-axis, the strength of field can be calculated along the x-axis. The x in the equations is the distance from the center of the coil in the direction of the x-axis. If the x-axis of a reader's coil and a tag's coil lie in the same plane, $M_{rt} = M_{tr} = M$ when $R_r = R_t$ [1]. It is clear that the mutual inductance is a function of inter-coil separation of the reader and the tag when other parameters of the coupled circuits are fixed, as is the case in many practical situations.

In the coupled circuits, an AC power source is supplied at the reader's side: $U_0 = V_0 \sin(\omega_0 t)$. The impedance of the reader and the tag circuits are

$$Z_r = R_r + j\omega L_r + \frac{1}{j\omega C_r} \quad \text{and} \quad Z_t = R_t + j\omega L_t + \frac{R_t}{1 + j\omega C_t R_t} \tag{7}$$

respectively. After coupling, we get

$$Z_r' = Z_r - \frac{\omega^2 M^2}{Z_t} \quad \text{and} \quad Z_t' = Z_t - \frac{\omega^2 M^2}{Z_r} \tag{8}$$

respectively. The source voltage of the tag is

$$V_t = \frac{U_0}{Z_r} j\omega M \tag{9}$$

and the output voltage of the tag is

$$V_{\text{out}} = \frac{V_t}{Z_t'} \frac{1}{j\omega C_t}. \tag{10}$$

2.3 Resonance

Electromagnetic resonance is a phenomenon in an LC circuit (L stands for inductor, while C stands for capacitor), where the charged capacitor releases a current that through the inductor builds a magnetic field to induce EMF. Consequently, this EMF charges the capacitor in the opposite polarity, and this process repeats with an angular frequency of $\omega_1 = \frac{1}{\sqrt{LC}}$. The electrical energy is gradually dissipated due to the presence of other resistors in the circuit (the left circuit in Fig. 3). However, if we add an additional AC power source (the right circuit in Fig. 3), this repeated process will be steady without energy dissipation.

This process can be illustrated by the following equation

$$U(t) = i(t) \left(R + j\omega_0 L + \frac{1}{j\omega_0 C} \right) \tag{11}$$

where ω_0 is the angular frequency of the AC power source. From Eq. (11), it is found that inductance and capacitance will cancel each other if $\omega_0 = \omega_1$. Then the imaginary part of Eq. (11) will be zero. By this means, the impedance reaches a minimum value.

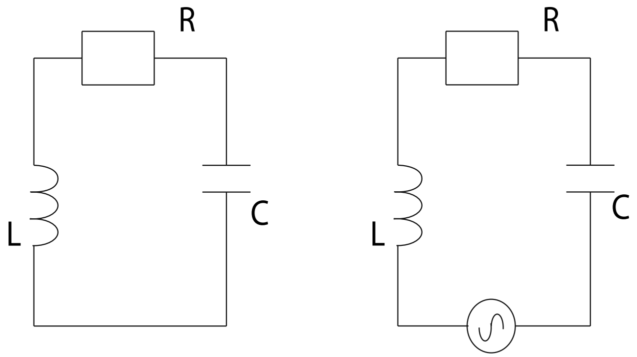

Fig. 3: *Resonance circuits*

If the frequency of the circuit equals that of the AC power source, this circuit is *tuned*. In a passive RFID system, the reader circuit generates maximum EMF when it is tuned. This can be achieved by associating the inductance and capacitance of the reader's circuit to a specified frequency which is the same as that of the power source [1]. According to this theory, a tag of a passive RFID system obtains maximum power from a reader when it is tuned.

2.4 Capacitors in the System

Consider the charging and discharging processes of a capacitor in a given circuit. The charging process can be represented as

$$V_0(t) = i(t)R + V_c \tag{12}$$

where V_0 is the sum voltage and R is the sum of other conductors in the circuit. Thus, $i(t)R$ is the voltage distributed on other conductors and V_c is the voltage of the capacitor. Let C denote the capacitance of the capacitor. Then, we get

$$V_0 = \left(C\frac{\mathrm{d}V_c}{\mathrm{d}t}\right) R + V_c, \tag{13}$$

$$\frac{\mathrm{d}V_c}{V_0 - V_c} = \frac{\mathrm{d}t}{RC}. \tag{14}$$

By integration:

$$\frac{t}{RC} = -\ln V_0 - V_c + k, \tag{15}$$

$$V_c = V_0(1 - e^{-\frac{t}{RC}}) \tag{16}$$

where $k = \ln V_0$. The discharging process is similar. These two processes of a capacitor exhibit exponential behaviors as

$$V(t)_{\text{charging}} = V_0(1 - e^{-\frac{t}{\tau}}), \tag{17}$$

$$V(t)_{\text{discharging}} = V_0 e^{-\frac{t}{\tau}} \tag{18}$$

where V_0 is the initial voltage of the circuit, τ is a *time constant* which is the product of the resistance and the capacitance. From Eq. (17) and Eq. (18), it can be seen that capacitors should be charged and discharged fully. In a data sending and receiving session, this relates to operating frequency. Only when the time period of one signal (0 or 1) is long enough ($t >> RC$) can the capacitor be fully charged or discharged, and signals can be constructed or reconstructed with little or no distortion.

3 Modulation

Data transmission in an information communication system generally involves channel encoding, decoding, modulation, and demodulation procedures. In an RFID system, both reader and tag have encoder/decoder, and modulator/demodulator function chips [2]. Reader and tag are either an information source or destination, as shown in Fig. 4. The channel is a transmission medium measured by bandwidth (in Hz) or its data rate. It is susceptible to thermal noise, delay, interference, etc.

In this section, modulation and demodulation schemes are introduced. Because signals from a tag to a reader are extremely weak compared with the signal from the reader itself, load modulation is applied by the tag when it responds. Load modulation using a subcarrier and a subharmonic of the reader's transmission frequency is explained in this section. Multiple access control is addressed in the context of RFID network communications.

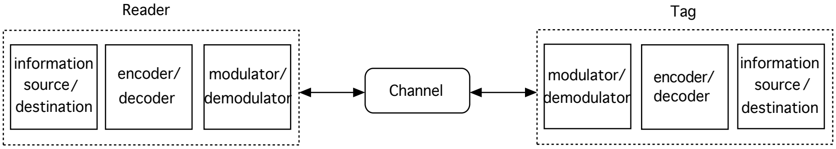

Fig. 4: *An RFID system*

3.1 AM and BPSK

Data transformation from discrete symbols to electromagnetic waveforms for transmission purposes is referred to as modulation, and the inverse procedure is called demodulation. Modulation schemes are designed for analog signals or digital signals. Signals are influenced by *power*, *frequency*, and *phase position*, thus analog and digital modulation schemes are designed with respect to these parameters. Classical analog modulation schemes include: amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM), while digital modulation schemes are: amplitude shift keying (ASK), frequency shift keying (FSK), phase shift keying (PSK), and other derived schemes [6]. Due to simplicity of implementation, we consider conventional AM and binary PSK (BPSK) in this section.

AM uses the amplitude of the message signal $m(t)$ to modulate the carrier signal $c(t)$. The transmitted signal is expressed as

$$u(t) = A_c[1 + m(t)]\cos(2\pi f_c + \phi_c). \tag{19}$$

The message signal is expressed as $m(t) = a\cos(2\pi f_m t)$, where f_m is the baseband frequency and a is the amplitude of the message signal. A_c, f_c, and ϕ_c are the amplitude, frequency, and phase of the carrier signal, respectively. A modulated signal 1 1 0 is shown in Fig. 5.

With the use of this modulation scheme, the carrier frequency can be chosen to avoid interference between the same or close frequencies. Since $A_c[1 + m(t)]$ is always positive, the demodulation process is facilitated by an envelope detector that crops the waveform of the received signal and reconstructs the original message signal from its edge. This is very simple to implement.

On the other hand, a digital signal with binary states '0' and '1' that is converted into bipolar phase states, is called a BPSK modulated signal. That is, for each symbol, $v \in \{0, 1\}$ is mapped to x according to

$$x = \sqrt{E_s}(1 - 2v) \tag{20}$$

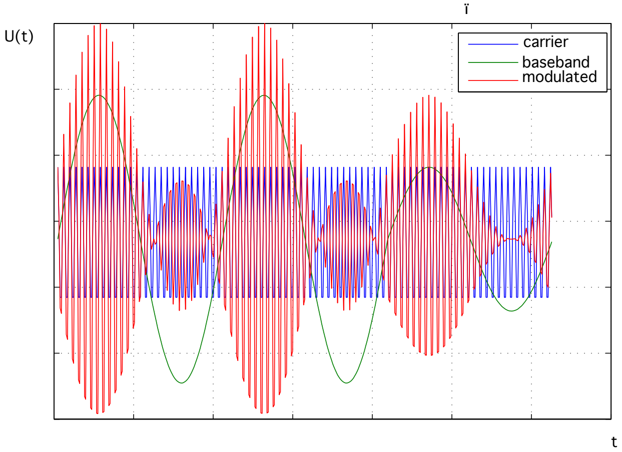

Fig. 5: *Amplitude modulation*

where E_s is average energy per symbol.

3.2 Load Modulation

As introduced in the previous section, the alternating magnetic fields between reader and tag are generated mutually. From Eq. (8), it is known that: when two circuits are coupled, the impedance of the reader circuit depends on the tag's feedback impedance Z_t, which is the *transformed impedance*. Therefore, changing the impedance of the tag may cause a detectable change in the reader's circuit. This property is utilized for data transmission from a tag to a reader. By changing the impedance of a tag to reply, the reader must be able to recognize the signals.

To vary a tag's impedance, we can either add an extra parallel resistor or capacitor to the tag's circuit, and use a switch to control it [1]. As Fig. 6 shows, a parallel resistor R_{mod} is added to the circuit of Tag 1. As the switcher S is alternately on and off, Z_t' switches between $Z_t(R_t)$ and $Z_t(R_t||R_{\text{mod}})$. In Tag 2's circuit, a parallel capacitor is added. When switcher S is alternately on and off, Z_t' switches between $Z_t(C_t)$ and $Z_t(C_t + C_{\text{mod}})$. These two methods are called *ohmic load modulation* and *capacitive load modulation*, respectively.

