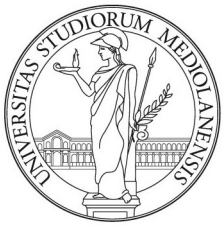
UNIVERSITÀ DEGLI STUDI DI MILANO

GRADUATED SCHOOL OF COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE



DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE
XXVIII CYCLE

A New System to Increase the Life Expectancy of Optical Discs. Performance Assessment with a Dedicated Experimental Setup.

INF/01

DOCTORAL DISSERTATION OF:
Ciro Polizzi

ADVISOR:
Prof. Goffredo Haus

CO-ADVISOR:
Prof. Raffaele G. Della Valle

DIRECTOR OF GRADUATED SCHOOL:
Prof. Paolo Boldi

Academic Year 2015 – 2016

Abstract

The past decade has witnessed an exponential growth in the use of digital supports for big data archiving. However, the expected lifespan of these supports is inadequate with respect to the actual needs of heritage institutions. Stemming from the issues raised by UNESCO [1, 2], we address the problem of alleviating the effects of aging on optical discs. To achieve this purpose, we propose a novel logical approach that is able to conteract the physical and chemical degradation of different types of optical discs, increasing their life expectancy.

In other words, the objectives of this thesis are the design and the implementation of a new intelligent strategy aimed at increasing the life expectancy of optical discs. An experimental setup has been developed in order to investigate the physical and chemical degradation processes of discs, by means of accelerated aging tests that simulated the operational disc-use conditions. *Critical areas* are identified where disc degradation is statistically faster than average, while in *safe areas* the degradation is relatively slow. To collect the needed data, two experimental devices have been built. The first is a climatic chamber, which is able to induce artificial disc aging. The second is a robotic device, which is able to detect the amount of errors in each data block prior to the "Reed Solomon" error correction stage (before and after the accelerate aging stage, without dust and in an environment with controlled temperature and humidity). The analysis of the data allows to identify the aforementioned physical areas where data blocks have a number of errors that approaches or exceeds the data correction capability of the standard Reed Solomon code.

The results of these analyses have led to develop an "Adaptive Reed Solomon Code" (A-RS code) that allows to protect the information stored within the *critical areas*. The A-RS code uses a redistribution algorithm that is applied to parity symbols. It is calculated from the fitting of the experimental errors obtained through the "degradation function". The redistribution algorithm shifts a certain number of parity symbols from Data Blocks in *safe areas* to Data Blocks in *critical areas*. Interestingly, these processes do not diminish the memory capability of Digital Versatile discs (DVDs) and of Blu-Ray discs (BDs). This strategy therefore avoids the early dismission of optical discs, due to possible losses of even minimal parts of the recorded information.

INDEX

CHAPTER 1

1 Introduction

Three generations of optical discs appeared in the market in the last 30 years. The reduction of the laser wavelength λ from Compact Discs (CDs) with $\lambda = 780nm$ to Blu-Ray discs (BD) with $\lambda = 405nm$) and the increase of the numerical aperture (Na) of the lens (from CDs with $Na = 0.45$ to Blu-Ray discs with $Na = 0.85$) allowed the reduction of the laser spot dimension and, as a consequence, the enhancement of the information density. *Fig.1.1* shows the dimension of the laser spot in the case of CDs, Digital Versatile Discs (DVDs) and BDs. The technological progress could be further improved (i.e. the latest storage system "Optical Disc Archive" developed by Sony which is based on high capacity BD (300 Gbytes) and can register on a single cartridge (Generation-2) up to 3.3 Tbyte).

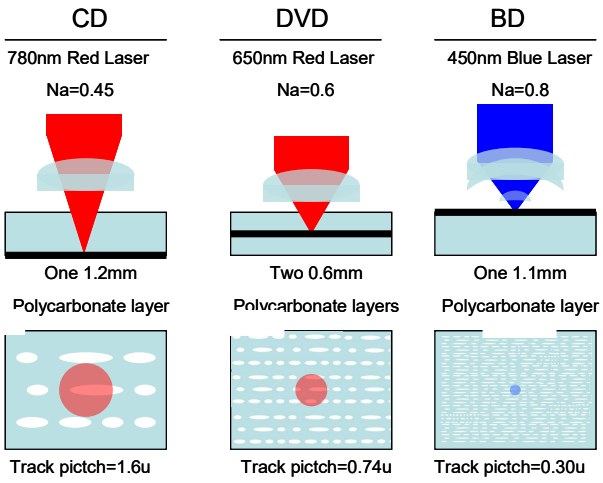


Fig.1.1 - Laser spot dimensions and track pitched for CDs, DVDs and BDs.

1.1 Optical Writing and Reading Systems

During the burning process the data bits are imprinted, with a high power laser beam, onto the optical disc in the form of marks of various lengths. These marks create a minimum reflectivity zone, called pit, that is different from the maximum reflectivity zone, called land. During the reading process the disc surface is scanned by a low power laser beam, and the *reflected light* is measured by a *photodetector*. The reading system recognizes the signal patterns formed by pits and lands and recovers the original information.

The write/read process of an optical disc can be considered as a transmitter/receiver system of a digital communication system [3] whose transmitter channel, with a time delay, is the disc.

Fig.1.2 illustrates the block diagram of this kind of system.

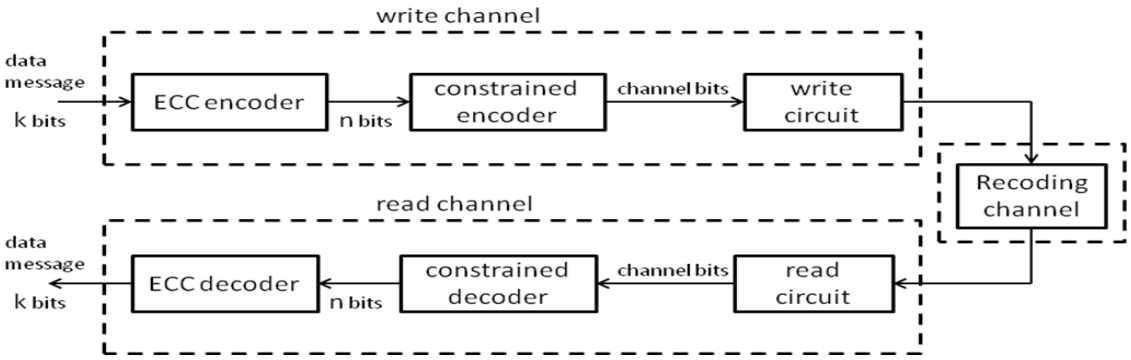


Fig.1.2 - Write/ Read system of an optical disc.

The system consists of three parts: the write channel (transmitter), the read channel (receiver) and the recording channel (the disc itself). The write channel converts the input binary data into a suitable form for writing onto the storage medium. The read channel recovers the original data by processing the output of the optical head in accordance with the appropriate algorithms.

In what follows a brief description of the write and read channels is given.

The error correction encoder (ECC) adds extra parity symbols to the data stream. These parity symbols are devoted to enable the ECC decoder to detect and correct errors in the data. There are many different types of ECC codes, [4, 5] among which the Reed-Solomon codes [6] are widely used. Reed-Solomon codes are a subset of *Bose Chaudhuri-Hochquenghem* (*BCH*) codes and are *linear block* codes. They are powerful error-correcting codes whose symbols are chosen from a finite field, [7] $GF(p^m)$ where p is prime (usually $p=2$) and m is an integer. A Reed-Solomon code is specified as RS(n,k). The encoder takes k data symbols of m bits each and adds parity symbols to make a code-word with at most n symbols where $n=2^m-1$. Therefore, in a code-word there are $n-k$ parity symbols of m bits each. A RS decoder can correct up to t symbols that contain errors in a code-word, where $t = (n-k)/2$.

Constrained codes [8] are used to improve the detector performance and to help in the operation of control loops at the receiver. A constrained code is characterized by the so called *(d,k)-runlength-limited (RLL)* constraint, where d,k are integers. The encoder imposes two conditions at the finite length binary sequence:

1. the binary sequences of *0*'s have a maximum length k
2. the binary sequences of *0's* between successive *1's* have a minimum length d.

In a binary transition-sequence the *d-constraint* reduces the effect of inter-symbol interference and the *k-constraint* is devoted to synchronize the time control.

The write circuit converts the constrained coded data into a write-current waveform on the storage medium by laser.

The read circuit collects the reflected light from the photo detector and converts it to a suitable signal.

Both constrained and ECC decoders operate on the output of the detector to provide an estimate of the original input to the recording system.

1.2 Optical Disc Reliability

The memory life-expectancy of a disc defines its reliability. In fact, the ISO/IEC 18927 states that the end-life of an optical disc is reached when the number of errors, in a single data block, exceeds the maximum number of errors correcting capability of the ECC. Many manufacturers advertise the "long-life" of their products, whereas life-expectancy between 1 and 10 years are estimated by several studies [9, 10, 11].

The optical disc is a removable storage (mainly made with polycarbonate), without electronic, electrical, and mechanical components. Optical discs are water resistant and insensitive with respect to magnetic fields. As a consequence, the disc deterioration is mainly due to physical and chemical degradation of the materials.

By testing on some CD-Rs and DVD-Rs recorded more than ten years ago I have found that the disc deterioration increases while moving towards the outer areas of the disc. *Fig. 1.3* shows an error test of a CD-R more than10 years old.

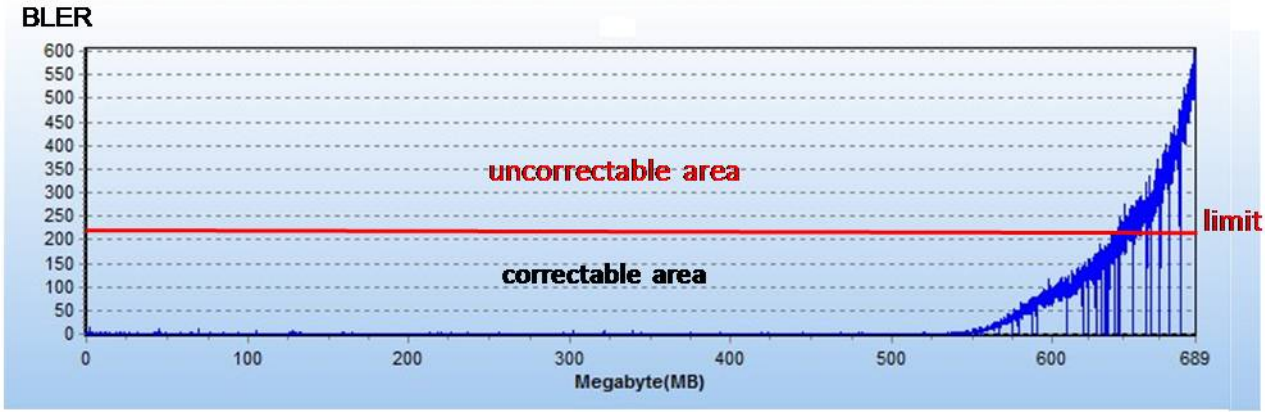


Fig.1.3 - Error test of a 10 years old CD-R. Block Error Rate (BLER) is the ratio of erroneous blocks to total blocks measured at the input of the RS decoder. Limit is the maximum number of errors correction capability.

This phenomenon suggests that system of errors correction within the burning devices (which

uniformly distributes the parity symbols) is not efficient. Actually, the RS decoder uses only a small portion of the parity symbols to correct the data written in the slightly degraded areas of the disc. However, in general, the RS decoder has not enough parity symbols to correct the errors in the most degraded areas.

1.3 Advantages of Adaptive RS codes (A-RS)

We suggest to use an Adaptive Reed Solomon (A-RS) code [12, 13] which, if implemented in the usual burning devices, could provide three new functionalities:

1 Add extra parity symbols, in Adaptive mode (A-RS), according to a suitable *degradation function*.

2 Allow the customer to define the reliability of the disc in Tunable mode (T-RS).

3 A combination of 1 and 2.

Difference between a uniform RS code and an Adaptive-RS code is illustrated in *Fig.1.4*.