For easy detection at the reader's side, *sideband* AM is used. If the tag's switcher turns on and off by a frequency f_s, which is high enough to be distinguished from the frequency f_{reader} of the reader's circuit,

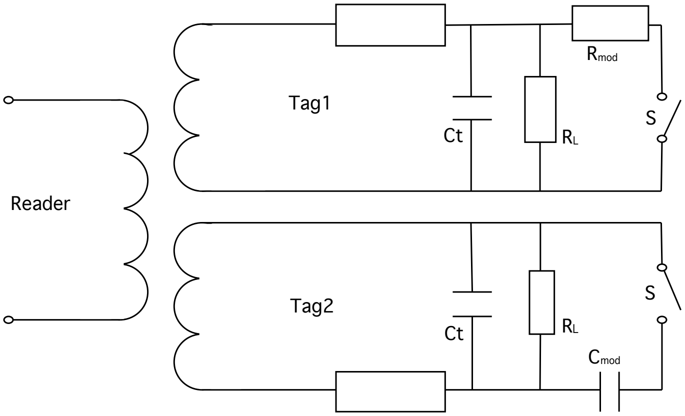

Fig. 6: *Load modulation*

then the reader could use a bandpass filter to obtain certain signals with frequencies that are different from the frequency generated by the reader itself, and demodulate them. Typical subcarrier frequencies are: 212kHz, 424kHz (ISO/IEC15693), and 848kHz (ISO/IEC14443) [1].

Another method is to use a *subharmonic* that divides the reader's operating frequency by an integer to produce a new frequency for the tag's reply. For example, the reader could use a frequency of 128kHz, and the tag could use a subharmonic to respond at 64kHz, being one half of the reader's operating frequency [2].

3.3 Multiple Access Control

In previous sections, the principle of duplex data transmission from a single reader to a single tag in the physical layer was introduced. However, in most of the practical situations, communication to several tags in an interrogation zone of a reader must be considered. How to manage communication from individual tags to the same reader within a channel without transmission collisions and mutual interference, as usually happens when individual tags simultaneously transfer data to a reader, is the main issue dealt with in the *Media Access Control/Multiple-Access Control (MAC)* layer. MAC usually involves addressing each participant tag and assigning individual channels to them. There are two main mechanisms for RFID systems: *frequency domain multiple access (FDMA)* and *Aloha*. Both of them use *reader talk first*, which means that all conversations are initialized by the reader. Other solutions include *code division multiple access (CDMA)* and *space division multiple access (SDMA)* [1].

3.3.1 FDMA

The reader initiates a conversation with operating frequency f_0; tags receive the initial command and reply with different frequencies f_i, on channels that are orthogonal sub-carriers of f_0 in order to avoid inter-symbol interference (ISI) [7]. The reply frequencies can be either assigned by the reader when it starts each conversation, or randomly selected by tags [1]. In this way, each tag has its independent channel connection with the reader. FDMA can be combined with *time division multiple access (TDMA)* as *FTDMA*, to provide good performance for simultaneous reply of multiple tags on different channels [8].

3.3.2 Aloha

Aloha is a network solution that allows every user in the network to transmit data whenever required. Users can know whether their frames are successfully received or not by feedback. If a user is aware that one of its data frames is unsuccessfully sent, it will wait for a random time and retransmit the data frame until it succeeds. This protocol is called *pure Aloha*. Another Aloha protocol is *slotted Aloha*, which divides time into discrete intervals for each data frame [9].

- Pure Aloha of a RFID system:
  The reader initially energizes all tags, then tags randomly reply. The reader is able to detect collisions among all the replies and report the received replies. The tags that have not been successfully replied to, can discover this from the reader's feedback and reply again. This mechanism is suitable for small number of tags at one time.

- Slotted Aloha of a RFID system:
  The reader initially energizes all tags and generates a set of discrete numbers for all the tags. Each tag randomly selects a number from the set corresponding to a time slot. The reader then deals with each tag by its number. Each tag is kept in an active state until its reply is successfully received, at which point it will exit this term of conversation.

To predict throughput of an Aloha network, it is assumed that there is a uniform frame size for all frames in the network and all transmission attempts, including new frames and old frames that are not successfully delivered. Throughputs of *pure Aloha* and *slotted Aloha* are: Ge^{-2G} and Ge^{-G}, respectively, where G is the mean of k transmission attempts according to a *Poisson distribution*. In an actual RFID system, the population of tags is finite, and the throughput is $G(1 - G/N)^{N-1}$, where N is the number of tags [9].

3.3.3 CDMA

Using *spread-spectrum* technology (*PN* sequences) and orthogonal coding schemes, each participant can communicate with others in the same channel. The CDMA techniques can be used to decode tag replies in collisions, especially in a system with large tag populations that requires fast identification [10].

3.3.4 SDMA

By rearranging communication range and distribution of antennas, or by implementing *a smart antenna* of a reader to specialized interrogation zones for each tag, communication between the reader and each tag can be separated. But there are some limitations in the location and size of the system, and smart antennas are costly [1].

4 Information Theory

Information theory is used to explore and solve the problems of data compression and data transmission, and study a trade-off between compression and transmission rates. These two problems present a case of duality: data compression aims at eliminating all the redundancy in the data to obtain the most compressed form possible for recording or storage, while data transmission involves addition of redundancy to combat errors that are induced by a transmission channel [11]. The foundation for information theory was laid in 1948 by Shannon in his overwhelming paper [12]. Shannon proved that the probability of error in transmission could be arbitrarily close to zero below a certain communication rate, which is defined as the *channel capacity*. Moreover, below a certain *irreducible complexity*, a signal can not be compressed further. This section discusses notations used in information theory and introduces the definition of channel capacity. Different channel models, which are conceptually used to describe communication channels in the real-world for investigating communication under given conditions, are discussed because capacities are critical for channel coding. Among them, the bit-shift channel is often used in the context of magnetic and optical recording, where model bits are lost and gained between a source and a receiver at unknown positions. The channel describes properties observed in inductively coupled channels too. Variations of the traditional bit-shift channel are proposed in this section as well.

4.1 Notations in Information Theory

Shannon defined *entropy* as follows: Let X be a discrete random variable with alphabet Λ. For any probability distribution $p(x) = P\{X = x\}$, $x \in \Lambda$, the *entropy* $H(X)$ of X is

$$H(X) = -\sum_{x\in\Lambda} p(x)\log p(x). \tag{21}$$

Entropy is a measure of the uncertainty in a signal consisting of random variables, it is a function of the distribution of the variables. If the base of log in Eq. (21) is 2, the entropy is expressed in bits.

Let (X, Y) be a pair of discrete random variables with a *joint distribution* $p(x, y)$, and Y be a discrete random variable with alphabet Σ and probability distribution $p(y) = P\{Y = y\}$, $y \in \Sigma$. Then the *joint entropy* $H(X, Y)$ is defined as $H(X, Y) = -\sum_{x\in\Lambda}\sum_{y\in\Sigma} p(x, y)\log p(x, y)$. The *conditional entropy* is given as $H(Y|X) = -\sum_{x\in\Lambda}\sum_{y\in\Sigma} p(x, y)\log p(y|x)$. The *Chain Rule* defines the relationship between joint entropy and conditional entropy as $H(X, Y) = H(X) + H(Y|X)$.

Depending on the properties of a communication channel, every input signal sequence induces a probability distribution in the output signal sequence. Considering the amount of *information* that one random variable obtains from another; *mutual information* is defined to measure the uncertainty in one random variable given the knowledge of another. The definition of *mutual information* is as follows: Given two random variables X and Y, with a joint probability distribution $p(x, y)$, *mutual information* $I = I(X; Y) = \sum_{x,y} p(x, y)\log\frac{p(x,y)}{p(x)p(y)}$. By the *Chain Rule*, we can derive alternative expressions for I as

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y). \end{aligned} \tag{22}$$

If a *discrete channel* has input and output alphabets Λ and Σ, respectively, with an input probability distribution $p(x)$, the probability of observing an output symbol y for a given sent symbol x is $p(y|x)$. The *capacity* of a discrete memoryless channel is $C = \max_{p(x)} I(X;Y)$. Note that the channel is *memoryless* if the channel transition probability only depends on the current input and output, i.e., is independent of the previous input or output. Furthermore, the channel has no *feedback*, which means the transmitter has no knowledge about previous or present receiver observations.

Besides channel capacity, mutual information is also applied to iterative decoding in the form of transfer characteristics of constituent decoders, as proposed by Stephan ten Brink in [13] and [14]. In iterative decoding, the exchange of extrinsic information is visualized as a decoding trajectory in the *extrinsic information transfer (EXIT) chart*, which provides a graphical tool for estimating the decoding thresholds of *low-density parity check (LDPC)* codes and *turbo* codes or other code ensembles [15].

Transmission rate (r) is defined as the ratio of the number of information bits from source (k) conveyed per channel use (n). Shannon proved that when the transmission rate is lower than the channel *capacity (C)*, error-free communication can be achieved. That is, for all rates $r < C$, there exists a sequence of $(2^{nr}, n)$ codes with a probability of error $\lambda^{(n)} \to 0$; conversely, for rates $r > C$, $\lambda^{(n)}$ is bounded away from 0. This approach to error-free communication is called channel coding. The *capacity* of a channel is determined by the channel's properties such as bandwidth, noise power spectral density, interference, etc [7].

4.2 Channel Models

4.2.1 Binary Symmetric Channel

A binary symmetric channel has a binary input, and the probability that the output is equal to the input is $1 - p$. When an error occurs with probability p, 0 is received as 1, or vice versa. Capacity (C) of a binary symmetric channel is given as $C = 1 - H(p)$, where $H(p)$ is the binary entropy function.

4.2.2 Binary Erasure Channel

The transmitter of a binary erasure channel (BEC) sends a bit, and the receiver receives either this bit or the message that indicates that this bit can not be received, that is, erased. Let α be the expected fraction of erased bits, then the channel capacity (C) is $C = 1 - \alpha$.