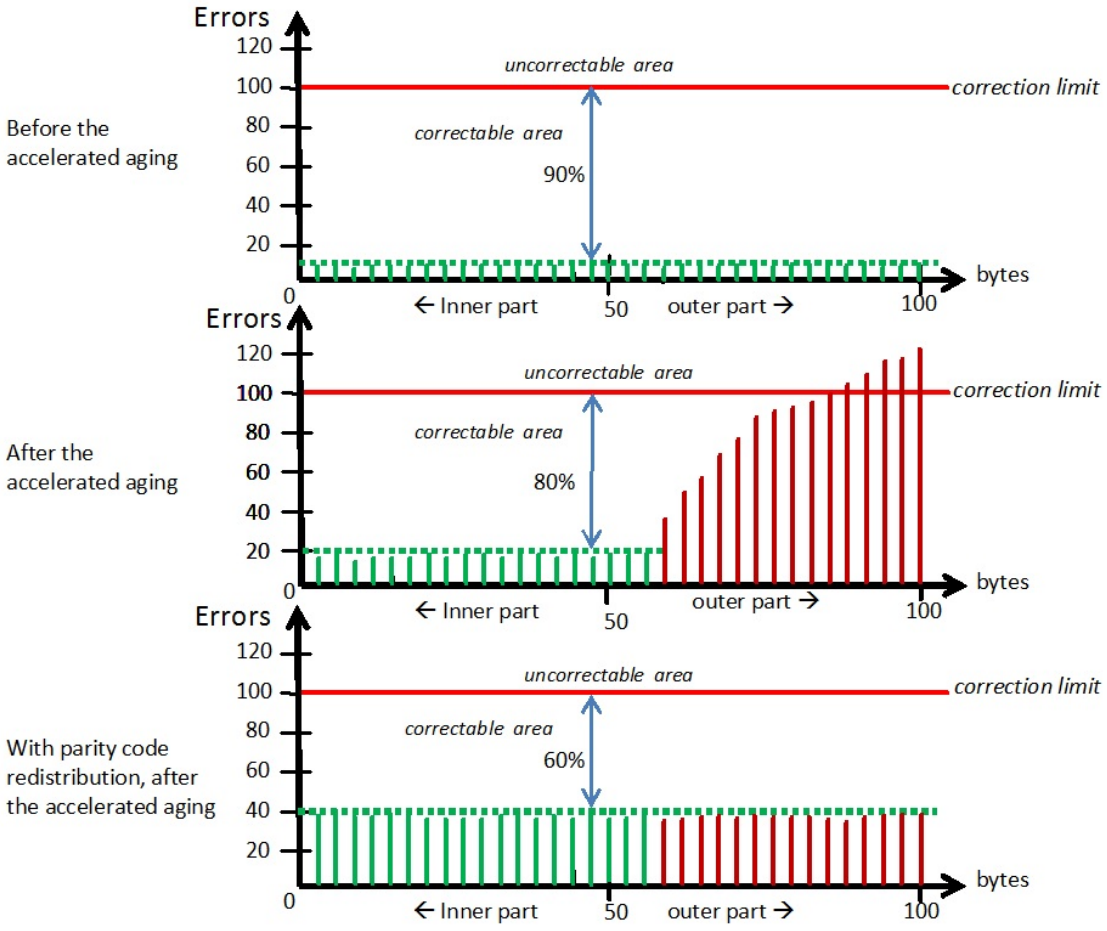


Fig.1.4 – Differences between an uniform RS code and an Adaptive RS code. From the top: errors in the disc immediately after burning, errors in the disc with uniform RS code after accelerate aging, errors in the disc with Adaptive RS code after accelerate aging.

A different parity symbols distribution will allow the reduction of the effects of the physical chemical degradation of the materials on the information data and prevent the early end-life of the disc.

1.3.1 A-RS code for Blu-Ray Discs

In order to develop an A-RS code suitable for optical device system, the best solution is to keep the code-word length *n* fixed and to change the number of parity symbols *n-k*. Obviously, if we increase the number of code-word parity symbols we will have less information symbols *k* for each code-word.

The RS code in the BD [14] is *RS(248,216)* where each code-word of *n=248 bytes* length contains *k=216 bytes* information data and *n-k = 32 bytes* parity symbols. The maximum number of symbols (or bytes) the RS decoder can correct in each code-word is *t=(n-k)/2 =16 bytes* so *2t=n-k.*

The Block Correction (BC) defines the percentage of amendable symbols over the total number of symbols in each code-word. For the Blu-Ray discs BC=*t/n=16/248=6,45%.*

As it will be shown, by adopting some changes to the data encoding system the Blu-Ray burning device and substituting the uniform RS code with an A-RS code, it is possible to continuously increase the BC value from 6,45% up to 25%. *Fig.1.5* illustrates a Blu-Ray write/read system with A-RS code.

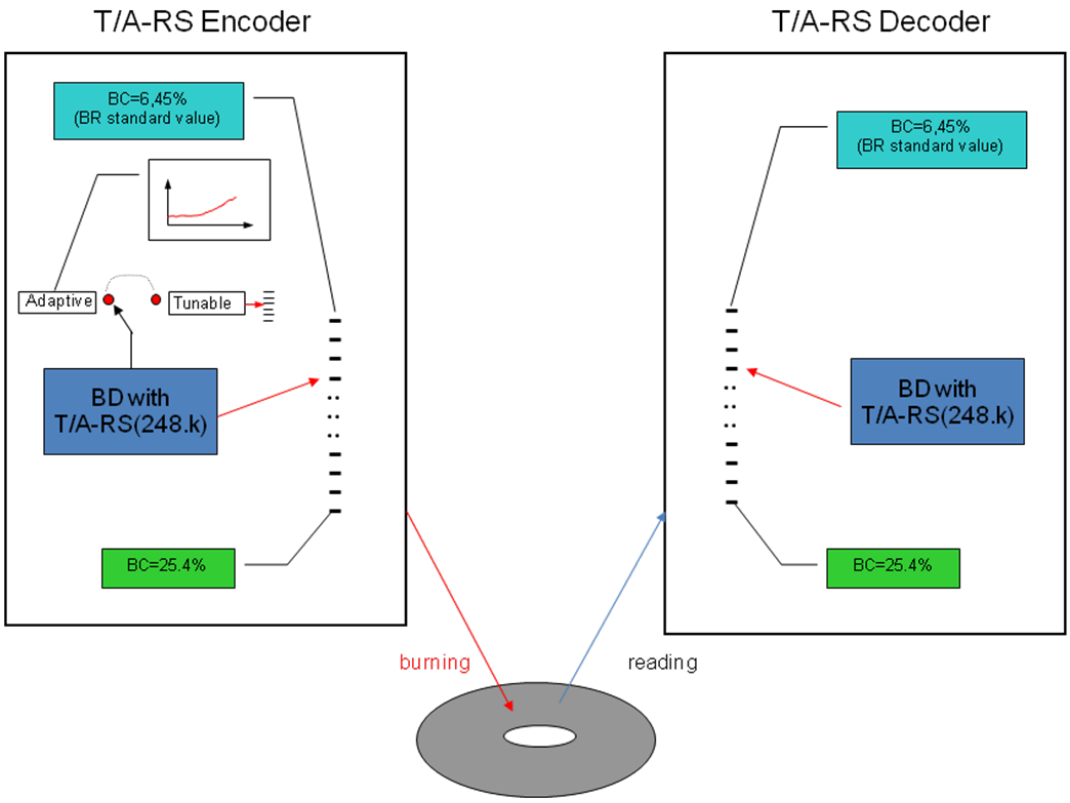


Fig.1.5 - Write/read system of Blu-Ray discs with Tunable/Adaptive RS code.

Fig. 1.6 shows the amount of data information compared to the different parity symbols within the A-RS (in *tunable mode*) in a Blu-Ray disc.

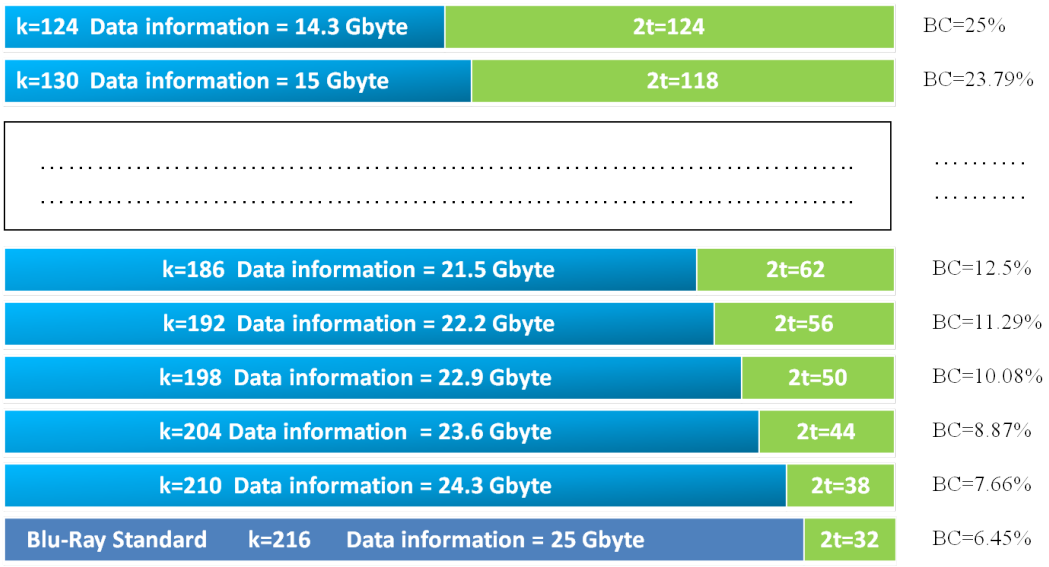


Fig.1.6 - Amount of data compared to the different parity symbols with a T/A-RS in a Blu-Ray disc.

With an A-RS the amount of information of the discs does not change because the parity is redistributed between Data Blocks according to a "*degradation function*".

1.4 Experimental Part: Overview

To collect the information necessary to define the degradation function to be used for the adaptive distribution of the A-RS parity symbols, several discs have been submitted to an accelerated aging test. The amount of degradation on each disc was evaluated as the difference between the number of errors present before and after the artificial aging.

Using the equipments provided by electronic laboratory of the Department Industrial Chemistry "Toso Montanari" (University of Bologna) and "mechanical laboratory" of the Department of Industrial Engineering (University of Bologna) the candidate has designed and built two devices:

1. Climatic Chamber
2. Optical Discs Automatic Testing Machine

1.5 State of the Art

1.5.1 Backup and Archiving

There is often confusion between a data archive and a backup [15]. Essentially, a backup is designed as a short-term insurance policy to facilitate disaster recovery, while an archive is designed to provide ongoing rapid access to decades of business information.
A backup application takes periodic images of active data in order to provide a method for recovering records that have been deleted, corrupted, or destroyed. Most backups are retained only for a few days or weeks as later backup images supersede previous versions.
Archived records, instead, are placed outside the backup cycle for long periods of time. An effective data archiving strategy is a necessary part of every IT organization.

1.5.2 Data Storage Technologies for Archiving

To date, the most commonly employed means of data storage are magnetic disks, optical discs, magnetic tapes, and solid state disks. Digital information does not have the same life expectancy of physical media (as for example books and other written documents). This remark applies to all the aforementioned digital types of storage, and therefore it is important to set early intervention measures to ensure long-term preservation of the data.
According to the guidelines submitted by the UNESCO/PERSIST Content Task Force in 2016 [1, 2], it is fundamental for heritage institutions to adapt their existing approaches to digital environments, meaning that the quantity of digital data will constantly increase over time. However, since digital data storages can be compromised a periodic media refresh is required - by reading the digital data, checking for errors, and rewriting on new media. Due to this, the knowledge of the mean life expectancies of different types of data storages is fundamental, and the topic has been deeply investigated. For example, data stored on magnetic tapes may still be accessible after 10 to 30 years, **[16]** yet it is known that this type of support has many limitations **[17]**. For magnetic hard disks the estimated life expectancy is within 5-7 years **[18, 19]**. For Solid State disks the life expectancy is even lower, of about 4 years **[20, 21]**. Instead, the life span of optical discs is estimated not to be longer than 10 years **[9, 10, 11]**.
Still, it is important to remember that different types of supports belonging to the same group may display different and reduced life spans, due to manufacturing technologies, to different external environment, to different quality and nature of the materials,. Therefore, it is difficult to foresee whenever a storage media will break and lose data. To overcome this problem it should be

necessary to perform extensive statistical analyses on large numbers of specimens, [22, 23] with a significant investment in terms of time and economical resources.

1.5.3 Latest Development and Applications of Optical Discs

Despite what may appear, technologies related to optical supports are far from obsolete. To date, notable efforts are still focused in this field. For example, Sony in collaboration with Panasonic has recently presented new Optical Disc Archives based on high capacity Blu-Ray discs: Generation 1 (dated 2014) has cartridges with a storage capacity of 1.5 TB, Generation 2 (dated 2016) has cartridges with a storage capacity of 3.3 TB, and Generation 3 (still in development) should have cartridges with a storage capacity of 5.5 TB [24].

However, it is important to remember that as data volumes rise, so do storage costs. Therefore, it would be essential to implement storage systems that distinguish between "*hot*" (frequently accessed), "*warm*" (occasionally accessed), and "*cold*" (infrequently accessed) data, selecting the best type of storage media solution for each category (*see Fig.1.7*)

Hot storages demand high performances, and are therefore best implemented with high-cost memories, as for example flash memories or SSDs. For what concerns *warm* data and *cold* data, the performances are less important. What is fundamental to achieve are a long-term reliability, the ability to maintain large quantities of data at relatively low costs, and the ability to maintain data integrity with limited environmental controls. Sony is convinced that optical discs storages could provide all of these requirements, and are therefore the ideal type of support for *warm* storages and *cold* storages. [24]

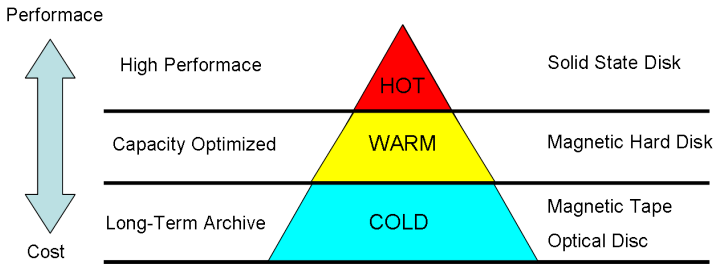


Fig. 1.7 - Hierarchy in Storage Systems

Optical discs are the most widely used medium for *cold* storages. Since there is backwards compatibility between new-generation disc systems and older, "conventional" optical discs, the need for data migration is essentially eliminated when considering optical discs. Moreover, discs are usually quite durable, water-resistant, and allow for random access.

Magnetic tapes are another widely used medium for *cold* storages. Magnetic tapes offers relatively high recording densities, and relatively low cost per unit of capacity. However, random access is not suitable, and magnetic tapes are also susceptible to damage when exposed to high humidity, water, or electromagnetic waves.