4.2.3 Gaussian Channel

A Gaussian channel is a time-discrete channel with input X_i, output Y_i, and with channel noise Z_i, which is *i.i.d.* from a Gaussian distribution with variance N. Therefore, the channel is represented as $Y_i = X_i + Z_i$, where $Z_i \sim \mathcal{N}(0, N)$. If an average power constraint (P) for the input $\mathbf{x} = (x_0, \ldots, x_{n-1})$ is assumed as $\frac{1}{n} \sum_{i=0}^{n-1} x_i^2 \leq P$, then the capacity (C) of a Gaussian channel with this power constraint P and noise variance N is $C = \frac{1}{2} \log(1 + \frac{P}{N})$ [11].

4.2.4 Bit-Shift Channel

A binary insertion/deletion channel is a channel model with synchronization loss, and was first studied by V. I. Levenshtein in [16]. A bit-shift channel is a kind of insertion/deletion channel. Consider the input binary sequence $\mathbf{x} = (x_1, \ldots, x_L)$ that is parsed into a sequence of phrase lengths $\tilde{\mathbf{x}}$, each of which are consecutive bits with the same value. The corresponding output binary sequence is $\mathbf{y} = (y_1, \ldots, y_{L'})$ where L' is equal to $L - 1, L$, or $L + 1$. The bit-shift channel model defines an inherent correlation of $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ as $\tilde{y}_i = \tilde{x}_i + \omega_i - \omega_{i-1}$, where ω_i is a random variable with values taken from $\{-1, 0, +1\}$, and with a probability distribution that depends on other properties of the system model. The positive (negative) value of ω_i corresponds to a right (left) bit-shift in the transition from time i to time $i + 1$. Additionally, the received sequences $\tilde{\mathbf{y}}$ do not contain any zeros; if $\tilde{y}_i = 0$, then this coordinate is removed, and $\tilde{y}_{i-1}$ and $\tilde{y}_{i+1}$ are merged [17]. Finally, the integer sequence $\tilde{\mathbf{y}}$ is converted to the binary output sequence $\mathbf{y}$.

4.2.5 Discretized Gaussian Shift Channel

With the same input sequence as in the bit-shift channel, the discretized Gaussian shift channel defines the received sequence as $\tilde{y} = \tilde{x} \cdot K$, where K is a random variable obeying a Gaussian distribution $\mathcal{N}(\alpha, \varepsilon^2)$ with mean α and variance ε^2. Consecutive samplings of K are assumed to be independent. With this definition, the input to the demodulator will be a sequence of alternating runs of high and low amplitude values; the detected duration $\tilde{y}$ of each run being a *real-valued* number. These real-valued numbers are quantized to *positive* integers using a quantization scheme, where the optimal choice for the quantization thresholds, i.e., the thresholds when mapping the real-valued numbers $\tilde{y}$ to *positive* integers, will depend on the code under consideration. Finally, the quantized sequence is converted to the binary output sequence $\mathbf{y}$.

5 Error Control Coding

As introduced in the last section, Shannon claimed that a good channel coding design can combat channel noise and offer reliable communication within the channel. In a communication system, an information source is represented by a binary k-tuple $\mathbf{u} = (u_0, \ldots, u_{k-1})$, called an information sequence. A channel encoder transforms $\mathbf{u}$ to a discrete encoded sequence $\mathbf{v} = (v_0, \ldots, v_{n-1})$, which is called a *codeword*. According to the encoding scheme and noise characteristics of a channel, the channel decoder transforms the received sequence $\mathbf{r}$ into a binary sequence $\hat{\mathbf{u}}$ called an *estimated information sequence*. A decoding error happens when $\hat{\mathbf{u}} \neq \mathbf{u}$. Channel coding aims to minimize the probability of decoding errors [18].

In this section, the cyclic redundancy check (CRC) code as a basic error control coding scheme is introduced. Trellis-based graphical representations of codes and two soft-decision decoding algorithms, the Viterbi and BCJR algorithms, which are widely implemented on this trellis structure, are explained. LDPC codes and code concatenation are discussed as well.

5.1 Linear Block Codes

There are two main types of channel codes: *block codes* and *convolutional codes*. The block encoder separates the information sequence into message blocks, each of which has k information symbols, and transforms each block to an $n-$tuple of discrete symbols, which is a *codeword*. A block code of *length* n and *dimension* k, denoted as an (n, k) code, is regarded as a nonempty subset C with size M over a finite alphabet $\mathcal{F}$. The *dimension* of C is $k = \log_{|\mathcal{F}|} M$, and the code rate is $r = k/n$. For a binary code, $\mathcal{F} = \{0, 1\}$. A block code is *linear* if and only if any linear combination of any two codewords is also a codeword. The *Hamming distance* between two codewords $\mathbf{x}, \mathbf{y} \in \mathcal{F}^n$ is the number of coordinates where $\mathbf{x}$ and $\mathbf{y}$ differ.

5.2 CRC Codes

As an important class of linear block codes, *cyclic codes* are fast and conveniently implemented for encoding and decoding due to their algebraic structure, and are particularly efficient for error detection. An (n, k) linear code C over a field F is called a *cyclic code* if every cyclic shift of a codeword in C is also a codeword in C. By this definition, we therefore represent a cyclic codeword $\mathbf{c} = (c_0, \ldots, c_{n-1})$ as the coefficients of a polynomial [19]:

$$\mathbf{c}(X) = c_0 + c_1 X + c_2 X^2 + \cdots + c_{n-1} X^{n-1}. \tag{23}$$

$\mathbf{c}$ shifted i positions will, in polynomial form, be

$$\mathbf{c}^{(i)}(X) = c_{n-i} + c_{n-i+1} X + \cdots + c_{n-1} X^{i-1} + c_0 X^i + \cdots + c_{n-i-1} X^{n-1}. \tag{24}$$

An algebraic relationship between $\mathbf{c}(X)$ and $\mathbf{c}^{(i)}(X)$ is

$$X^i \mathbf{c}(X) = \mathbf{q}(X)(X^n + 1) + \mathbf{c}^{(i)}(X) \tag{25}$$

where $\mathbf{q}(X) = c_{n-i} + c_{n-i+1} X + \cdots + c_{n-1} X^{i-1}$. The *generator polynomial* of an (n, k) cyclic code is

$$\mathbf{g}(X) = 1 + g_1 X + g_2 X^2 + \cdots + g_{n-k-1} X^{n-k-1} + X^{n-k}. \tag{26}$$

A cyclic code polynomial is a multiple of $\mathbf{g}(X)$. Furthermore, the generator polynomial $\mathbf{g}(X)$ of an (n, k) cyclic code is a factor of $X^n + 1$ [18].

A message polynomial $\mathbf{u}(X)$ is encoded by an (n, k) cyclic code in systematic form by following the steps below.

1. Multiply the message polynomial $\mathbf{u}(X)$ by X^{n-k}.
2. Divide the result by $\mathbf{g}(x)$ to obtain the remainder $\mathbf{b}(X)$.
3. Combine the results from 1 and 2, and generate a codeword as $\mathbf{b}(X) + X^{n-k}\mathbf{u}(X)$.

A CRC code is a cyclic code. The remainder $\mathbf{b}(X)$ is called a *checksum* in the CRC mechanism. In the data transmission sessions, the transmitter computes a *checksum* by dividing by $\mathbf{g}(X)$ and appends the *checksum* to the end of the data block which is about to be transmitted. At the receiver's end, $\mathbf{r}(X) \mod (\mathbf{g}(X))$ is computed. A value of zero, meaning that $\mathbf{g}(X)$ is a divisor of $\mathbf{r}(X)$, indicates that there is no error within this data block. The *XOR* operation is generally used for division and for getting the remainder (*checksum*). In an RFID system, the CRC code is typically used for error detection. There are several standardized generator polynomials, among which $\mathbf{g}(X) = x^{16} + x^{12} + x^5 + 1$ is a recommendation by the X.25 ITU-T standard protocol. The error detection property of a CRC code is mainly utilized in its implementation. However, it can also correct errors by means of computing syndromes and error patterns.

5.3 Trellises for Linear Block Codes

Using a trellis to represent a code is a graphical approach, by which the structure, properties, encoding, and decoding of a code can be studied and developed from an alternative perspective. The trellis was first introduced by Forney in [20] (1973), who implemented *maximum-likelihood decoding (MLD)* on a trellis and devised the *Viterbi* algorithm [21]. In 1974, Bahl, Cocke, Jelinek, and Raviv first used a trellis to represent linear block codes in [22]. They proposed the *BCJR* decoding algorithm based on the trellis structure and the properties of linear block codes. Wolf first proved an upper bound on the number of states of an n-section trellis diagram of a q-ary (n, k) linear block code in [23].

Given a binary linear block code (n, k), the encoding process is successively shifting k information bits into the encoder and n code bits $\mathbf{v} = (v_0, \ldots, v_{n-1})$ out chronologically. The output codeword sequence is finite, consisting of $n + 1$ *time* instants $(t_0, \ldots, t_n)$. A trellis consists of nodes and branches. This encoding process is described as [18, 24]:

1. Corresponding to the *time* instants, there are $n + 1$ levels of nodes. There is a set of nodes at level i ($0 \leq i \leq n$), where nodes are represented by states. At time 0, there is only one node at level 0, called the *initial node*. At time n, there is only one node at level n, called the *final node*.

2. A branch connecting a state s_i at level i to a state s_{i+1} at level $(i + 1)$, is labeled by encoder output v_i at time i.

3. The *initial node* does not have an incoming branch, while the *final node* does not have an outgoing branch. All the other nodes have one or two incoming and outgoing branches. Any two branches from one node have different labels indicating two different transitions from the same state.

4. Any path of branches from the *initial node* to the *final node* represents a codeword $(v_0, \ldots, v_{n-1})$ by the label on each branch.

In a trellis, states at each level are established by an encoder (generator matrix). A generator matrix $\mathbf{G}$ can be arranged to a special form by Gaussian elimination in order to construct a trellis. This special form is called *trellis oriented form (TOF)*, which satisfies the following conditions: the leading 1 of each row in $\mathbf{G}$ is in a column before any other leading 1 of any other row below it; and no two rows have a trailing 1 in the same column. These states are also labeled in order to determine nodes in the encoding and decoding processes. Labels of states are computed from the generator matrix or parity check matrix.