As one can see, optical discs and magnetic tape use entirely different technologies. While each has its own benefits, optical discs offer clear advantages when it comes to long-term *cold* storages.
The aforementioned Sony's Archival Discs are one of the newest implementation of Blu-Ray technologies, in terms of performances and reliability. A notable application of these supports is Everspan, the first producer of Optical Archives [24]. The total capacity of a single Everspan robotic system can reach 181 Petabytes, and up to four systems can be linked to offer 724 Petabytes of total addressable storage.
Facebook, for example, has noticed that only about the 8% of its records is continuously used, while the remainder is seldom, if ever, used. In 2014 -2016 these could records were moved to an Everspan system consisting of 10.000 BDs that can store 1 Petabyte of data. In this way Facebook has halved the cost and has improved the energy usage by the 80%, compared to the previous system that was based on magnetic supports [25].

1.5.4 RS codes: Current Applications

All data storage technologies and most digital communication systems use Forward Error Correction (FEC) systems. These protocols allow to protect data before their storage on physical devices, or before their transmission. Usually, FEC systems add redundancy to the information, that can be later used by the decoding systems to detect and correct errors that were generated during the writing and the reading tasks, or those due to the noise caused by the transmissive media. In particular, most FEC systems are based on Reed-Solomon codes.
Reed-Solomon codes are a group of error-correcting codes that were introduced by Irving S. Reed and Gustave Solomon in 1960.[CIT] In coding theory, the Reed-Solomon code belongs to the class of non-binary cyclic error-correcting codes. The Reed–Solomon code is based on univariate polynomials over Galois fields. They have many applications, the most prominent of which include consumer technologies such as CDs, DVDs, Blu-ray Discs, QR Codes, data transmission technologies such as DSL and WiMAX, broadcast systems such as DVB and ATSC, and other storage systems. More in detail, they can be employed in:

- storage devices (compact disk, DVD, Blu-Ray, barcodes,etc….) [26, 27],

- wireless and mobile communications (including cellular telephones, microwave links, etc…) [28, 29],
- digital satellite communications [30],
- digital television, digital video broadcasting (DVB) [31],
- high speed modem such as ADSL [32],
- power line communications (PLC) [33],
- digital vestigial sideband (VSB) system [34],
- cable modem [35].

1.5.5 A-RS codes: Current Applications

RS codes have been used as bases for the coding of Adaptive Reed-Solomon codes (A-RS codes). A-RS codes are useful as they can provide variable redundancy to the information [12]. These codes are versatile, due to the fact that different code rates and hence variable correcting capabilities can be applied. In a conventional Forward Error Correction FEC [36] technique, where the error-correcting capability is fixed, the design principle is to obtain the desired average performance under the worst-case channel environments. However, in many applications the channel remains in its worst state during only a small fraction of time. So, it results in low efficiency, and can not meet the varying requirements. On the contrary, a A-RS code can provide a flexible trade-off between transmission efficiency and power consumption [37].

A-RS codes can be used for a wide set of systems. The redundancy increase/decrese is dynamically fixed between the transmitter and the receiver, depending on the noise level of the transmissive channel. For example, they are employed in:

- ATM communication systems [38],
- WiMax Networks [39],
- DVB and wireless technologies [40],

By considering these applications, it is possible to notice that, to date, they all are refer communication systems. This is due to the fact in all cases the A-RS codes are designed to respond to the channel conditions. This observation suggests a possible way for implementing the same approach for physical supports as optical discs. At this purpose we need to predict *a priori* in which physical locations a higher degree of redundancy would be needed.

CHAPTER 2

2. Equipment and Tools

To experimentally evaluate the degradation in time of optical supports, two devices have been designed, engineered, and developed in both their hardware and software components. The motivation behind the choice of building our own machines has been to ensure that only the effects due to the aging of the disc materials are taken into account. Moreover, these customisations are significantly more efficient with respect to our scopes, and more affordable, than any available technology currently available, at least to the knowledge of the author. The candidate has built two devices:

1. A climatic chamber, used to induce on the discs an accelerated aging process comparable to a natural aging process.
2. An Automatic Testing Machine (ATM), developed to automatically perform the analyses on 50 discs at a time, before and after having performed the accelerated aging process. Furthermore, the ATM ensured the integrity of the experimental protocol. Both aging and burning tasks have been performed in a controlled, dusts-free environment, and without any manual operator.

Both machines were based on two Open Source projects:

1. Arduino Uno [41, 42] (*Fig. 2.1*): It is a microcontroller board based on the ATmega328P. Arduino has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, 32 Kbyte EEPROM and USB connection.
   Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) used to write code and execute it on the physical board. The programming language is a simplified version of C++.

2. Theremino [43] (*Fig. 2.2*): It is a 16-bit Flash Microcontroller with USB connection. Unlike Arduino, through *real time commands* sent from a computer, it can write and read both analogue and digital signals. Thanks to the modularity of the system Theremino, HAL (Hardware Abstraction Layer), the slots (Inter process Communication), the UDP

(Communications via network and Internet) and the OSC Protocol (Open Sound Control), all components can communicate with each other, in a particularly efficient way.

Fig. 2.1 - Arduino Uno R3 Board.

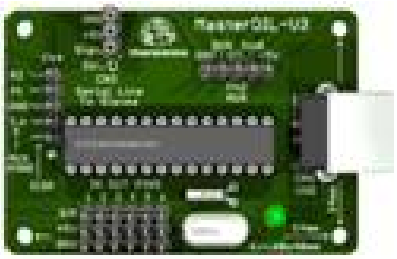


Fig. 2.2 – Theremino Board.

2.1 Climatic Chamber

The climatic chamber [44] consists of a steel box with internal dimensions *505x300 mm*. Its weight is *2.24 kg*, thickness *0.8 mm* and capacity *26.5 l*. The steel cover box has weight *1,12 kg*, dimensions *530x325mm* and thickness *0.7mm*. A gasket cover hermetically closes the chamber. The chamber is equipped with a safety valve to prevent over-pressurisation. Finally, the steel box is placed inside a thermally insulated container which decreases the possible heat losses. There are many electronic circuits, transducers, sensors, and even hardware systems whose functions can be controlled and programmed by using Arduino board (see *Fig. 2.3 and 2.4*).

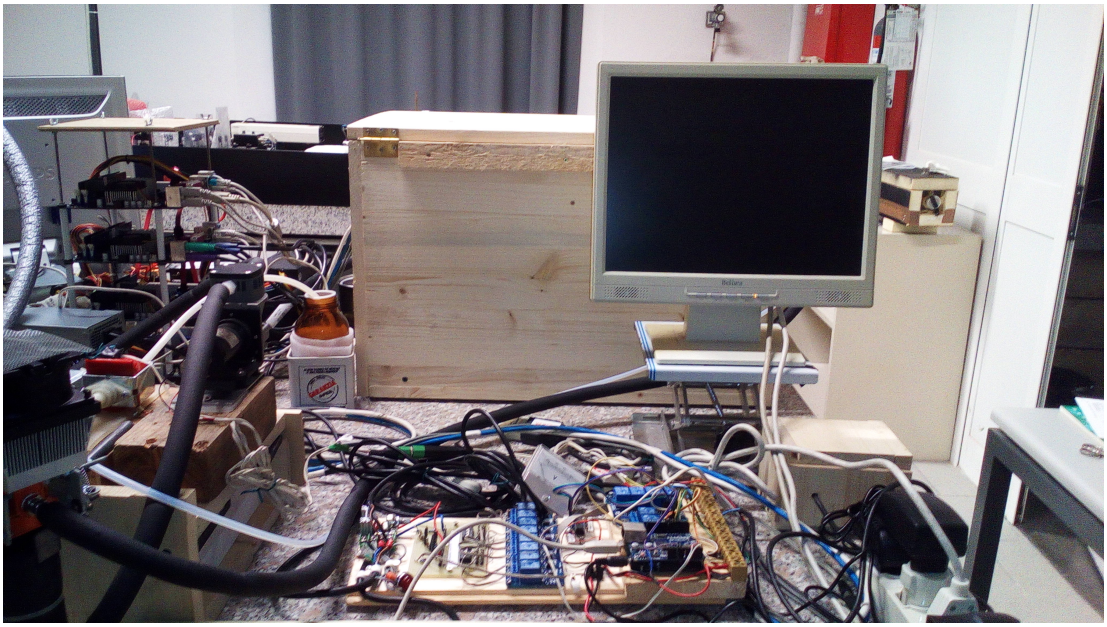


Fig.2.3 - Picture of the Climatic Chamber.

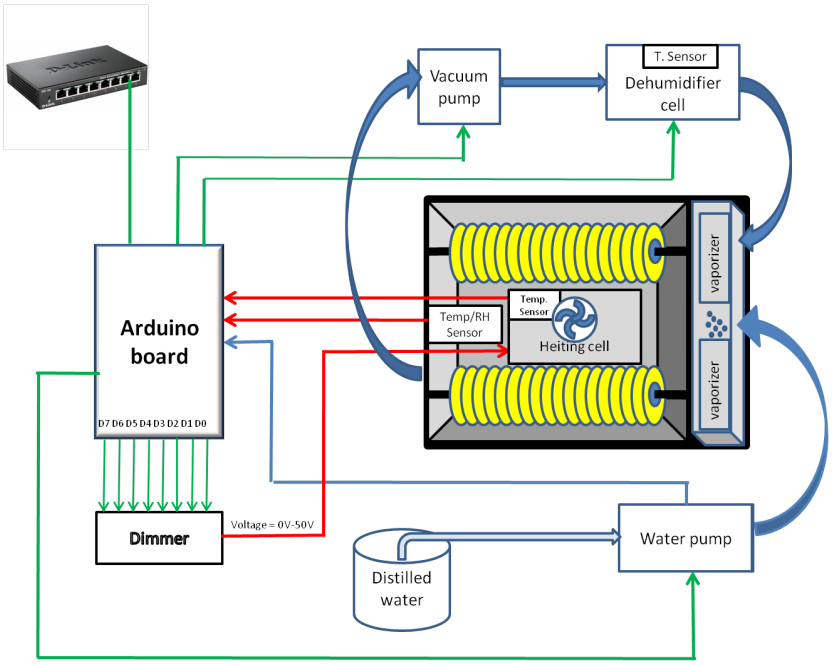


Fig.2.4 - Block diagram of the Climatic Chamber.

- The heating system is composed by an electrical resistor (*900W*), which is placed inside an aluminium plate (*400x 250 x 20 mm*) called heating cell.
- A type-k thermocouple records the temperature of the plate. The electric resistor is powered by a dimmer, that allows to tune the voltage that is applied to the resistor itself, and therefore the emitted heat. The dimmer is controlled by a binary sequence of 8 bits, through the software of the Arduino board. The air temperature within the chamber is registered by an electronic sensor.
- The humidity system is formed by a tank of distilled water, a water pump, and a nebulizer. The water pump transports the water from the container to the chamber through the nebulizer.
- The dehumidity system is formed by a vacuum pump and a dehumidifier cell.
  The dehumidifier cell is composed of a vacuum filter with a Peltier cell [45], and a cooler. The vacuum pump takes the air from the chamber and transfers it through the vacuum filter. The Peltier cell generates a thermal jump that condensates the humidity (at dew temperature) [46]. The resulting air is quickly cooled and therefore humidity is dropped in the water tank in liquid form. At last, the dried air goes into the chamber.

2.1.1 Relative Humidity Control System

The dehumidifier cell, as well as the humidifier cell, operates in a ON/OFF modality (*Fig. 2.5*) [47]. The control is activated when the registered value rises over a set threshold (*i.e.* 85%), and acts by

interrupting the dehumidification process. The ON/OFF control operates with a maximum error equal to ±1%.

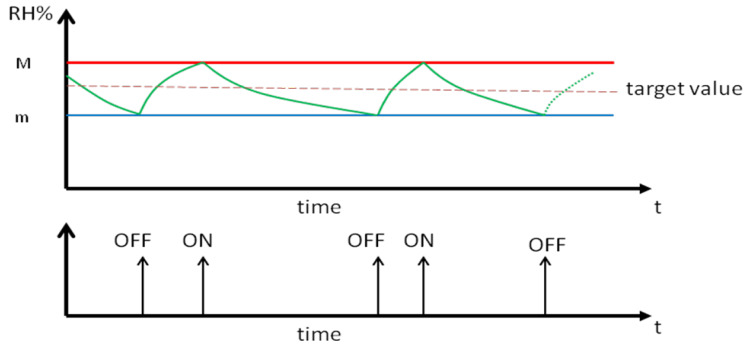


Fig. 2.5 – ON/OFF Dehumidity Control System.

2.1.2 Temperature Control System

The software developed to control the temperature uses a system called PID (Proportional-Integral-Derivative) [48, 49] *see Fig.2.6*. The error signal, E, is equal to the difference between the *set reference value* (the temperature that has to be reached) and the actual of the temperature. This signal is elaborated through three blocks, one proportional, one integral, and one derivative.

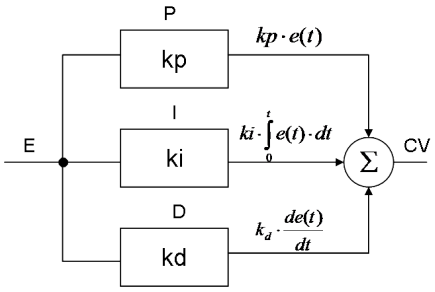


Fig. 2.6 – PID System Control.
In each block the error signal is evaluated under a different operator.

In proportional regulators the output is linearly related to the input:

$u(t) = k_p \cdot e(t)$,

where k_p is the proportionality constant equal to the transfer function of the regulator.
In integrative regulators, the output is equal to the summation of a function over a given interval. In this case is the sum of error over time is equal:

$u(t) = k_i \cdot \int_0^t e(t) \cdot dt$ where k_i is the integral constant.