The number of states at time instant i is called the *state space complexity*, and is denoted as s_i. Define the *state space dimension* for a binary linear code as $\rho_i = \log_2 s_i$, and the sequence $(\rho_0, \ldots, \rho_n)$ as the *state space dimension profile*. It is clear that $\rho_0 = \rho_n = 0$, since there is only one state at time 0, as well as at time n. The complexity of a trellis can be measured by its state complexity, i.e., the *maximum state space dimension* $\rho_{\max}(C)$. For a binary (n, k) linear block code C, the maximum size of a state space is $2^{\rho_{\max}(C)}$. Wolf proved that its upper bound is $\rho_{\max}(C) \leq \min\{k, n-k\}$, which is tight for cyclic codes.

Example: Consider the $(216, 200)$ cyclic code with generator polynomial $\mathbf{g}(X) = x^{16} + x^{12} + x^5 + 1$ and generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \ldots & 0 \\ & & & & & & & & & & \ddots & & & & & & & & & & \\ 0 & 0 & \ldots & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

As its generator matrix is already in TOF, its *state space dimension profile* is $(0, 1, 2, \ldots, 14, 15, 16, \ldots, 16, 15, 14, \ldots, 2, 1, 0)$, and the maximum state space dimension is $\rho_{\max}(C) = 16$. Furthermore, its trellis is represented as follows: from the initial node (time 0) to the 16th level (time 16), $(0 \leq i \leq 16)$, there are 2^i states at the ith level; each state has one branch in and two branches out. From the 16th level to the $(n-16)$th level, all the levels have 2^{16} states, each of which has two branches in and two branches out. From the $(n-16)$th level to the final node (time n),

($n - 16 \leq i \leq n$), there are 2^{n-i} states on each level; each state has two branches in and one branch out.

5.4 Soft-Decision Decoding

Transmission is the process in which the encoded sequence $\mathbf{v}$ is passed through communication channels, and is received as $\mathbf{r}$ at the other side of the channel by a receiver. The decoder has to compute $\hat{\mathbf{u}}$, as an estimation of the information sequence $\mathbf{u}$, from the received sequence $\mathbf{r}$. For binary codes, hard-decisions are made by using a certain threshold matching each received signal to two levels, denoted as 0 and 1, respectively. Hard-decision decoding is based on codes' algebraic structures, matching the received sequence to the closest codeword in terms of *Hamming distance* by making a hard-decision on each symbol. Soft-decision decoding, on the other hand, matches each received signal to more than two levels, even up to the continuum between 0 to 1. This additional information provides reliability for decoding. A decoding strategy that processes soft-decision input and produces soft-decision output is called a *soft-in/soft-out (SISO) decoding algorithm*. Soft-decision decoding has better performance than hard-decision decoding since the soft-decision offers additional information contained on multiple levels quantized for codeword estimation. For instance, it is widely utilized in iterative decoding.

5.5 MLD and Maximum a Posteriori Probability Decoding

Suppose a codeword $\mathbf{v}$ of length n over $\mathcal{F}$ is transmitted over a noisy channel with output alphabet R. Assume that the channel is a *discrete and memoryless channel (DMC);* R is therefore a discrete set. Let $P(\mathbf{r}|\mathbf{v})$ be the probability of observing the sequence $\mathbf{r}$ at the receiver, given that the sequence $\mathbf{v}$ is transmitted. This is the *likelihood* of the codeword $\mathbf{v}$. The probability $P(\mathbf{v}|\mathbf{r})$ is called the *a posterior probability (APP)* of the codeword $\mathbf{v}$. This is the conditional probability that $\mathbf{v}$ is transmitted after observing sequence $\mathbf{r}$ at the receiver.

The optimal decoding is to find the codeword $\mathbf{v}$ that either maximizes the probability $P(\mathbf{r}|\mathbf{v})$ called MLD, or to maximize the probability $P(\mathbf{v}|\mathbf{r})$ called *maximum a posteriori probability (MAP) decoding* [25]. When a vector $\mathbf{r} = (r_0, \ldots, r_{n-1})$ is observed as the channel output, then according to *Bayes' rule*, the relationship between MLD and MAP decoding is according to

$$P(v_i|r_i) = \frac{P(r_i|v_i)P(v_i)}{P(r_i)}. \tag{27}$$

$P(r_i)$ is same for all the v_i. Thus, if we assume that all v_i are equiprobable, then maximizing $P(v_i|r_i)$ is equivalent to maximizing $P(r_i|v_i)$. That is, the two decoding approaches coincide when all codewords $\mathbf{v}$ are equally likely to be transmitted ($P(v_i)$ is a constant).

Additionally, by converting to logarithmic form, which is a monotonically increasing function, maximizing Eq. (27) becomes equivalent to maximizing $\log P(\mathbf{v}|\mathbf{r}) = \sum_i \log P(v_i|r_i)$. This allows us to simplify the multiplication of $P(v_i|r_i)$ to the addition of $\log P(v_i|r_i)$. MLD is implemented in the Viterbi algorithm, while MAP decoding is implemented in the BCJR algorithm [15].

5.6 Viterbi Decoding

The Viterbi algorithm is a well-known MLD algorithm that searches a code trellis for a maximum-likelihood path. MLD chooses $\hat{\mathbf{v}}$ as a codeword $\mathbf{v}$ that maximizes the log-likelihood function $\log P(\mathbf{r}|\mathbf{v})$ as we argued in the last subsection. When it comes to a trellis, let us denote $M(\mathbf{r}|\mathbf{v}) = \log P(\mathbf{r}|\mathbf{v})$ as a path (or codeword) *metric*. Furthermore, any branch in M is denoted by $M(\mathbf{r}_i|\mathbf{v}_i)$ and called a *branch metric*, and the terms $\log P(r_i|v_i)$ are denoted by $M(r_i|v_i)$ and called *bit metrics*. A partial path metric for the first t branches of a path is computed as

$$M([\mathbf{r}|\mathbf{v}]_t) = \sum_{i=0}^{t-1} M(\mathbf{r}_i|\mathbf{v}_i) = \sum_{i=0}^{t-1} \log P(\mathbf{r}_i|\mathbf{v}_i). \tag{28}$$

The Viterbi decoding algorithm finds the path through the trellis with the largest metric when the received sequence is $\mathbf{r}$. It includes the *add* operation that at each time unit adds branch metrics to each previous stored path metric, the *compare* operation that compares all paths entering each state, and the *select* operation that selects the path with the largest metric. Viterbi decoding processes $\mathbf{r}$ in a recursive manner. A Viterbi decoding algorithm that also produces soft values is called a *soft-output Viterbi algorithm (SOVA).*

5.7 BEAST

BEAST is the abbreviation of *Bidirectional Efficient Algorithm for Searching code Tree*, an algorithm devised to reduce decoding complexity. It was first introduced in [26] to compute the weight spectrum of convolutional codes. In [27], BEAST was used to implement soft-decision MLD of block codes.

Bidirectional decoding that searches a decoding path from two opposite directions on the trellis simultaneously has a reasonably reduced complexity (in term of the number of visited nodes during the search). Especially for cyclic codes with symmetrical structure, BEAST uses forward recursion on the first half of a code trellis, and backward recursion on the second half of the trellis, matching nodes in the middle of the trellis to find a codeword.

Consider that a trellis from one direction is in a tree form, and denoted by δ a node in this tree. Given that the weight of the codeword is w, the weight of a path from the initial node to the node δ is denoted as w_i, while the weight of a path from the final node is denoted as w_f. Further, the weight of a parent node is w_i^p, while the weight of a child node is w_f^c . Then, the forward recursion finds the set of nodes

$$\mathcal{F} = \left\{ \delta | w_i \geq \frac{w}{2}, w_i^p < \frac{w}{2} \right\}. \tag{29}$$

The backward recursion finds the set of nodes

$$\mathcal{B} = \left\{ \delta | w_f \leq \frac{w}{2}, w_f^c > \frac{w}{2} \right\}. \tag{30}$$

If a state of $\mathcal{F}$ is equal to a state of $\mathcal{B}$, and the sum of the lengths of the two paths is equal to the total length of the trellis, then the combination of the two paths is a codeword.

5.8 BCJR Decoding

The BCJR algorithm is a soft-decision decoding algorithm based on a code's trellis structure, and it can be implemented for decoding both convolutional and block codes.

The BCJR algorithm implements MAP decoding on a trellis structure. Since $P(v_i|r_i)$ depends on $P(v_i)$, with the assumption that $P(v_i)$ is not necessarily the same for all i, we represent it in logarithmic form as

$$L(v_i) \triangleq \log \frac{P(v_i = 1|\mathbf{r})}{P(v_i = 0|\mathbf{r})}. \tag{31}$$

This is called the *log-likelihood ratio (LLR)*, and v_i is estimated by the sign of this value: $v_i = 1$ if $L(v_i) > 0$; otherwise $v_i = 0$. $L(v_i)$ can be computed from joint probabilities as $L(v_i) = \log \frac{P(v_i=1,\mathbf{r})}{P(v_i=0,\mathbf{r})}$, since $P(v_i|r_i)P(r_i) = P(v_i, r_i)$ according to *Bayes' rule*.