In derivative regulators the output is given by the derivative of the error with respect to time:

$$u(t) = k_d \cdot \frac{de(t)}{dt}$$ where k_d is the derivative constant.

Therefore, a PID regulator has all these three types of regulators, combined as follows:

$$u(t) = k_p \cdot e(t) + k_i \cdot \int_0^t e(t) \cdot dt + k_d \cdot \frac{de(t)}{dt}$$

The proportional, integral, and derivative terms are summed in order to calculate the output of the PID controller.

All three of these PID controller components create output based on measured error of the process being regulated. If a control loop functions properly, any changes in error caused by set reference changes or process disturbances are quickly eliminated by the combination of the three terms P, I, and D (i.e k_p, k_i, *and* k_d).

2.1.3 Stress Conditions

The control system that operates over humidity and temperature has been programmed by following the International Standard guidelines [23], by means of the Arduino_E software.

The stress level that has been used for the accelerated aging tests of the discs is reported in the following *Table 2.1*:

Test stress T_{inc}/RH_{inc}	Incubation duration (h)	Minimum equilibration duration (h)
80°C/85% RH	500	6

Table 2.1- Stress conditions.

The process that is described in the International Standard guidelines requires a transition from ambient conditions to stress condition, and back. The transition duration (ramp) and its operative conditions have been chosen in order to allow a sufficient equilibration of adsorbed moisture by the substrate. Large departures from equilibrium conditions may result in the build-up of condensation inside the substrate, or at the interface between the substrate and the information-recording layer. Gradients in the water concentration through the thickness of the substrate shall also be limited, because they can cause significant disc curvature.

In order to minimize the effects of moisture-concentration gradients, the ramp profile outlined in *Table 2.2* shall be used. The objectives of the profile are:

- Avoidance of any situation that may cause moisture condensation within the substrate;
- Minimization of the time during which substantial moisture gradients exist in the substrate;

- Production, at the end of the profile, of a disc that is sufficiently equilibrated to proceed directly to testing without delay.

Process step	Temperature °C	Relative humidity %	Duration h
Start	at T_{amb}	at RH_{amb}	—
T, RH ramp	to T_{inc}	to RH_{int}	1,5 ± 0,5
RH ramp	at T_{inc}	to RH_{inc}	1,5 ± 0,5
Incubation	at T_{inc}	at RH_{inc}	500 ± 0,5
RH ramp	at T_{inc}	to RH_{int}	1,5 ± 0,5
Equilibration	at T_{inc}	at RH_{int}	6 ± 0,5
T, RH ramp	to T_{amb}	to RH_{amb}	1,5 ± 0,5
End	at T_{amb}	at RH_{amb}	—

Table 2.2 – Stess conditions. T_{amb} and RH_{amb} are room ambient temperature and relative humidity. T_{inc} and RH_{inc} are the stress incubation temperature relative humidity.[23]

The profile accomplishes this by varying the moisture content of the disc only at the stress incubation temperature, and allowing sufficient time for equilibration during ramp-down based on the diffusion coefficient of water in polycarbonate.

Fig. 2.7 graphically portrays the temperature and humidity changes that would occur during one cycle of incubation.

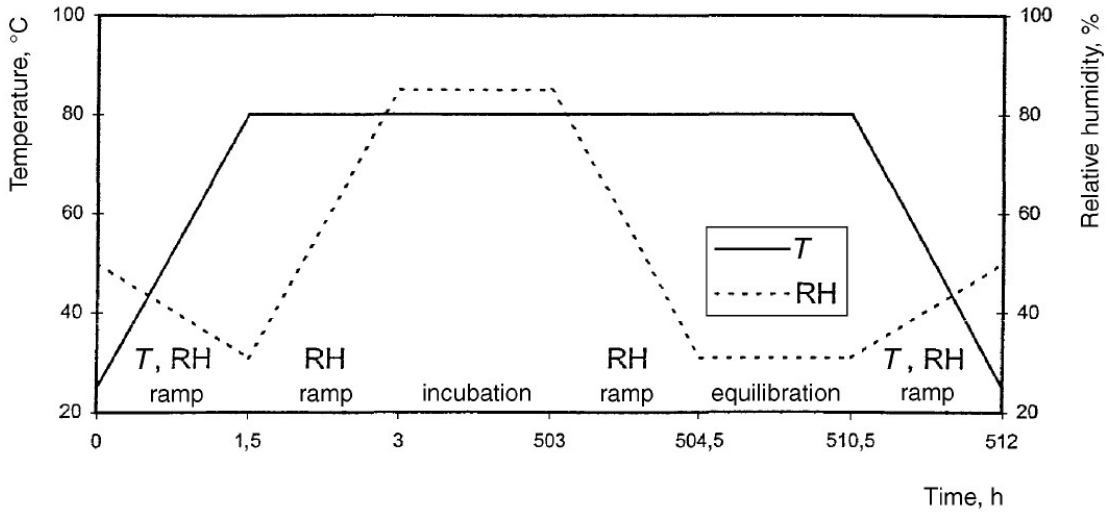


Fig. 2.7 - Graph of nominal 80°C / 85% RH transition.[23]

The left ramp specifies the increase of the temperature and humidity values when the testing starts. The right ramp specifies the decrease of the temperature and humidity values when the testing ends.

When the testing starts, it is necessary to increase the humidity after that the temperature reaches 80°C. When the testing ends, it is necessary to decrease the humidity before the temperature drops to external ambient temperature, to prevent condensation on the disc surface.

2.2 Automatic Testing Machine (ATM)

An automated testing machine (ATM) has been one of the cores of this work, because it allows to test several optical discs in a clean environment (to avoid the errors that could be triggered by the presence of dust [50]), with controlled temperature and relative humidity. This machine is necessary for burning and checking discs, and is also used to detect errors produced by chemical and physical degradation of the materials. In addition, the machine considerably reduces the time needed for the tests and it is able to keep track of each disc during the steps of the testing procedure. The temperature and humidity control allows one to perform the tests with identical environmental conditions, so that the laser works with the same light power on each disc [51]. In *Fig.2.8* the block diagram of the system is illustrated while a picture of the testing system is presented in *Fig.2.9 – 2.10.*

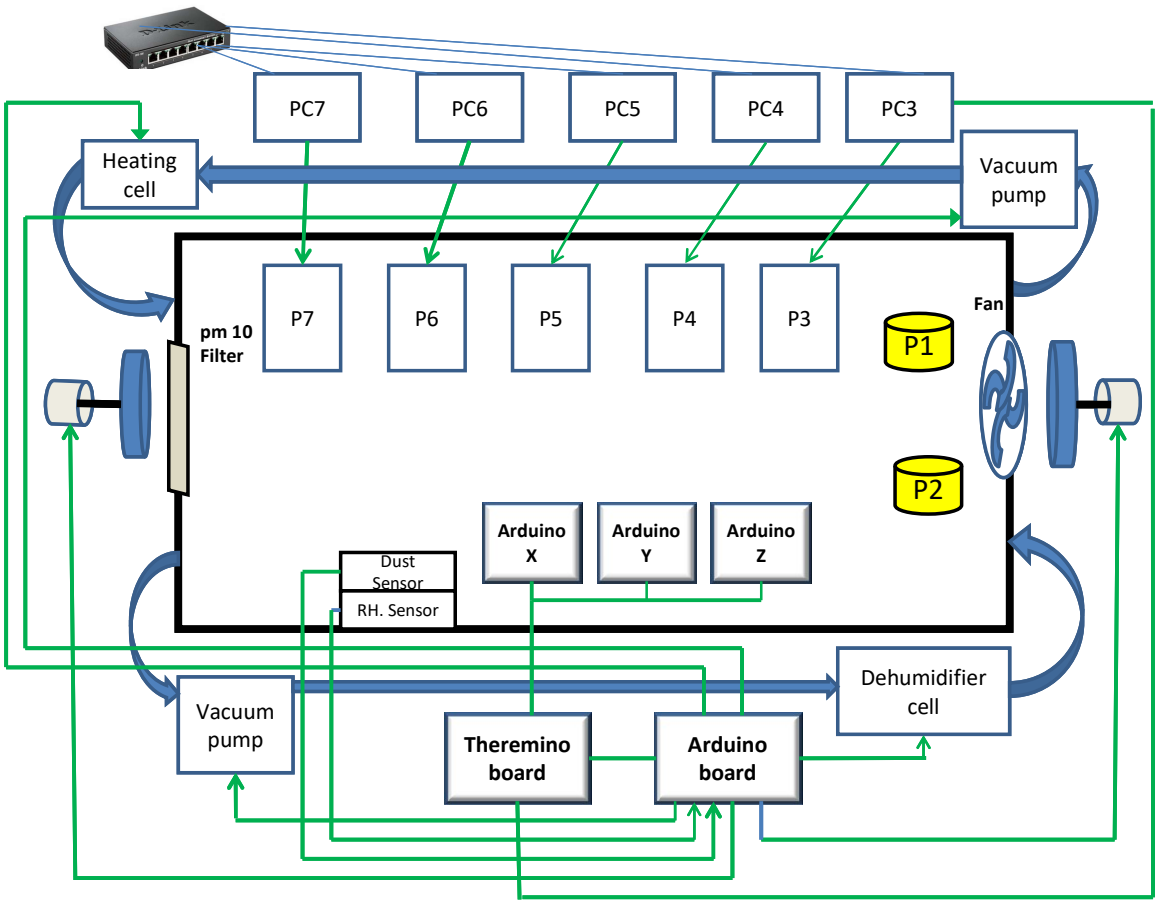


Fig.2.8 - Schematic representation of the Automatic Testing Machine.

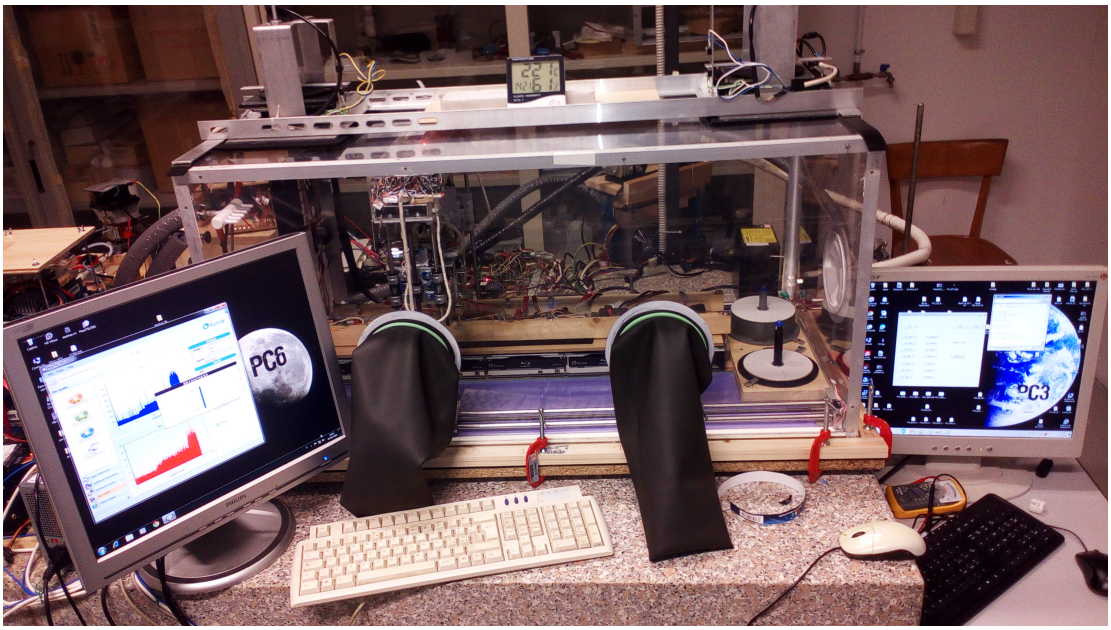


Fig.2.9 - Picture of the Automatic Testing Machine.

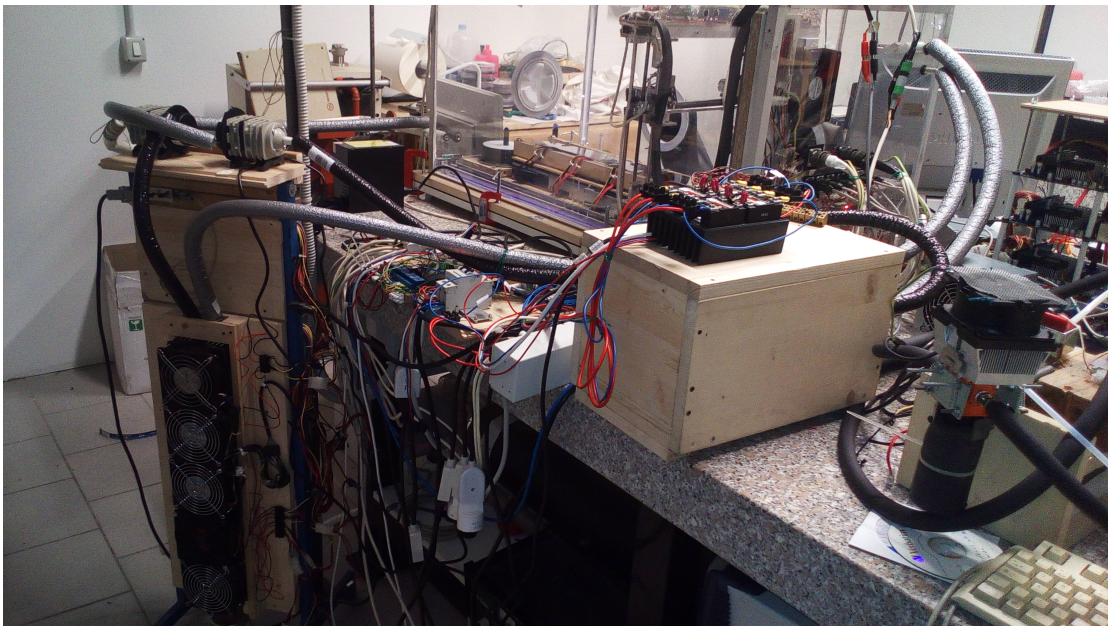


Fig.2.10 - Picture of the Automatic Testing Machine.

The ATM consists of a sealed chamber made of plexiglass (1 x 0.5 x 0.5 m) which is equipped with:

- 4 test devices P4-P7 (each of which is wired to a computer slave PC4-PC7)
- 1 burning device P3 (wired to the master computer PC3)
- 2 chargers for P1-P2 discs (each of which can contain up to 50 discs)
- 1 robotic system (which automatically moves the discs inside the chamber)
- 1 cleaning system (which controls the presence of dusts)

- 1 T/RH control system

All the automatic systems of the ATM are controller by PC3 (master). PC3 communicates to the "Theremino" board (master) via USB, communicates to PC4-PC7 via LAN, and is connected to the burning device. PC4-PC7 are connected to the tester devices via USB.

The Theremino board is linked to several other boards via bidirectional bus. In particular:

a) Arduino E – controls the environment of the chamber
b) Arduino X, Arduino Y, Arduino Z – each one controls the movements of the robot on a specific axis.

2.2.1 System control by Aduino_E

Arduino E runs a specific C++ software, which is aimed to control the presence of dusts, the humidity and the temperature. Moreover, there is a bidirectional bus which connects Arduino E to Theremino.

2.2.1.1 Clean Chamber System

The cleaning system that controls the air consists of two shutters. These shutters are electronically controlled by two step motors (*Fig.2.11*), that are placed in the upper part of the chamber.

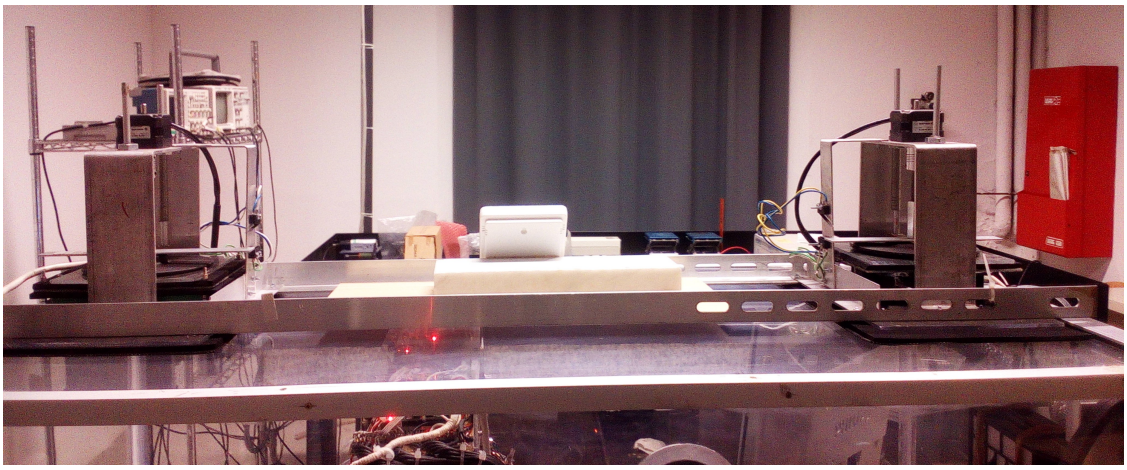


Fig. 2.11 – Picture of the clearing system.

While performing the automatic cleaning task, air flows through the input shutter and is filtered by a pm10 filter. The air then flows out of the chamber through a powerful fan, which eliminates the dusts. The amount of dusts inside the chamber is monitored by an internal sensor.

2.2.1.2 Temperature Control System

The system that controls the temperature is formed by a external heating cell, that introduces hot air in the chamber through a vacuum pump.

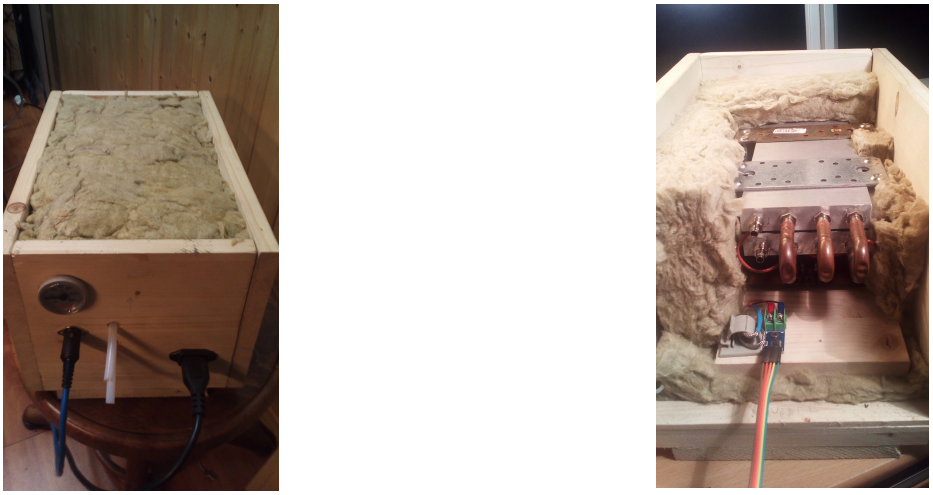


Fig. 2.12 – Pictures of the heating cell, closed (left) and open (right).

The heating cell (*Fig.2.12*) consists of an electrical resistor (800W), placed within two perforated aluminium plates. The air is heated inside this plates and then re-introduced into the chamber.
A k-type thermocouple records the temperature of the plates, and another electronic sensor records the temperature inside the chamber.

2.2.1.3 Humidity Control System

The humidity control system is composed of an external dehumidification cell (*Fig.2.13*). This cell "extracts" air from the chamber through a vacuum pump, dries it and then re-introduces it in the chamber.

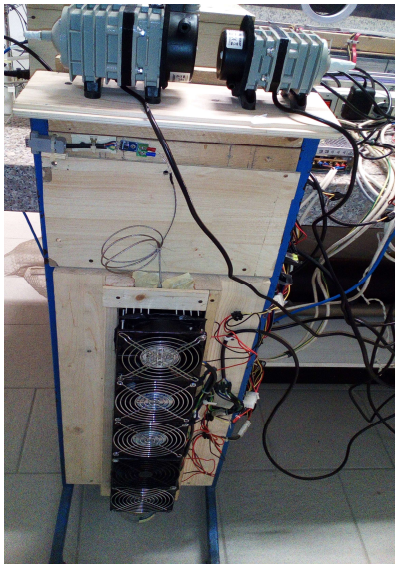


Fig. 2.13 - Picture of the dehumidification cell.

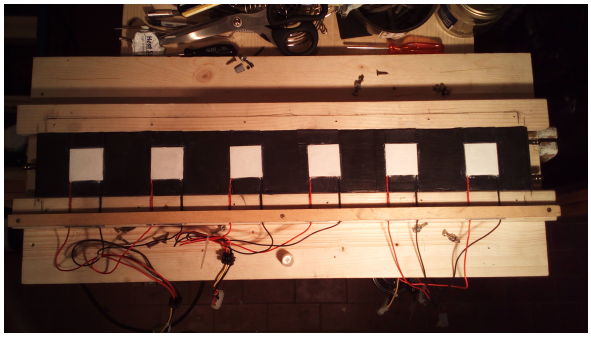


Fig. 2.14 - Picture of the Peltier cell.

The dehumidification cell is a tubular squared-sectioned cylinder (70 x 8 cm) which cooled by six Peltier cells (75W). The Peltier cells (*Fig.2.14*) are cooled by five powerful fans. A k-type thermocouple records the temperature of the Peltier cells, and another electronic sensor records the humidity inside the chamber.

2.2.2 Robot System

The robot system is shown in Fig.*2.15*.

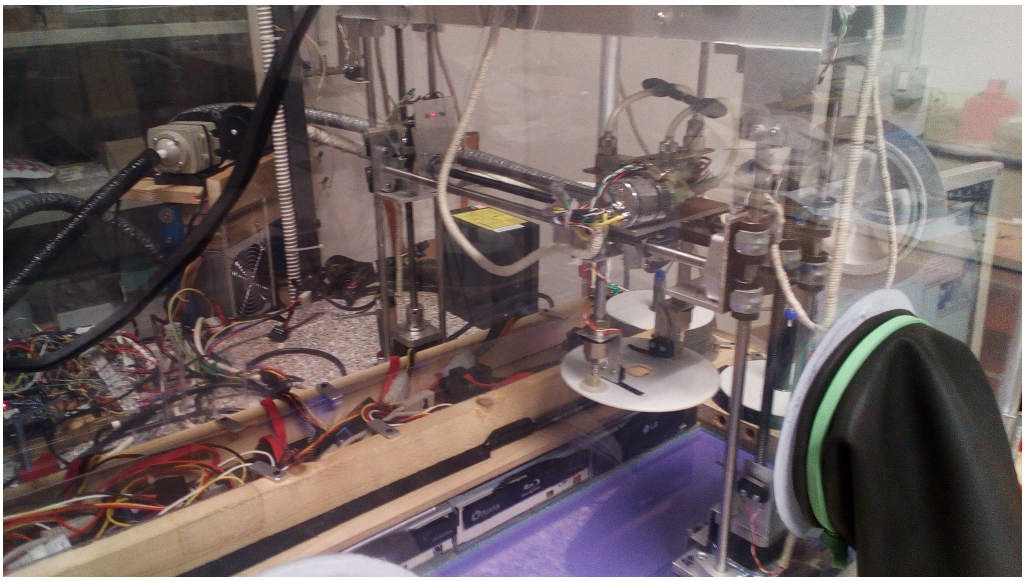


Fig. 2.15 – Picture of the Robot System.

Due to the complexity of the robotic system, we decided to perform a logical modelling of the apparatus, by means of a Finite State Automaton (FSA) [52].
It is based on the description of the evolution of the system, and therefore is able to describe the following state of a system starting from the initial state.

Definition. [Finite State Automaton] A Deterministic finite automaton (DFA) is a five-tuple:
$A = (Q, \Sigma, \delta, q_0, F)$, where:

- Q is a finite set of states
- Σ is a finite input alphabet
- q_0 is the initial / starting state, $q_0 \in Q$
- F is a set of final/accepting states, $F \subseteq Q$
- δ is a transition function, which is a map of $Q \times \Sigma \to Q$
- $\delta : Q \times \Sigma \to Q$ δ is defined for any $q \in Q$ and $s \in \Sigma$
- $\delta(q, s) = q'$ is equal to some state $q' \in Q$, could be $q' = q$

Definition. [from δ to strings] $\delta^* : (Q \times \Sigma^*)$, where Σ^* is the alphabet of available strings. The transition function in this case is $\delta(q, w)$ where w is a word.

Definition. [Language] Let $A = (Q \times \Sigma, \delta, q_0, F)$ be a DFA and let w be in Σ^*.
Then w is accepted by A iff $\delta(q_0, w) = p$ for some state p in Q.
Let $A = (Q \times \Sigma, \delta, q_0, F)$ be a DFA. Then the language accept by A is the set:
$L(A) = (w \; s.t. \; w \in \Sigma^* \wedge \delta(q_0, w) \in F\}$

An automaton evolves as follows: when the automaton is in a q_j state, then the w_k event changes the state of the automaton, according to the function δ. It is clear that, in order to trigger a change, the w_k event has to be allowed by q_j state – meaning that the (q_j, w_k) couple needs to belong to the δ dominium.
An example of a simulation of the system that uses the Jflap platform [53] is shown in *Fig.2.16*.
The robot is initially in the *Start* state, and upon receiving the vector sequences, moves to appropriate state. The active state is indicated in brown colour.
In this example the robot moves two discs from P1 position to P3 and P4 positions.

a) 000000 → robot is in the initial position P0 (where X=0,Y=0, Z=0) – the disc tray is close – No disc;
b) 000001 → robot moves to P1 position - the disc tray is closed – sucker head takes the disc;
c) 000011 → robot moves to P3 position –the disc tray is opened – sucker head releases the disc;
d) 000100 → robot moves to P1 position - the disc tray is closed - sucker head takes the disc
e) 000101 → robot moves to P4 position - the disc tray is opened - sucker head releases the disc;
f) 001000 → Moves sucker head to initial position – the disc tray is closed – No disc;

The commands are binary sequences (*w*) which are sent from Theremino via bus to Arduino boards (see *Fig.2.17*).