Based on the code's trellis structure, by examining states from time i to time $i+1$, since each branch connects a state s_i to a state s_{i+1} with a label that either outputs 0 or 1, joint probabilities can be represented as

$$\lambda_i(s', s) \triangleq P(s_i = s', s_{i+1} = s, \mathbf{r}). \tag{32}$$

Then, let $P(v_i = 0, \mathbf{r}) = \sum_{s' \to s,0} \lambda_i(s', s)$ represent the sum of $\lambda_i(s', s)$ over all transitions from s' to s that produce output 0, and $P(v_i = 1, \mathbf{r}) = \sum_{s' \to s,1} \lambda_i(s', s)$ the sum of $\lambda_i(s', s)$ over all transitions from s' to s that produce output 1. Let $\mathbf{r}_{j,k} \triangleq (r_j, r_{j+1}, r_{j+2}, \ldots, r_k)$ be a section of the received sequence, and define

$$\alpha_i(s) \triangleq P(s_i = s, \mathbf{r}_{0,i}), \tag{33}$$

$$\beta_i(s) \triangleq P(\mathbf{r}_{i,n} | s_i = s). \tag{34}$$

For the initial and final states, we have $\alpha_0(s_0) = \beta_n(s_f) = 1$. Define the probabilities of state transition which depend on the transmission channel as follows

$$\gamma_i(s', s) \triangleq P(s_{i+1} = s, r_i | s_i = s'). \tag{35}$$

>From the definitions above, we have the joint probabilities

$$\lambda_i(s', s) = \alpha_i(s')\gamma_i(s', s)\beta_{i+1}(s). \tag{36}$$

The computations of $\alpha_i(s)$ and $\beta_i(s)$ are recursive processes, called *forward recursion* and *backward recursion*, respectively. The detailed expressions are

$$\alpha_i(s) = \sum P(s_{i-1} = s', s_i = s, r_{i-1}, \mathbf{r}_{0,i-1}) = \sum \alpha_{i-1}(s')\gamma_{i-1}(s', s), \tag{37}$$

$$\beta_i(s) = \sum P(r_i, \mathbf{r}_{i+1,n}, s_{i+1} = s' | s_i = s) = \sum \gamma_i(s, s')\beta_{i+1}(s'). \tag{38}$$

5.9 LDPC Codes

LDPC codes are linear error correcting codes, first introduced in [28] by Gallager in 1962. They are the first class of codes (together with turbo codes) that closely approach channel capacity as described by

Shannon. But they did not really draw much attention until Tanner represented them from a graphical point of view in [29] in 1981. Later, Mackay *et al*. rediscovered these codes [30] and confirmed their excellent performance. Subsequently, research progressed rapidly, and graphical coding and iterative decoding became more and more feasible. It was quickly found that irregularly constructed LDPC codes could have better performance than regular ones [31, 32]. In particular, [32] showed that such irregular constructions of LDPC codes can get to within 0.0045 dB of the Shannon limit.

An LDPC code is specified by a *sparse* parity check matrix, one with few 1's. Such a code can be constructed in a graphical way by a *Tanner graph*. In [33], a general method that constructs a Tanner graph by a *progressive edge-growth (PEG)* algorithm is presented. Decoding can be hard-decision, soft-decision, or a combination of both. An efficient decoding algorithm is the *sum-product algorithm* which is an iterative decoding algorithm based on *belief propagation*.

5.9.1 Code Structure and Tanner Graph

A *(γ, ρ)-regular* LDPC code has a parity check matrix $\mathbf{H}$ with ρ 1's in every row and γ 1's in every column. If $\mathbf{H}$ is $j \times n$ matrix, then the density of $\mathbf{H}$ is $r = \rho/n = \gamma/j$.

To represent a linear block code that is specified by a parity check matrix ($\mathbf{H}$), a graph $\mathcal{G}$ consisting of two sets of vertices $\mathcal{V}$ and $\mathcal{C}$ is constructed. Vertices in $\mathcal{V}$ are variable nodes, denoted as $v_0, \ldots, v_{n-1}$, and vertices in $\mathcal{C}$ are check nodes, denoted as $c_0, \ldots, c_{j-1}$. For all codewords $\mathbf{v}$, we have $\mathbf{v}\mathbf{H}^T = \mathbf{0}$; therefore every check node is a check-sum of the variable nodes in its row. A variable node is connected to a check node by an edge. The number of edges from one node is the *degree* of this node. This graphical representation is the *Tanner graph*.

By definition we know that in a regular LDPC code all the rows of $\mathbf{H}$ have the same weight, indicating that all check nodes have the same degree. For an irregular LDPC code, on the other hand, the degree of each node may be different. Degree distributions of variable nodes and check nodes are defined as

$$\gamma(x) = \sum_{i=2}^{d_v} \gamma_i x^{i-1} \quad \text{and} \quad \rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1} \tag{39}$$

where d_v (resp. d_c) is the maximum variable (resp. check) node degree. An example of a parity check matrix is shown below, with degree distributions $\gamma(x) = 0.8x + 0.2x^2$ and $\rho(x) = 0.25x + 0.75x^2$.

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

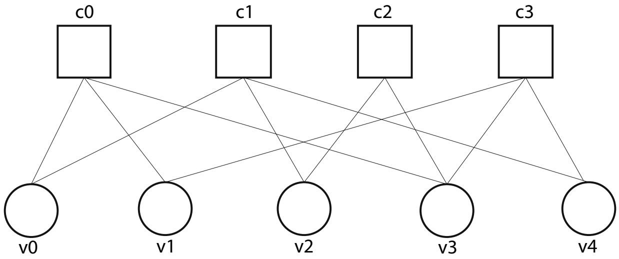

Fig. 7: *An example of a Tanner graph of an irregular LDPC code*

The *rate* of irregular LDPC codes is

$$\text{rate}(\gamma, \rho) = \frac{n-j}{n} = 1 - \frac{\int_0^1 \rho(x)\mathrm{d}x}{\int_0^1 \gamma(x)\mathrm{d}x}. \tag{40}$$

5.9.2 Design of LDPC Codes

Since LDPC codes are specified by their parity check matrices, to design an LDPC code, a parity check matrix has to be constructed. The PEG algorithm is a powerful algorithm to generate Tanner graphs with large girth. It is an edge selection procedure that examines edges successively and selects the one with the smallest possible impact on the girth. After the best edge is selected, the graph with this new edge is updated, and the selection of the next edge begins.

Given a required code rate, maximum degrees of check and variable nodes, and a check node degree distribution, an optimized variable node degree distribution can be obtained with the help of linear programming. Using the number of symbol nodes and the degree distributions, the PEG algorithm can be implemented to generate a Tanner graph. This is one approach to design an LDPC code.

5.10 Concatenation of Binary Codes

Concatenated coding is a technique that constructs a long code with short code components [18]. As illustrated in Fig. 8, one concatenates two binary codes $C_1(n_1, k_1)$ and $C_2(n_2, k_2)$, to form an (n_1n_2, k_1k_2) code with code rate k_1k_2/n_1n_2. The encoding process is to arrange the information sequence into a matrix with k_2 rows, where each row has k_1 bits. Then each row is encoded to a codeword in C_1. These codeword matrices with k_2 rows and n_1 columns are encoded to codewords in C_2. C_1 and C_2 are called *inner* and *outer* codes, respectively. The decoding process is from rows to columns. Uncorrected error(s) after row decoding might be corrected by columns decoding. In this manner, the concatenated codes have an advantage in error correction and decoding complexity. By the help of *SISO* and iterative decoding, decoding performance can be improved even more.

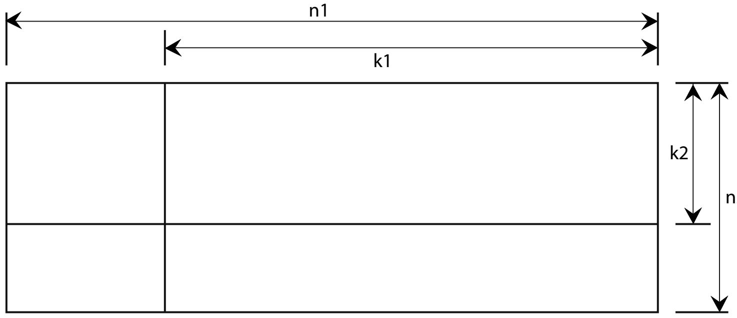

Fig. 8: *Concatenation*

Binary concatenation can be implemented in either *serial* or *parallel* form. For instance, we consider the serial concatenation of an outer binary error correcting code, C_O, and an inner LDPC code, C_I. The information sequence is encoded by the outer encoder of rate R_O. The resulting codeword is mapped to a codeword of an LDPC code C_I of rate R_I. The overall code rate is $R = R_OR_I$. For the given inner LDPC code C_I and outer code C_O, the performance in the error floor region on the BEC can be optimized by adding an interleaver/permutation between the outer and the inner code to increase the *stopping distance* of the overall concatenated code.

6 Constrained Codes

Designing codes by constraints was initially implemented in magnetic recording systems, in which recorded content is converted from binary sequence to data track using magnetic polarities (two-level waveforms). The recorded data is converted to voltages by means of *peak detection* in the reading process. For any recording of a binary sequence within one bit time unit, if a high voltage peak is detected, a 1 is declared; otherwise, a 0 is declared [34]. With some modulation schemes, too many consecutive ones may cause ISI; conversely, too many zeros between two ones may cause problems in timing control. Thus, a *runlength* of consecutive zeros between two successive ones is a critical *constraint* when designing codes for magnetic recording systems [35]. However, constrained codes are also used in code design for passive RFID systems. In this section, properties of runlength constrained codes are addressed. Power spectra of runlength codes are discussed since the power spectral density of codes should be matched to the channel characteristics.

6.1 Runlength Constrained Codes

Define a constrained binary *dk-runlength-limited* sequence $(0 \leq d \leq k)$ as a sequence that satisfies the following two conditions:

1. *d-constraint*: the length of a run of consecutive zeros that separates any two ones is at least d.
2. *k-constraint*: the length of a run of consecutive zeros is at most k.

If only the first condition is satisfied, the sequence is *d-limited* ($k = \infty$). In the *dk-runlength-limited* sequence, a *d-constraint* reduces the effect of ISI and a *k-constraint* assists timing control [36]. This is due to the fact that the information sequence is first encoded using a (d, k)-constraint encoder and then post-encoded using a recursive encoder with generator polynomial $1/(1 + D)$. This is called *differential mode*.

6.2 Labeled Graphs and Capacity

Any constrained system can be represented by a labeled directed graph $G = (V, E, L)$, where V is a finite set of states; E is a finite set of edges in which each edge e is an edge from a state v_i to a state v_j, $v_i, v_j \in V$. Each edge has a label $l \in L$, where L is a finite set of labels. A *labeled graph* is shown in Fig. 9.