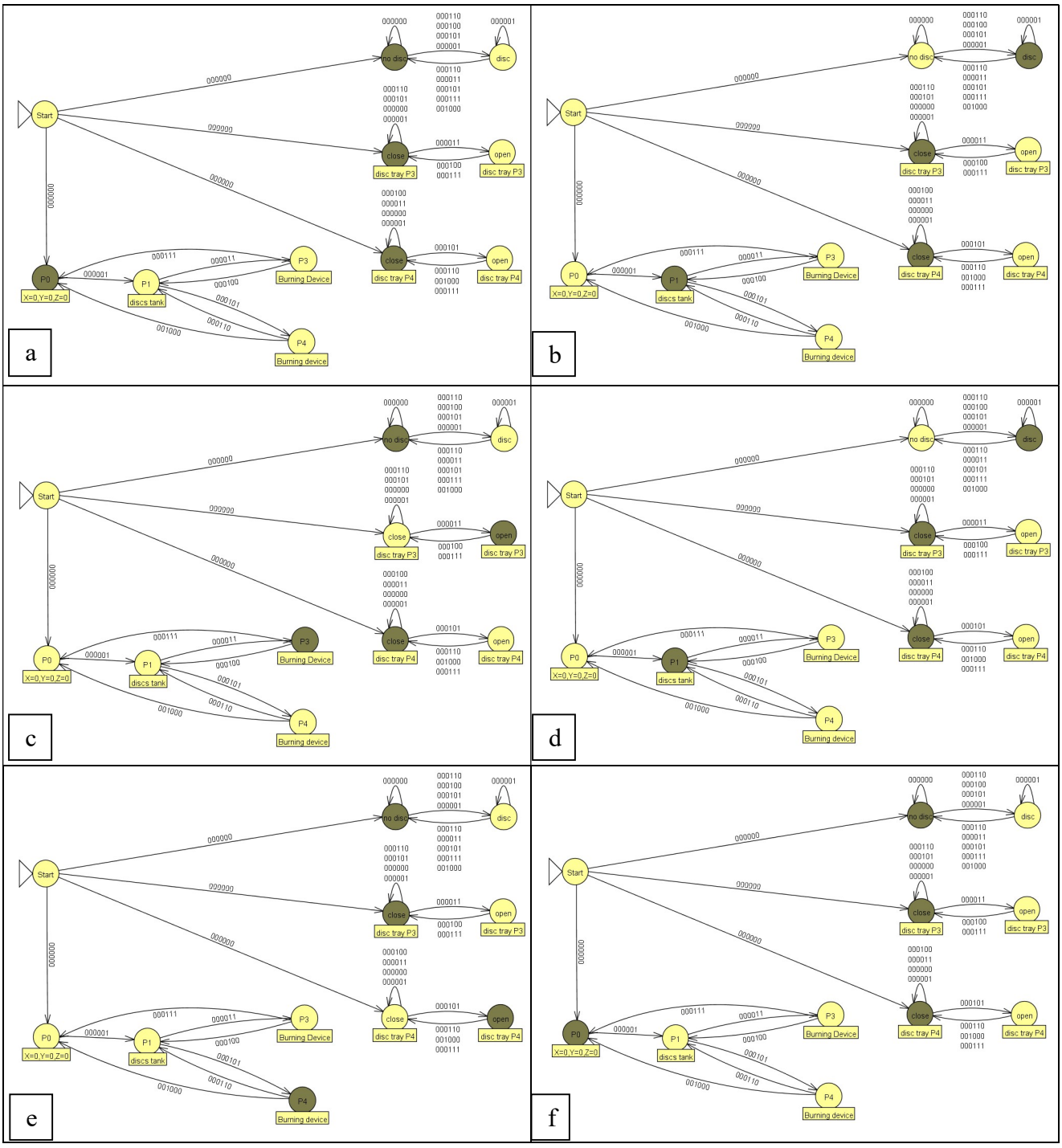


Fig. 2.16 - An example of a simulation of the robot system.

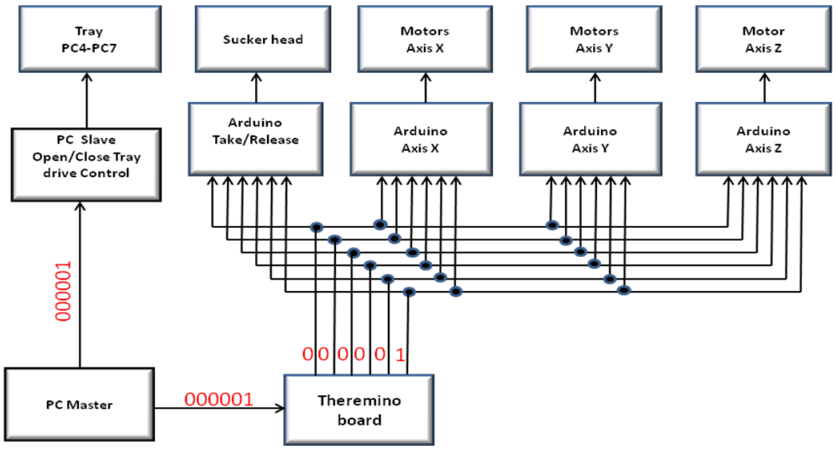


Fig. 2.17 – Theremino – Arduino Bus.

In *Fig.2*.18 the bus that connects the Theremino board and the X,Y,Z Arduino boards is shown.

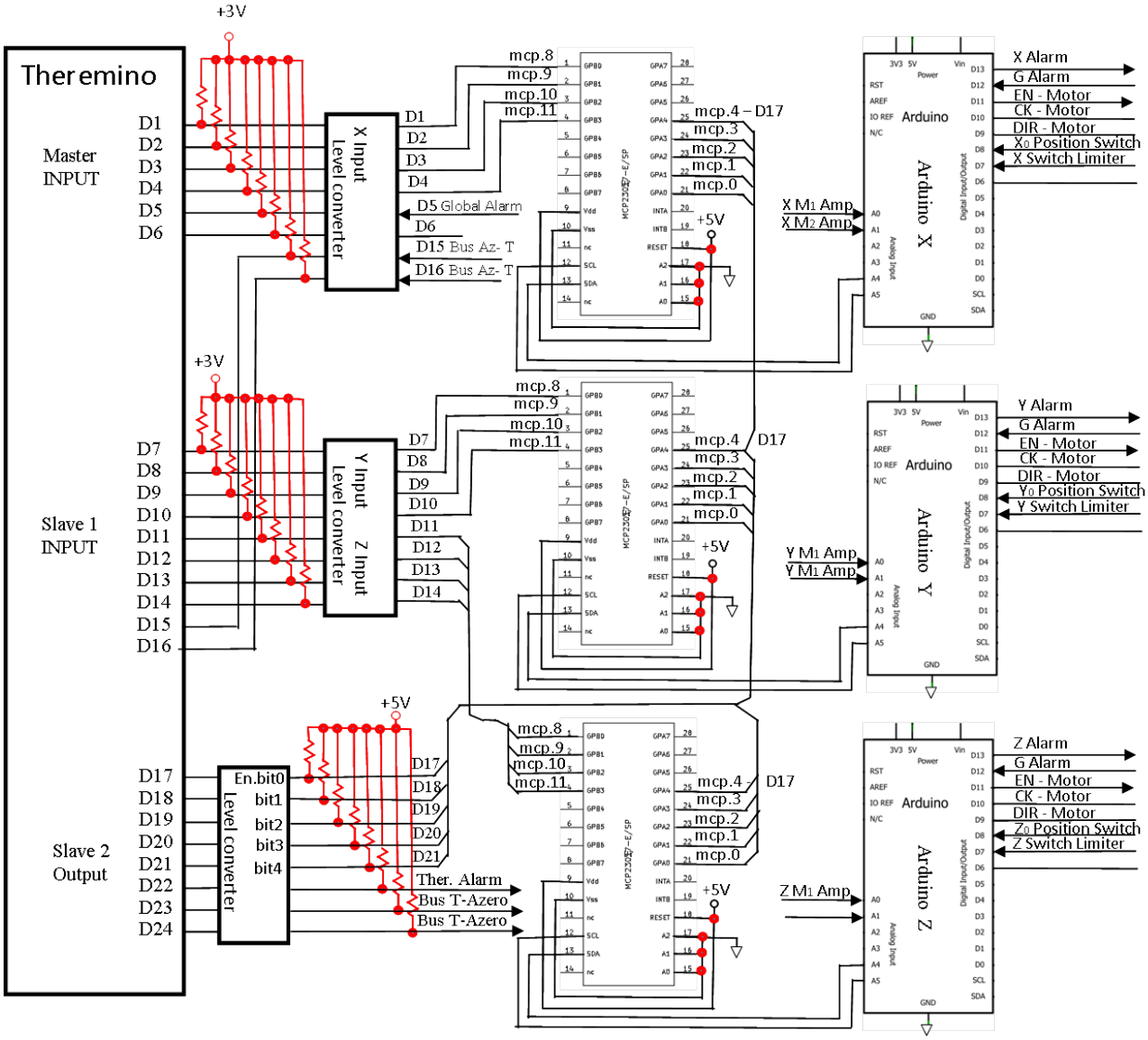


Fig. 2.18 – bus that connects the Theremino board and the X,Y,Z Arduino boards.

In *Fig.2.19* the block diagram of the robot system is shown.

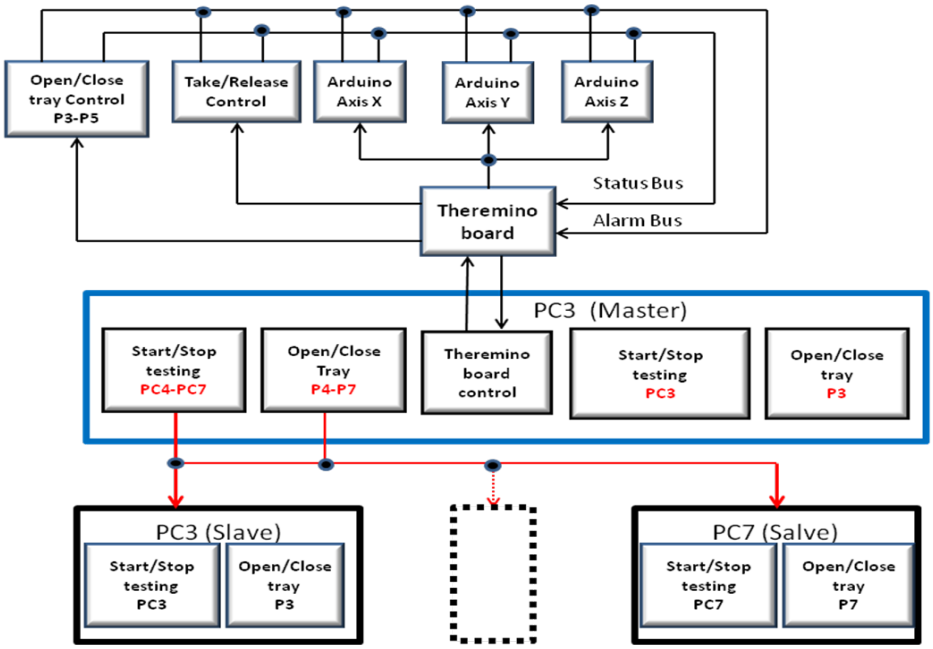


Fig. 2.19 - Block diagram of the robot system.

A script file on the PC-master allows to program all the operations for the automatic tests.

In *Fig. 2.19* the summary table of the commands is reported.

It is possible to control the execution of each single task. (i.e. the Robot shifting after the input of the command "move"), or to control several tasks through macro-instructions (i.e. MTEST, multi tests) which allow to automatically play a full cycle of tests on *n* disks.

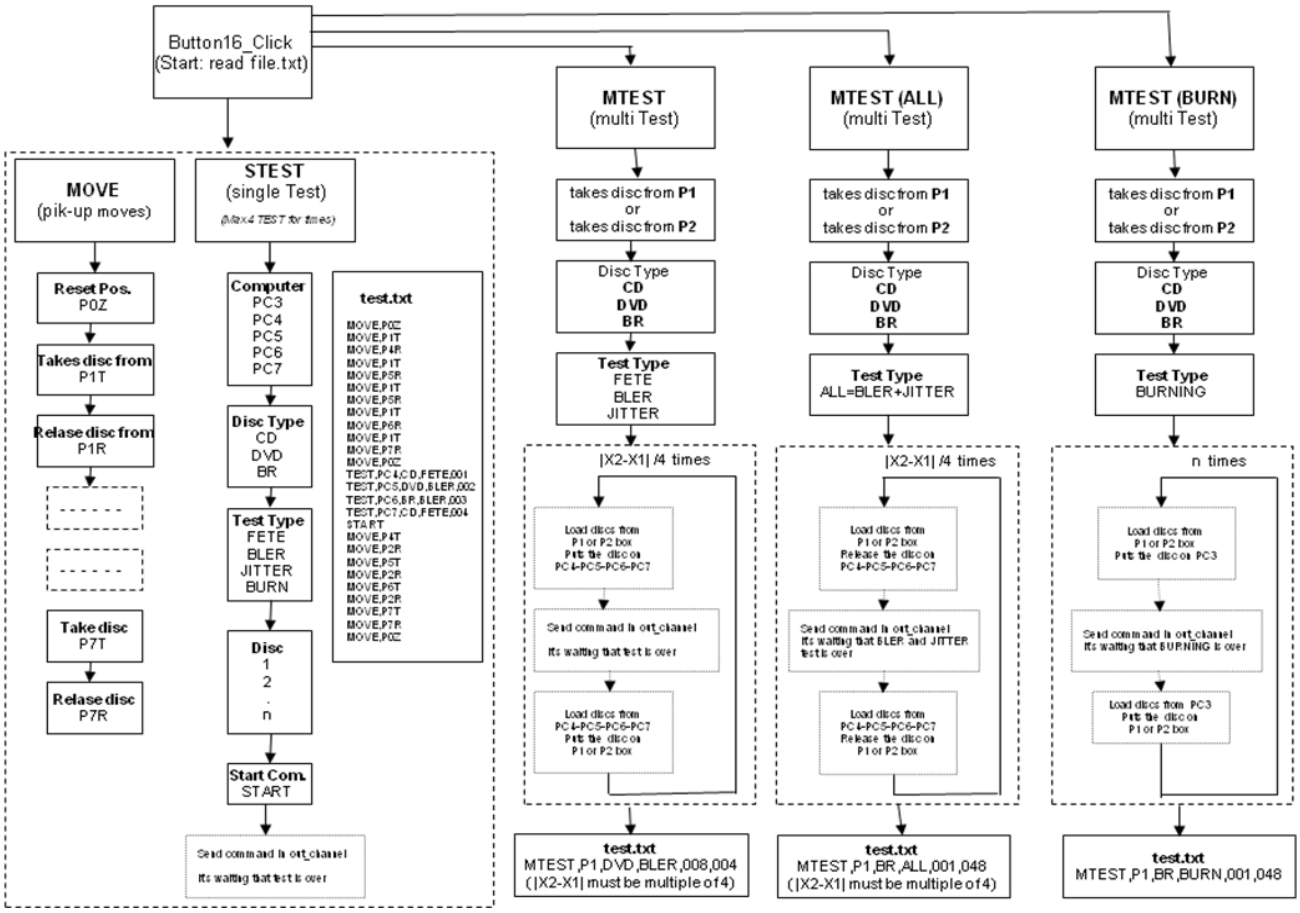


Fig. 2.20 - Summary table of the robot commands.

CHAPTER 3

3.Algebraic Preliminaries, Standard RS Code, and Adaptive RS Code

After introducing some basics concepts of Algebra and Galois theory [18], we will describe the Reed Solomon code (RS), that is widely used in applications such as digital data storage and digital transmission. Afterwards, the core of this thesis, which is an adaptive version of the Reed Solomon code (A-RS) will be presented.

3.1 Algebraic Preliminaries

In this section some basic algebraic concepts will be introduced in order to clarify the role that Galois theory occupies in the RS and A-RS codes. Although the main concepts are defined, some technical details have been skipped on purpose. A detailed introduction to field theory, and Galois theory can be found in [54].

3.1.1 Groups, Rings, and Fields

Definition [Group]: Let G be a non-empty set, and the operator $\cdot : G \times G \rightarrow G$.

A pair $(G, \cdot)$ is a group if

1. G is closed under $\cdot$, i.e. for every $a,b \in G$, $a \cdot b \in G$;
2. it exists $e \in G$ such that $e \cdot a = a \cdot e = a$, for every $a \in G$;
3. for every $a \in G$ it exists $a^{-1} \in G$, such that $a \cdot a^{-1} = 1$.
4. for every $a,b,c \in G$ $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

Remark: A group $(G, \cdot)$ is said to be Abelian if $\cdot : G \times G \rightarrow G$ is commutative, i.e. $a \cdot b = b \cdot a$ for every $a, b \in G$.

Example: the set of integer numbers Z is a group with respect to the sum, however it is not a group with respect to the multiplication. Let $z \in Z$ in fact this would require the existence of an element $z^{-1} \in Z$ such that $z \cdot z^{-1} = 1$, this element is $\frac{1}{z} \notin Z$.

Definition [Ring]: A ring is a set R equipped with two binary operations $+, \cdot : R \times R \rightarrow R$, such that:

1. $(R, +)$ is an abelian group;
2. is associative and there exists an element 1_R in R such that $1_R \cdot a = a \cdot 1_R = a$ for every $a \in R$;
3. the distributive law holds, i.e. for every $a,b,c \in R$, we have: $(a + b) \cdot c = a \cdot c + b \cdot c$.

Definition [Field]: Let F be a set, and $+, \cdot : F \times F \rightarrow F$ two operations defined on F. $(F, +, \cdot)$ is a field if the following conditions hold:

1. $(F, +)$ is an abelian group;
2. $(F_0, \cdot)$ is an abelian group, where $F_0 = F \setminus \{0\}$;
3. for every $a,b,c \in F$, $(a + b) \cdot c = a \cdot c + b \cdot c$.

Example [Field]: The set of rational numbers Q is a field when equipped with the usual operations of sum and multiplication.

Example [Polynomial ring]: Let F be a field, the polynomial ring $F[X]$ over F is the ring whose elements are of the form

$$a_0 X^0 + a_1 X^1 + \cdots + a_{m-1} X^{m-1} + a_m X^m,$$

where $X^0 = 1$, $X^k X^l = X^{k+l}$, and $\{a_0, \ldots, a_m\} \subseteq F$.

3.1.2 Galois Fields

A second key-concept for the introduction of the RS code is the one of Galois field.

Definition [Galois field]: A field whose underlying set F is finite (it has finite cardinality) is a Galois field (or finite field) denoted by $GF(p^n)$, where p is the (prime) number of elements of F and n is a positive integer.

We motivate the above definition by the following result: The order of a Galois field is always a prime number, or a power of a prime number [55]. Moreover, a unique Galois field is associated to each prime number.

Remark: Uniqueness in the category of fields has to be intended up to isomorphisms. Let $(F_1, +, \cdot)$, $(F_2, \ast, \Delta)$ be two finite fields. If there exists a homomorphism $\varphi: F_1 \rightarrow F_2$, such that φ is bijective, φ is said to be a field isomorphism.

Example: The simplest representation of finite field is given by set of integers modulo a certain prime number. Consider the set R of residue classes of a prime number p. For example, for $p = 2$, $R = \{0,1\}$. The set R equipped with the sum and multiplication modulo 2 is a field denoted by $Z/2Z$. In particular $Z/2Z$ is isomorphic to $GF(2)$.

3.1.3 Field Extension and Primitive Polynomial

Galois theory and some powerful results in abstract algebra assure the effectiveness of the RS code. First, we present the concept of field extension, that allows to enlarge a field, in order to solve otherwise impossible expressions. Second, we will show how the polynomial ring introduced in Sec. 3.1.1 plays a fundamental role in the representation of the element of a finite field.

Definition [Field extension]: Let L be a field. A subset $K \subsetneq L$ is a subfield of L if it is e field and it inherits its operations from the larger field L. In particular, L is said to be an extension of K, denoted by L / K.

Example: As a trivial example, one can observe that the field C of complex number is an extension of the real number R. As a second, more interesting example, consider the field of rational numbers Q equipped with usual operations. In this setting, the polynomial $p(x) = x^2-2$ has no root, since both $\pm\sqrt{2} \notin Q$. It is possible to extend the Q in order to include the roots of $p(x)$ and keep the field as small as possible by considering the field extension $Q(\sqrt{2}) = \{a + b\sqrt{2}, \textit{such that } a, b \in Q\}$. We denote this extension by $Q(\sqrt{2}) / Q$.

Definition [Primitive element]: Let L / K be a field extension. An element $\alpha \in L$ is primitive if $L = K(\alpha)$, and L / K is said to be a simple extension.

Example: Trivially, $\sqrt{2}$ is a primitive element for the extension $Q(\sqrt{2}) / Q$. It is also possible to prove that $Q(\sqrt{2}, \sqrt{3}) / Q$ is simple. It suffices to set $\gamma = \sqrt{2} + \sqrt{3}$ and show that it is possible to express both $\sqrt{2}$ *and* $\sqrt{3}$ as a linear combination of power of γ, with coefficient in Q. For instance we have

$$\sqrt{2} = \frac{\gamma^3 - 9\gamma}{2}.$$

Given a finite field extension L / K, if the primitive element exists, any element of L can be written as a polynomial of the form

$$k_0 + k_1 \alpha^1 + \cdots + k_{m-1} \alpha^{m-1} + k_m \alpha^m,$$

where $k_i \in K$ for every i, $\alpha \in L$ is fixed.

Considering a Galois field $E = GF(p^n)$, a field of order n over $GF(p)$. E has $q = p^n$ elements, its nonzero elements are roots of the polynomial $X^{(q-1)} - 1$, and all elements of E are roots of $X^q - X$. From this simple observation, we can conclude that all the finite fields of q elements are isomorphic, in the usual sense for fields.

Definition [Minimal polynomial]: Let L / K be a field extension, $K[x]$ the ring of polynomial in x over K, and α an element of L. The minimal polynomial of α is the monic polynomial $f(x) \neq 0_{K[x]}$ of least degree in $K[x]$, such that $f(\alpha) = 0$

It is important to observe that a minimal polynomial is irreducible. To proof this fact it is sufficient to assume by contradiction that $f = gh$, where $g,h \in K[x]$, and observe that this assumptions contradicts the hypothesis of minimality on the degree of f.

Definition [Primitive polynomial]: Let $K(\alpha) / K$ be a simple field extension. The primitive polynomial is the minimal polynomial associated to the element α.

It is possible to reformulate this definition by using the concept of irreducibility. Consider the field $GF(p)$ for p prime. An irreducible polynomial $f(x) \in GF(p)[x]$ of degree m is primitive if the smallest n for which $f(x)$ divides $x^n - 1$ is $n = p^m - 1$.
Example: The polynomial $x^2 + x + 1$ is primitive in $GF(2)$. Thus the degree $m = 2$ and $p = 2$. According to the statement above, it has to be that $x^n - 1 = x^{4-1} - 1 = x^3 - 1$ is the smallest degree polynomial of the form $x^n - 1$ that divides $x^2 + x + 1$. This is easy to show by remembering that

$$x^3 - 1 = (x - 1)(x^2 + x + 1), \text{ equivalently}$$
$$x^3 + 1 = (x + 1)(x^2 + x + 1) \text{ in } GF(2)[x].$$

3.1.4 Polynomial Representation of Finite Fields

Primitive polynomials are used to give a particularly effective representation of the elements of a finite field. This is done by observing that the root α of a *m-th* degree primitive polynomial has degree $q = p^m - 1$. Thus, $\alpha^0, \ldots, \alpha^q = 1$ form a multiplicative group of size q. It follows that the elements of the group $GF(p^m)$ can be written as a linear combinations of powers of α of degree less than m.

Example [Construction of GF(4)]: We already proved that $x^2 + x + 1$ is primitive in the case of $p = 2$ and $m = 2$. Let α be a root of $x^2 + x + 1$. Since $x^2 + x + 1$ divides $x^3 + 1$ in $GF(2)[x]$, we have $\alpha^3 = 1$. Hence $\alpha^0 = 1$, $\alpha^2 = \alpha + 1$, $\alpha^3 = 1$, and $GF(4)$ can be represented as $\{0, 1, \alpha, \alpha + 1\}$.

Example [Field symbols for GF(16)]: With a primitive polynomial i.e. $F(X) = 1 + X + X^4$ of degree m=4, we can generate a Galois Field $GF(p^m) = GF(2^4) = GF(16)$. Since we want to generate the $GF(2^4)$, we need any fourth order $F(X)$.

Beside the elements {0,1} of $GF(2)$ subfield of $GF(2^m)$ in order to develop our field, set the *primitive element* $a(X)$, often denoted as α, equivalent to $0X^{m-1} + 0X^{m-2} + \ldots + 1X + 0 = X$.

We set $a(X) = \alpha = X$ to obtain the 4-tuple $i_3X^3 + i_2X^2 + i_1X^1 + i_0$ for each element α^i of GF(16).

Therefore $\alpha^{-\infty} = 0$ and $\alpha^o = 1$. To develop the field elements α^i we will use a modulo method [56, 57]. We can performed directly from $\alpha(X) \bmod F(X) = (i_3X^3 + i_2X^2 + i_1X^1 + i_0) \bmod F(X)$ where $F(X) = 1 + X + X^4$. Thus we have (*Table 3.1*):

$0 = 0 = 0 \bmod F(X) = 0$	$\alpha^7 = X^7 = X^7 \bmod F(X) = X^3 + X + 1$
$1 = 1 = 1 \bmod F(X) = 1$	$\alpha^8 = X^8 = X^8 \bmod F(X) = X^2 + 1$
$\alpha = X = X \bmod F(X) = X$	$\alpha^9 = X^9 = X^9 \bmod F(X) = X^3 + X$
$\alpha^2 = X^2 = X^2 \bmod F(X) = X^2$	$\alpha^{10} = X^{10} = X^{10} \bmod F(X) = X^2 + X + 1$
$\alpha^3 = X^3 = X^3 \bmod F(X) = X^3$	$\alpha^{11} = X^{11} = X^{11} \bmod F(X) = X^3 + X^2 + X$
$\alpha^4 = X^4 = X^4 \bmod F(X) = X + 1$	$\alpha^{12} = X^{12} = X^{12} \bmod F(X) = X^3 + X^2 + X + 1$
$\alpha^5 = X^5 = X^5 \bmod F(X) = X^2 + X$	$\alpha^{13} = X^{13} = X^{13} \bmod F(X) = X^3 + X^2 + 1$
$\alpha^6 = X^6 = X^6 \bmod F(X) = X^3 + X^2$	$\alpha^{14} = X^{14} = X^{14} \bmod F(X) = X^3 + 1$

Table 3.1 - Definition of α elements.

Summary of the field representations is reported in *Table 3.2.*

GF(16) elements	*Power represent.*	*Polynomial representation*	*Vector represent.*	*Decimal represent.*
0	0	0	0000	0
1	1	1	0001	1
α	X	X	0010	2
α^2	X^2	X^2	0100	4
α^3	X^3	X^3	1000	8
α^4	X^4	$X+1$	0011	3
α^5	X^5	X^2+X	0110	6
α^6	X^6	X^3+X^2	1100	12
α^7	X^7	X^3+X+1	1011	11
α^8	X^8	X^2+1	0101	5
α^9	X^9	X^3+X	1010	10
α^{10}	X^{10}	X^2+X+1	0111	7
α^{11}	X^{11}	X^3+X^2+X	1110	14
α^{12}	X^{12}	X^3+X^2+X+1	1111	15
α^{13}	X^{13}	X^3+X^2+1	1101	13
α^{14}	X^{14}	X^3+1	1001	9

Table 3.2 - Representations of the Equivalent Elements.

In *Table 3.2* there are some equivalent ways to represent a finite field symbol. Since we chose the special case of setting the primitive element $\alpha(X)$ equivalent to X to generate the field, we will represent the field elements α^j in terms of the α^j instead of the X^j. The power and vector representations are most popular. Multiplication by hand is easily performed using the power representation and addition using the vector representation and addition is performed using the vector representation.