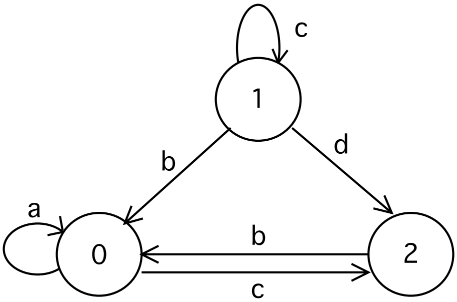

Fig. 9: *Labeled graph*

A matrix $\mathbf{A}_G$ describing a labeled graph G by adjacency between states is called the *adjacency matrix*. The *adjacency matrix* of the graph in Fig. 9 is

$$\mathbf{A}_G = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

The *capacity* of a constrained system is calculated by $C_{\text{ap}} = \log_2 \lambda(\mathbf{A}_G)$, where $\lambda(\mathbf{A}_G)$ is the largest real eigenvalue of the *adjacency matrix* of its labeled directed graph G [36]. Thus, the capacity of the constrained graph in Fig. 9 is 0.6942. Alternatively, the capacity of a (d, k)-constrained sequence can be evaluated as the base-2 logarithm of the largest real root σ of

$$x^{k+2} - x^{k+1} - x^{k-d+1} + 1 = 0. \tag{41}$$

Additionally, if a (d, k)-constrained code has the property that the probability of i consecutive zeros followed by a one is equal to $\sigma^{-(i+1)}$, then the code achieves *maximum entropy*, that is, its information entropy equals the constraint capacity [36].

State splitting [37, 38] is a technique that transforms any constraint graph. A state splitting of state u is performed as follows: let E_u be the set of edges that start in state u, partition E_u into two disjoint and nonempty sets, E_{u_1} and E_{u_2}; replace state u with two states u_1 and u_2, where each of the states u_i is assigned the set E_{u_i} of output edges; for every edge into state u in the original graph, create two edges both starting in the same state and with the same label as the original edge, leading into states u_1 and u_2.

6.3 Variable-Length Codes

A variable-length code maps source symbols to a variable number of bits. It retains the merits of both short and long codes. Simple variable-length codes can be constructed with rates that are very close to capacity. They reduce encoder and decoder complexities compared to a fixed-length code with relative rate or sequence properties [39]. In a passive RFID system, the messages that are sent to tags require low decoding cost. Variable-length codes are therefore considered due to their simple window-shift decoding. Additionally, the variable-length codes can offer synchronization. The timing errors that are induced by bit-shift channels can be detected and corrected by window-shift decoding, or a combination of window-shift decoding and other error correcting codes.

6.4 Levenshtein Distance

As mentioned in Section 4, Levenshtein studied the insertion/deletion channels and also proposed the *Levenshtein distance* that is a measure of the number of symbol insertions/deletions and substitutions that are necessary to transform a codeword $\mathbf{x}$ to another codeword $\mathbf{y}$. It is sometimes referred to as the *edit distance*. Computing the Levenshtein distance of two received sequences is based on a matrix that hold the Levenshtein distances between all prefixes of the first sequence and all prefixes of the second sequence.

6.5 Power Spectra Matched for the Inductively Coupled Channel

Since power transfer is important in inductively coupled channels, code designs have to ensure that the *power spectral density* of the signal matches the frequencies of the channel transfer function. Ideally, the power spectral density should be zero outside the frequency band of the channel, but this is practically impossible. In this subsection, the calculation method for the power spectral density of signals generated by runlength constrained codes is presented.

The *autocorrelation function* R of a stochastic process Y_t is defined by

$$R_Y(t_1, t_2) = E(Y_{t_1} Y_{t_2}) \tag{42}$$

where $E(.)$ denotes expectation. A process is said to be *stationary* if its statistics are time-invariant, and is said to be *wide-sense stationary* if its mean and autocorrelation function are time-invariant, that is,

$$\begin{aligned} E(Y_t) &= M_Y, \\ E(Y_t Y_{t+\tau}) &= R_Y(\tau) \end{aligned} \tag{43}$$

where M_Y is the mean value of the process Y_t.

The spectral analysis of block-coded sequences is complicated by the fact that the statistical properties are cyclostationary rather than stationary [39]. The cyclostationarity of a process W_t is defined by

$$\begin{aligned} M_W(t) = E(W_t) &= M_W(t+T), \\ R_W(t, t+\tau) &= R_W(t+T, t+\tau+T) \end{aligned} \tag{44}$$

where T, the smallest time interval giving equality, is the *period* [39]. There is a technique that defines an equivalent stationary process whose statistics are those of the cyclostationary process averaged over the period T. The cyclostationary process W_t is changed into a stationary process $W_{\theta,t}$ by adding a random variable θ that is uniformly distributed over $0 \leq \theta < T$ [39].

The signals generated by runlength constrained codes are *cyclostationary* in general. After mapping a binary code to the two-level waveform, the signal is represented as

$$X(t) = \sum_{i=-\infty}^{\infty} A_i P_T(t - iT) \tag{45}$$

where $P(t)$ is a standard pulse shape with *clock period* T; A_i is the signal generated by codeword symbol value for the time interval $iT \leq t \leq (i+1)T$, it is cyclostationary; and

$$P_T(t) = \begin{cases} 1 & \text{if } 0 \leq t < T \\ 0 & \text{otherwise} \end{cases} . \tag{46}$$

In [40], under an assumption that the process is ergodic, a process $\{Z_t\}$ is defined by $Z_t = A_{t+\theta}$, where θ is a discrete random variable uniformly distributed over the period r of the $X(t)$ cyclostationary process. This new process can be shown to be wide-sense stationary. Then the Fourier transform relationship between the autocorrelation function and the power spectral density can be applied to compute the power spectrum. The power spectrum of $X(t)$ is [41]

$$\begin{aligned} S_X(f) &= |P(f)|^2 T^2 S_A(e^{2\pi f T\sqrt{-1}}) \\ &= \frac{1}{T}|P(f)|^2[R_A(0) + 2\sum_{j=1}^{\infty} R_A(j)\cos(2\pi j f T)] \end{aligned} \tag{47}$$

where $P(f)$ is the Fourier transform of $P_T(t)$ and

$$S_A(D) = \sum_{j=-\infty}^{\infty} R_A(j)D^j = \sum_{j=-\infty}^{\infty} E(A_0 A_j)D^j. \tag{48}$$

7 Summary of Papers

This thesis consists of six papers. In the following sections, a short overview of each paper is given.

7.1 Paper I

The first paper is entitled: *Exploiting the CRC-CCITT Code on the Binary Erasure Channel.*

CRC codes are mostly used for error detection in many automatic-repeat-request (ARQ) protocols, and also in many RFID standards. They can in principle be used for error correction. This work is an investigation of the error correction performance of the CRC-CCITT code on the BEC.

Let C be a binary linear code of length n and dimension k. Let $E_t = E_t(C) = E_t(C, \mathcal{D})$ be the number of erasure patterns P with t erasures that are uncorrectable under decoding algorithm $\mathcal{D}$ for code C. Then the probability of recovery failure with decoding algorithm $\mathcal{D}$ is

$$P_{rf} = P_{rf}(\mathcal{D}) = \sum_{t=d} E_t \epsilon^t (1-\epsilon)^{n-t}$$

where ϵ is the channel erasure probability and d is the minimum size of an uncorrectable set under decoding algorithm $\mathcal{D}$ for code C. Let d be the minimum distance of C and $\mathcal{D}$ be MLD. Then from [42], [43], and [44] we formulate E_t as

1. $E_t = 0$ for $t < d$,
2. $E_t \leq \binom{n}{t}$ for all t, with equality for $t > n - k$,

3. $E_t = \sum_{w=d}^{t} A_w \binom{n-w}{t-w}$ for $d \leq t < d_2$, where d_2 is the second generalized Hamming weight, and

4. $E_t \leq \sum_{w=d}^{t} A_w \binom{n-w}{t-w}$ for any t, $d \leq t \leq n$

where A_w is weight distribution, i.e., the number of codewords of weight w. The weight distribution of the CRC-CCITT code can be computed by application of the MacWilliams identities. We compute upper bounds on P_{rf} for the CRC-CCITT code for different erasure probabilities. These upper bounds agree very well with simulation results.

Secondly, we improve the decoding performance by a serial concatenation of the CRC code as an outer code and an LDPC code as an inner code. The code design strategy is: 1) pick an overall block length and target code rate; 2) determine the optimum degree distributions for the inner LDPC code and design a parity check matrix **H** using an improved PEG algorithm; 3) compute the *initial* part of the input-output stopping set distribution for **H** and a corresponding list L of small-size stopping sets using the algorithm from [45]; and 4) find a good permutation/interleaver (to be added prior to the LDPC encoder) such that most of the stopping sets in L are not stopping sets in the overall concatenated code. We find that the CRC-LDPC serial concatenation provides a significant lowering of the error floor, and the error floor can be further improved by a suitable choice of interleaver.

7.2 Paper II

The second paper is entitled: *Constrained Codes for Passive RFID Communication*.

In this work, we study the physical layer coding of information on inductively coupled channels, with emphasis on coding for error control for the *reader-to-tag* channel. A tag in a passive RFID system has no internal power source, it collects the power from the carrier of the reader. After some initial transient delay, the tag's power circuitry should be charged sufficiently to provide operating power for the tag. The amount of transferred power can be influenced by the encoding scheme used. We define the *power content* of a binary vector $\mathbf{a} \in \mathrm{GF}(2)^n$, denoted by $P(\mathbf{a})$, as the rational number $w(\mathbf{a})/n$, where $w(.)$ denotes the Hamming weight of its binary argument. Let $\mathcal{C}$ denote a block code or a variable-length code, and let $\mathcal{C}^{[N]}$ be the set of sequences of length $N \geq 1$ over

$\mathcal{C}$, i.e., the set of N consecutive codewords. We define the *average power* of $\mathcal{C}$ by

$$P_{\text{avg}}(\mathcal{C}) = \frac{\sum_{j=1}^{|\mathcal{C}|} w_j}{\sum_{j=1}^{|\mathcal{C}|} n_j}$$

where w_j is the Hamming weight and n_j the length of the jth codeword of $\mathcal{C}$. For codes defined by a state diagram, the average power content P_{avg} can be computed from the stationary probabilities of the states. Further, the *minimum sustainable power* is defined as $P_{\min}(\mathcal{C}) = \min_{\mathbf{a}\in\mathcal{C}} P(\mathbf{a})$, and the *local minimum power* is defined as the minimum positive value of the ratio m_p/n_p (m_p out of every n_p consecutive transmitted bits are $1'$s) over all possible sequences in $\mathcal{C}^{[N]}$, for any finite value of N, where $n_p \geq m_p$ are arbitrary positive integers.

Consider transmission under different channels, such as the additive white Gaussian noise channel and the bit-shift channel. We propose a discretized Gaussian shift channel model from the reader to the tag. Suppose the reader transmits a run of $\tilde{x}$ consecutive equal symbols (or bits). This corresponds to an amplitude modulated signal of duration $\tilde{x}$. At the tag, we assume that this is detected (according to the tag's internal clock) as having duration

$$\tilde{y} = \tilde{x} \cdot K$$

where K is a random variable with, in general, a Gaussian distribution $\mathcal{N}(\alpha, \varepsilon^2)$ with mean α and variance ε^2. Consecutive samplings of K are assumed to be independent. If $\alpha \neq 1$, it means that the tag has a systematic drift, which may affect the reader's ability to function at all. Thus, we will focus on the case $\alpha = 1$. With this definition, the input to the demodulator will be a sequence of alternating runs of high and low amplitude values; the detected duration $\tilde{y}$ of each run being a *real-valued* number.

Two different quantization schemes were proposed, denoted by $\mathcal{Q}(\mathcal{A})$ and $\mathcal{Q}_{\text{rounding}}$. The quantization scheme $\mathcal{Q}_{\text{rounding}}$ is based on rounding the received values to the nearest positive integer values, while the second quantization scheme has quantization thresholds $t_l = 2a_{l-1}a_l/(a_{l-1} + a_l)$, $l = 2, \ldots, |\mathcal{A}|$. Here, $\mathcal{A}$ will be the positive integers.

We design runlength constrained codes for *error avoidance* on this channel model based on the principle that is introduced in [46, 47]. We denote a particular binary runlength limitation as RLL$(\mathcal{L}_0, \mathcal{L}_1)$, where

$\mathcal{L}_b$ is the set of admissible runlengths of binary symbol b. In particular, $\text{RLL}(\{1,2\},\{1,2\}), \text{RLL}(\{1\},\{1,2\}), \text{RLL}(\{1,3\},\{1,3\}), \text{RLL}(\{1\},\{1,3\})$, $\text{RLL}(\{1,2,4\},\{1,2,4\})$, and $\text{RLL}(\{3^i : i = 0,\dots,L\},\{3^i : i = 0,\dots,L\})$-limited codes are evaluated by several properties, for instance, sustainable power, local minimum power, and average power. Furthermore, the *frame error rate (FER)* performance on the discretized Gaussian shift channel with quantization schemes $\mathcal{Q}_{\text{rounding}}$ and $\mathcal{Q}(\mathcal{A})$ is compared by simulation. The two variable-length codes $\{10, 011\}$ and $\{101, 01101\}$, introduced in [47], are also included in the comparison.

7.3 Paper III

The third paper is entitled: *On the Capacity of a Discretized Gaussian Shift Channel.*

In this work, we study the capacity of a discretized Gaussian shift channel. We define a truncated version of the channel, denoted by $\mathcal{H}_{L,T}$, with input alphabet $\mathcal{X} = \{1,\dots,L\}$, output alphabet $\mathcal{Y} = \{1,\dots,L'\}$, where L and L' are integers greater than one, and channel transition probabilities $p(y|x)$. The parameter L' is the smallest integer output of the discretized Gaussian shift channel (with any of the two quantization schemes $\mathcal{Q}_{\text{rounding}}$ or $\mathcal{Q}(\mathcal{A})$ such that the probability of observing L' for any given input $x \in \mathcal{X}$ is smaller than some small threshold probability T. The normalized mutual information between the channel input X and channel output Y is

$$I(X;Y) = \frac{\sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} p(x)p(y|x)\log_2\left(\frac{p(y|x)}{\sum_{j\in\mathcal{X}} p(j)p(y|j)}\right)}{\sum_{j\in\mathcal{X}} j \cdot p(j)} = \frac{I_{\text{num}}(X;Y)}{I_{\text{denom}}(X;Y)} \tag{49}$$

and the capacity of $\mathcal{H}_{L,T}$ (in bits per symbol) can be obtained by maximizing the fraction in Eq. (49) over all input probability distributions $p(x)$. We address two propositions of the mutual information $I(X;Y)$: 1) the mutual information $I(X;Y)$ in the fraction is quasi-concave in $p(x)$; and 2) the partial derivative of the mutual information $I(X;Y)$ in the fraction with respect to $p(x)$, $x = 1,\dots,L-1$, is

$$\frac{\partial I(X;Y)}{\partial p(x)} = \frac{\frac{\partial I_{\text{num}}(X;Y)}{\partial p(x)} \sum_{j\in\mathcal{X}} j \cdot p(j) - I_{\text{num}}(X;Y)(x-L)}{\left(\sum_{j\in\mathcal{X}} j \cdot p(j)\right)^2}$$

where

$$\frac{\partial I_{\text{num}}(X;Y)}{\partial p(x)} = \sum_{y\in\mathcal{Y}} p(y|x)\log_2\left(\frac{p(x|y)}{p(x)}\right) - \sum_{y\in\mathcal{Y}} p(y|L)\log_2\left(\frac{p(L|y)}{p(L)}\right).$$

By computation, using a gradient ascent algorithm, we observe that choosing $L = 12$ gives capacity results very close to the *exact* capacity of the discretized Gaussian shift channel with quantization scheme $\mathcal{Q}_{\text{rounding}}$. Another observation is that the quantization scheme $\mathcal{Q}_{\text{rounding}}$ gives the best performance for intermediate-to-large values of ε, while the quantization scheme $\mathcal{Q}(\mathcal{A})$ performs better when ε decreases. Additionally, in the optimal input distributions $p(x)$ we observe that the shortest runlengths have the highest probabilities. By these observations, we conclude that: an error control code for this channel should be designed to give coded sequences in which small runlengths occur more frequently than longer runlengths.

7.4 Paper IV

The fourth paper is entitled: *Numerical Study of Power Transfer in a Passive RFID System.*
The communication from a reader to a tag under different separations that relate system power consumption and channel capacity is studied in this work. The *mutual inductance* of the tag circuit in relation to a reader circuit is $M_{rt} = \frac{\mu_0 \cdot N_r \cdot R_r^2 \cdot N_t \cdot R_t^2 \cdot \pi}{2\sqrt{\left(R_r^2+x^2\right)^3}}$, which is a function of x, being the inter-coil separation of the reader and the tag.

In continuous frequency bands, the *efficiency* of power transfer is defined in [48] as

$$\eta(f) = \frac{|H(f)|^2 R_{\text{tag}}}{R_{\text{reader}} + |H(f)|^2 R_{\text{tag}}}$$

where $H(f) = I_{\text{received}}(f)/I_{\text{transmitted}}(f)$ is the frequency response of the current transfer function. Let $w = 2\pi f$, then by applying *Kirchhoff's Voltage law* to the tag's circuit, we obtain $H(jw,d) = \frac{I_{\text{tag}}(jw)}{I_{\text{reader}}(jw)} = \frac{jwM(d)}{jwL_t(1+jwR_tC_t)+R_t}$, which is a function of frequency and inter-coil separation d.

By the defined power efficiency, the power consumption along the inter-coil separation can be achieved. Further, the capacity of the channel can also be reached by $C = \int_f \log_2(1 + \frac{\eta(f)P^*(f)}{N_0})df$ where $P^*(f)$ is the optimized power allocation obtained from the *water-filling* algorithm which allocates more power to sub-channels with higher *signal-to-noise ratio (SNR)*, and N_0 is the power spectral density of the additive white Gaussian noise.

7.5 Paper V

The fifth paper is entitled: *On the Power Transfer of Error-Control Codes for RFID Communications*.

In this work, we consider the *power spectrum* of error control codes that are designed for the reader-to-tag channel. Let $\{b_i\}$ denote a binary sequence, and let $A_i = 1 - 2b_i$, being the BPSK-modulated version of $\{b_i\}$. We assume that the sequence $\{A_i\}$ is wide-sense stationary. The spectrum (in the D-domain) of a wide-sense stationary (discrete-time) process $\{A_i\}$ is:

$$S_A(D) = \sum_{j=-\infty}^{\infty} R_A(j)D^j = \sum_{j=-\infty}^{\infty} E(A_0 A_j)D^j$$

where $R_A(j) = E(A_0 A_j)$ is the jth autocorrelation coefficient of $\{A_i\}$, and $E(.)$ denotes statistical expectation. We use $D = e^{2\pi f T\sqrt{-1}}$, where f is the frequency and T is the symbol period. The transmitted signal $V(t) = \sum_{i=-\infty}^{\infty} A_i P_T(t - iT)$ has power spectral density [40]

$$S_V(f) = \left(\frac{\sin(\pi f T)}{\pi f T}\right)^2 T S_A\left(e^{2\pi f T\sqrt{-1}}\right).$$

This power spectral density can be computed using a process $\{X_i\}$ defined by $X_i = 1/2\cdot(A_i - A_{i-1})$. The relation between $S_X(D)$ and $S_A(D)$ is

$$S_A(D) = \frac{4S_X(D)}{2-(D+D^{-1})}.$$

We compute $S_X(D)$ based on the *one-step transition matrix* $\mathbf{G}(D)$ of the runlength state diagram of the code. The procedure to construct a runlength state diagram corresponding to a binary sequence is: 1) Insert intermediate states such that each edge is labeled by a single bit. 2) Perform state splitting so that all edges coming into (resp. leaving) a given state have the same label. 3) Perform state merging to simplify the resulting state diagram. State merging of two states can be performed if the output *edge structure* is identical. 4) Transform the state diagram into a runlength state diagram by removing all states with the property that the incoming bit (which is the same on all incoming edges) and the outgoing bit (which is also the same on all outgoing edges) are the same.

From [40], $S_X(D) = p(1)\pi\left((\mathbf{I} + \mathbf{G}(D))^{-1} + (\mathbf{I} + \mathbf{G}(D^{-1}))^{-1} - \mathbf{I}\right)\mathbf{u}^T$, where $\mathbf{u} = (1, 1, \ldots, 1)$, $(\cdot)^T$ denotes the transpose of its argument, $\mathbf{I}$ is the $L \times L$ identity matrix, L is the number of states in the runlength diagram, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$ is the solution to $\boldsymbol{\pi}\mathbf{G}(1) = \boldsymbol{\pi}$, $p(1) = (\boldsymbol{\pi}\mathbf{G}^{'}(1)\mathbf{u}^T)^{-1}$, and $\mathbf{G}^{'}(D)$ is the element-wise derivative of $\mathbf{G}(D)$ with respect to D.

The power spectral density is computed for the codes that are proposed from runlength constraints in Paper II, and two new codes are also introduced. Furthermore, for different codes, we compute the total power transferred to the tag as a function of the inter-coil separation based on Paper IV.

7.6 Paper VI

The sixth paper is entitled: *Error Correction on an Insertion/Deletion Channel Applying Codes From RFID Standards*.
In this work, we study the communication from a tag to a reader. Since CRC and Manchester codes are mandated by RFID standard protocols, we investigate their decoding performance under the discretized Gaussian shift channel in this work.

Our coding strategy is: the encoder structure of the tag is a serial concatenation of a CRC code as the outer code and a Manchester code as the inner code; decoding using a joint trellis for the overall serially concatenated code, and map the most likely transmitted frame to an information sequence using the encoder mapping of the Manchester code and that of the CRC code. In particular, we establish a metric table for the Manchester code on the insertion/deletion channel and use a *stack algorithm* to estimate the most likely transmitted frame with smallest Levenshtein distance to the received sequence by the help of a joint trellis structure of the overall code. We prove that the coding system is single error correcting for any information block length. The complexity of the stack decoder depends on a *bounded distance decoding threshold*.

8 Future Work

We exploited CRC-LDPC code concatenation on the BEC in Paper I. Similar exploitation of this concatenation on other channels can be an interesting study for future work. In Paper VI, we discussed CRC-Manchester code concatenation on a discretized Gaussian shift channel. Other modulation codes described in Paper II and Paper V can be considered as alternatives to the Manchester code in future work that optimize the code concatenation on this channel with respect to channel capacity and power transfer. Further, coding for multi-tags to single reader communication will be a challenging, but also a practical problem.

References

[1] Finkenzeller, K.: *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. John Wiley & Sons, 2010.

[2] Glover, B., Bhatt, H.: *RFID Essentials*. Theory in practice. O'Reilly, 2006.

[3] Zhang, L., Wang, Z.: Integration of RFID into wireless sensor networks: Architectures, opportunities and challenging problems. In *Proc. Fifth International Conference on Grid and Cooperative Computing Workshops (GCCW)*, pp. 463–469. Hunan,China, Oct. 2006.

[4] Griffiths, D. J.: *Introduction to Electrodynamics*. Prentice Hall, 1999.

[5] Horler, G.: Inductively coupled telemetry. In Baraton, M.-I. (ed.), *Sensors for Environment, Health and Security*, NATO Science for Peace and Security Series C: Environmental Security, pp. 239–252. Springer Netherlands, 2009.

[6] Proakis, J. G., Salehi, M.: *Communication Systems Engineering*. Prentice Hall, 2002.

[7] Goldsmith, A.: *Wireless Communications*. Cambridge University Press, 2005.

[8] Yoon, S.-R., Lee, J.-H., Park, S.-C.: Anti-collision protocol tuning for the ISO/IEC 18000-3 mode 2 RFID system. In *Proc. IEEE*

68th Vehicular Technology Conference (VTC), pp. 1–5. Calgary, BC, Canada, Sep. 2008.

[9] Tanenbaum, A. S.: *Computer Networks*. Prentice Hall, 1989.

[10] Mutti, C., Floerkemeier, C.: CDMA-based RFID systems in dense scenarios: Concepts and challenges. In *Proc. IEEE International Conference on RFID*, pp. 215–222. Las Vegas, NV, Apr. 2008.

[11] Cover, T. M., Thomas, J. A.: *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2006.

[12] Shannon, C. E.: A mathematical theory of communication. *Bell System Technical Journal* pp. 379–423, 623–659, Jul. and Oct. 1948.

[13] ten Brink, S.: Convergence of iterative decoding. *Electronics Letters* 35(10), 806–808, May 1999.

[14] ten Brink, S.: Convergence behavior of iteratively decoded parallel concatenated codes. *Communications, IEEE Transactions on* 49(10), 1727–1737, Oct. 2001.

[15] Ryan, W. E., Lin, S.: *Channel Codes: Classical and Modern*. Cambridge University Press, 2009.

[16] Levenshtein, V. I.: Binary codes capable of correcting deletions, insertions, and reversals. *SOVIET PHYSICS-DOKLADY* pp. 707–710, 1966.

[17] Rosnes, E., Barbero, Á. I., Ytrehus, Ø.: Coding for a bit-shift channel with applications to inductively coupled channels. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6. Honolulu, HI, Dec. 2009.

[18] Lin, S., Costello, Jr., D. J.: *Error Control Coding: Fundamentals and Applications*. Pearson-Prentice Hall, 2004.

[19] Roth, R.: *Introduction to Coding Theory*. Cambridge University Press, 2006.

[20] Forney, Jr., G. D.: The Viterbi algorithm. *Proceedings of the IEEE* 61(3), 268–278, Mar. 1973.

[21] Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on* 13(2), 260–269, Apr. 1967.

[22] Bahl, L., Cocke, J., Jelinek, F., Raviv, J.: Optimal decoding of linear codes for minimizing symbol error rate. *Information Theory, IEEE Transactions on* 20(2), 284–287, Mar. 1974.

[23] Wolf, J.: Efficient maximum likelihood decoding of linear block codes using a trellis. *Information Theory, IEEE Transactions on* 24(1), 76–80, Jan. 1978.

[24] Schlegel, C., Perez, L.: *Trellis Coding*. IEEE Press, 1997.

[25] Richardson, T. J., Urbanke, R. L.: *Modern Coding Theory*. Cambridge University Press, 2008.

[26] Bocharova, I. E., Handlery, M., Johannesson, R., Kudryashov, B. D.: A BEAST for prowling in trees. *Information Theory, IEEE Transactions on* 50(6), 1295–1302, Jun. 2004.

[27] Bocharova, I. E., Johannesson, R., Kudryashov, B. D., Lončar, M.: BEAST decoding for block codes. *Telecommunications, European Transactions on* 15(4), 297–305, Jul./Aug. 2004.

[28] Gallager, R.: Low-density parity-check codes. *Information Theory, IRE Transactions on* 8(1), 21–28, Jan. 1962.

[29] Tanner, R.: A recursive approach to low complexity codes. *Information Theory, IEEE Transactions on* 27(5), 533–547, Sep. 1981.

[30] MacKay, D. J. C.: Good error-correcting codes based on very sparse matrices. *Information Theory, IEEE Transactions on* 45(2), 399–431, Mar. 1999.

[31] Luby, M. G., Mitzenmacher, M., Shokrollahi, M. A., Spielman, D. A.: Improved low-density parity-check codes using irregular graphs. *Information Theory, IEEE Transactions on* 47(2), 585–598, Feb. 2001.

[32] Chung, S.-Y., Forney, Jr., G. D., Richardson, T. J., Urbanke, R.: On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *Communications Letters, IEEE* 5(2), 58–60, Feb. 2001.

[33] Hu, X.-Y., Eleftheriou, E., Arnold, D.: Regular and irregular progressive edge-growth Tanner graphs. *Information Theory, IEEE Transactions on* 51(1), 386–398, Jan. 2003.

[34] Wang, S. X., Taratorin, A. M.: *Magnetic Information Storage Technology*. Electromagnetism Series. Academic Press, 1999.

[35] Brian, H. M., Roth, R. M., Siegel, P. H.: *An Introduction to Coding for Constrained Systems*. Stanford, 2000.

[36] Immink, K. A. S.: Runlength-limited sequences. *Proceedings of the IEEE* 78(11), 1745–1759, Nov. 1990.

[37] Adler, R., Coppersmith, D., Hassner, M.: Algorithms for sliding block codes—An application of symbolic dynamics to information theory. *Information Theory, IEEE Transactions on* 29(1), 5–22, Jan. 1983.

[38] Immink, K. A. S., Siegel, P. H., Wolf, J. K.: Codes for digital recorders. *Information Theory, IEEE Transactions on* 44(6), 2260–2299, Oct. 1998.

[39] Immink, K. A. S.: *Coding Techniques for Digital Recorders*. Prentice Hall, 1991.

[40] Gallopoulos, A., Heegard, C., Siegel, P. H.: The power spectrum of run-length-limited codes. *Communications, IEEE Transactions on* 37(9), 906–917, Sep. 1989.

[41] Bosik, B. S.: The spectral density of a coded digital signal. *Bell System Technical Journal* 51(4), 921–932, Apr. 1972.

[42] Rosnes, E., Ytrehus, Ø.: Turbo decoding on the binary erasure channel: Finite-length analysis and turbo stopping sets. *Information Theory, IEEE Transactions on* 53(11), 4059–4075, Nov. 2007.

[43] Weber, J. H., Abdel-Ghaffar, K. A. S.: Results on parity-check matrices with optimal stopping and/or dead-end set enumerators. *Information Theory, IEEE Transactions on* 54(3), 1368–1374, Mar. 2008.

[44] Wei, V. K.: Generalized Hamming weights for linear codes. *Information Theory, IEEE Transactions on* 37(5), 1412–1418, Sep. 1991.

[45] Rosnes, E., Ytrehus, Ø.: An efficient algorithm to find all small-size stopping sets of low-density parity-check matrices. *Information Theory, IEEE Transactions on* 55(9), 4167–4178, Sep. 2009.

[46] Barbero, Á. I., Horler, G. D., Rosnes, E., Ytrehus, Ø.: Modulation codes for reader-tag communication on inductively coupled channels. In *Proc. International Symposium on Information*

Theory and its Applications (ISITA), pp. 578–583. Auckland, New Zealand, Dec. 2008.

[47] Rosnes, E., Barbero, Á. I., Ytrehus, Ø.: Coding for inductively coupled channel. *Information Theory, IEEE Transactions on*, to appear.

[48] Grover, P., Sahai, A.: Shannon meets Tesla: Wireless information and power transfer. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pp. 2363–2367. Austin, TX, Jun. 2010.