3.2 Reed Solomon Codes

Reed Solomon codes [6] are non binary cyclic codes with code symbols from a Galois field, $GF(p^m)$ where p is prime, which in our case is equal to *2*, and m is an integer. A Reed-Solomon code is specified as RS(n,k). The encoder takes k data symbols of m bits each and adds parity symbols to make a code-word with at most n symbols where $n=2^m-1$. Therefore, in a code-word there are $n-k$ parity symbols of m bits each. A RS decoder can correct up to t symbols that contain errors in a code-word, where $t = (n-k)/2$.

3.2.1 Reed Solomon Encoding

Given a single value of the Galois field extension *m*, *GF(p^m)*, a set of RS codes with varying error correction capabilities, block lengths, and rate can be constructed. The p^m unique code symbols are constructed from the field primitive polynomial *F(X)* and the primitive element $\alpha(X)$. The parity symbols are obtained using the generator polynomial $g(X)$ with roots from *GF(p^m)*. A *RS(n,k) code* is defined given values for *m, n* and $g(X)$.

However, when we get into implementation we need to also know *p* (which for us is always *2*), *F(X)* (a primitive polynomial *p(X)*, $\alpha(X)$ (for us it is $X = \alpha$), and α^G (which is any primitive element of *GF(p^m)* using *F(X)* and is almost always set to α^1).

The generator polynomial is a generic polynomial that is used to generate the encoding polynomial; for an RS code takes the following form:

$$g(X) = g_0 + g_1 X + g_2 X^2 + \ldots + g_{2t-1} X^{2t-1} + X^{2t}$$

Since the generator polynomial is of degree 2t, there must precisely *2t* (where *t=(n-k)/2*) successive powers of α that are roots of the polynomial. We designate the roots of $g(X)$ as: $\alpha, \alpha^2, \ldots, \alpha^{2t}$

Consider as an example, the RS(15,9). We describe the generator polynomial in terms of its *2t=n-k=6* roots, as follows:

$$g(x) = \prod_{i=1}^{2t} (X - \alpha^i) = (X-\alpha)(X-\alpha^2)(X-\alpha^3)(X-\alpha^4)(X-\alpha^5)(X-\alpha^6)$$

$$g(X) = X^6 + \alpha^{10} X^5 + \alpha^{14} X^4 + \alpha^4 X^3 + \alpha^6 X^2 + \alpha^9 X + \alpha^6$$

To generate the encoding in systematic form we have to shift a message polynomial *m(X)* into the rightmost *k* stages of a codeword register and then appending a parity polynomial, *p(X)*, by placing it in the leftmost *n-k* stages. Therefore, we multiply *m(X)* by X^{n-k}, thereby the message polynomial is positioned on the left part of the code-word, before of *n-k*. Next, we divide X^{n-k} by the generator polynomial $g(X)$, which is written as, where $q(X)$ is quotient and $p(X)$ is remainder polynomial:

$$X^{n-k} m(X) = q(X) g(X) + p(X)$$

The remainder polynomial is the parity.

To generate a parity symbols p(X) we have to divide the message $X^{n-k}m(X)$ *by modulo-g(X)*
At last the resulting codeword polynomial *V(X)* is:

$$V(X) = p(X) + X^{n-k}m(X)$$

3.2.2 Reed Solomon Decoding

3.2.2.1 Syndrome Calculation

The Syndrome [58] is the first step in the Reed Solomon decoding process. This is done to detect if there are any errors in the received sequence. Since the *code-word* is generated by multiplying the message by the generator polynomial, if the receiver code-word is error free then its modulus with respect to the generator polynomial should evaluate zero.
The Syndrome can then be defined as *S=R(X) mod g(X)*. The Syndrome is the remainder obtained by dividing the received sequence by *g(X)* and *R(X) = V(X)+E(X)*.
Since *V(X)* is a multiple of *g(X)*, S actually evaluates to *S=E(X) mod g(X)*.

Let: V(X) be the *code-word* without any error, *R(X)* be the received code-word and *E(X)* be any error that the transmission channel introduce. Rewriting *V(X), E(X)*, and *R(X)*, in polynomial form we have the following, where $v_i, e_i, r_i, \quad i = 0,1,2,...,n-1$ are elements of GF(2^n):

$$V(X) = v_0 + v_1 x + v_2 x^2 + \ldots + v_{n-1} x^{n-1} \qquad (1)$$

$$E(X) = e_0 + e_1 x + e_2 x^2 + \ldots + e_{n-1} x^{n-1} \qquad (2)$$

$$R(X) = r_0 + r_1 x + r_2 x^2 + \ldots + r_{n-1} x^{n-1} \qquad (3)$$

R(X) is related to the error polynomial E(X) and the code-word polynomial V(X) as follows:

$$R(X) = V(X) + E(X)$$

where the error pattern E(X) added by channel is expressed as:

$$E(X) = R(X) - C(X)$$

3.2.2.2 Error Location

E(X) can be described by a list of values and locations of its non-zero components. The error location [59] will be given in terms of an error-location number, which is α^j for the $(n-j)^{\text{th}}$ symbol.

We called p the *couple* (x_j,e_j) where x_j indicates the error location number and e_j indicates the error value. Assume that $E(X)$ is an error pattern of p errors at locations $j_1,j_2,....,j_p$. Then:

$$E(X)_t = e_{j_1}x^{j_1} + e_{j_2}x^{j_2} + ... + e_{j_p}x^{j_p} \quad \text{where} \quad p \le t \quad \text{and} \quad 0 \le j_1 < j_2 < .... < j_p . \qquad (4)$$

If $V(X)$ codeword does not contain errors, the syndrome is equal zero. A nonzero syndrome indicates that an error has been detected. For RS codes the Syndrome is defined as vector S with $2t$ components as follows:

$$S_1 = r(\alpha^i) = r_0 + r_1\alpha^i + r_2\alpha^i ... + r_{n-1}(\alpha^i)^{n-1} \quad \text{for} \quad \text{i=1,2,3, ,2t} \qquad (5)$$

$$\text{Since} \quad R(X)_t = V(X)_t + E(X)_t, \quad \text{we have} \quad S_i = V(\alpha^i) + E(\alpha^i) \qquad (6)$$

$$\text{Since} \quad V(\alpha^i) = 0, \quad \text{we have} \quad S_i = E(\alpha^i) = \sum e_{j_l} x_{j_j}^i \qquad (7)$$

From equations (4) and (7) we have:

$$S_i = r(\alpha^{2t}) = e_{j_1}(\alpha^{j_1})^i + e_{j_2}(\alpha^{j_2})^i + ... + e_{j_p}(\alpha^{j_p})^i \quad \text{for } i=1,2,,...,2t \qquad (8)$$

By expanding equation (8):

$$\begin{aligned} S_1 &= r(\alpha) = e_{j_1}\alpha^{j_1} + e_{j_2}\alpha^{j_2} + ... + e_{j_p}\alpha^{j_p} \\ S_2 &= r(\alpha^2) = e_{j_1}(\alpha^{j_1})^2 + e_{j_2}(\alpha^{j_2})^2 + ... + e_{j_p}(\alpha^{j_p})^p \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ S_{2t} &= r(\alpha^{2t}) = e_{j_1}(\alpha^{j_1})^{2t} + e_{j_2}(\alpha_2^{j_2})^{2t} + ... + e_{j_p}(\alpha_p^{j_p})^{2t} \end{aligned} \qquad (9)$$

Equation (9) is non linear equation.

We define an error locator number as $\beta_p = \alpha^{j_l}$. We obtain the $2t$ syndrome symbols by substituting α^i into the received polynomial (9) for $i=1,2,...2t$:

$$
\begin{aligned}
S_1 &= r(\alpha) = e_{j_1}\beta_1 + e_{j_2}\beta_2 + \ldots + e_{j_p}\beta_p \\
S_2 &= r(\alpha^2) = e_{j_1}\beta_1^2 + e_{j_2}\beta_2^2 + \ldots + e_{j_p}\beta_p^2 \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
S_{2t} &= r(\alpha^{2t}) = e_{j_1}\beta_1^{2t} + e_{j_2}\beta_2^{2t} + \ldots + e_{j_p}\beta_p^{2t}
\end{aligned}
\tag{10}
$$

We have *2t* unknown *t* error values and *t* locations, and *2t* simultaneous non linear equations. To solve this equations we used a Reed Solomon decoding [60].

Once a nonzero syndrome vector has been computed, that signifies that an error has been received. We have need to learn the location of the error or errors.

An error locator polynomial [61], $\sigma(X)$, can be defined as follows:

$$
\sigma(X) = (1+\beta_1 x)(1+\beta_2 x)....(1+\beta_p x) = 1+\sigma_1 x+\sigma_2 x^2 + ... + \sigma_p x^p \tag{11}
$$

The roots of $\sigma(x)$ are $\frac{1}{\beta_1},\frac{1}{\beta_2},....,\frac{1}{\beta_p}$. The reciprocal of the roots $\sigma(x)$ are the error location numbers of the error pattern $E(X)$. Then, using autoregressive modeling techniques, we form a matrix from the syndromes, where the first *t syndromes* are used to predict the next Syndrome:

$$
\begin{bmatrix}
S_1 & S_2 & S_3 & \ldots & S_{t-1} & S_t \\
S_2 & S_3 & S_4 & \ldots & S_t & S_{t+1} \\
\vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\
S_{t-1} & S_t & S_{t+1} & \ldots & S_{2t-3} & S_{2t-2} \\
S_t & S_{t+1} & S_{t+2} & \ldots & S_{2t-2} & S_{2t-1}
\end{bmatrix}
\bullet
\begin{bmatrix}
\sigma_t \\ \sigma_{t-1} \\ \vdots \\ \sigma_2 \\ \sigma_1
\end{bmatrix}
=
\begin{bmatrix}
-S_{t+1} \\ -S_{t+2} \\ \vdots \\ -S_{2t-1} \\ -S_{2t}
\end{bmatrix}
\tag{12}
$$

We apply the autoregressive model of previous equation by the largest dimensioned matrix that has a nonzero determinant.

To solve for coefficients $\sigma_1,\sigma_2,\cdots,\sigma_t$ and of the error locator polynomial $\sigma(X)$, we first take the inverse of the matrix in previous equation.

The inverse of a matrix is equal:

$$
Inv[A] = \frac{cofactor[A]}{\det[A]} \tag{13}
$$

When we have found:

$$\sigma(X) = 1 + \sigma_1 x + \sigma_2 x^2 + ... + \sigma_p x^p \qquad (14)$$

The error location number are the reciprocals of the roots of $\sigma(X)$.

The roots of $\sigma(X)$ can be found simply by substituting $\alpha^0, \alpha^1, \alpha^2, ...\alpha^{n-1}$ into $\sigma(X)$.

$\sigma(X) = 0$ indicates that one root exists at $1/\beta_l = \alpha^i$ and then $\beta_l = 1/\alpha^i = \alpha^{-i}$ where the β index "*l*" refer to the first, second,…p^{th} error.

We have *p* symbol errors, so that the error polynomial is the following form:

$$E(X) = e_{j_1} X^{j_1} + e_{j_2} X^{j_2} + ... + e_{j_p} X^{j_p} \qquad (15)$$

The location errors will be indicated with $\beta_p = \alpha^{j_p}$ i.e $\beta_1 = \alpha^{j_i} = \alpha^2$

3.2.2.3 Error Values

An error had been denoted e_{j_1} where index j refers to the location and the index *p* identifies the j[th] error. We will determinate the error value e_{j_p} associate with locations β_p [61]

We write the equation (10) in matrix form:

$$\begin{bmatrix} \beta_1 & \beta_2 & \dots & \beta_p \\ \beta_1^2 & \beta_2^2 & \dots & \beta_p^2 \\ \vdots & \vdots & \dots & \vdots \\ \beta_1^p & \beta_2^p & \dots & \beta_p^p \end{bmatrix} \bullet \begin{bmatrix} e_{j_1} \\ e_{j_2} \\ \vdots \\ e_{j_p} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_p \end{bmatrix} \qquad (16)$$

To solve for error values $e_1, e_2, ..e_p$ we have to invert the equation (16)

$$Inv[A] = \frac{cofactor[A]}{\det[A]}$$

Correcting the Received polynomial

We have the equation

$$\hat{E}(X) = e_{j_1} X^{j_1} + e_{j_2} X^{j_2} + ... + e_{j_p} X^{j_p} \qquad (17)$$

$$\hat{U}(X) = R(X) + \hat{E}(X) = U(X) + E(X) + \hat{E}(X) \qquad (18)$$

3.3 Adaptive Reed Solomon Code A-RS(n,k)

In an adaptive RS code [12] it is possible to increase or decrease the correction capability by increasing or decreasing the redundancy (that is the number of parity symbols in each code-word). By considering as example the standard RS(15,9) code (*Fig 3.1*), it has:

n = 15	number of symbols of each code-word
k = 9	number of symbols of message in each code-word
n-k = 2t = p = 6	number of parity symbols in each code-word (redundancy)
(n-k)/2 = t = 3	maximum number of symbols that can be corrected.

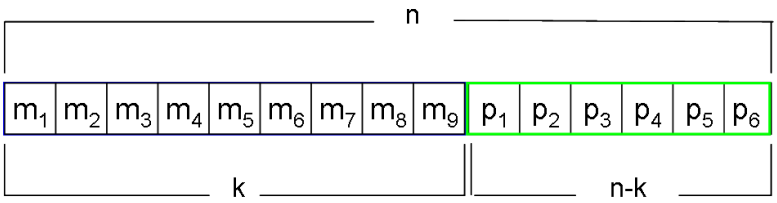


Fig. 3.1 - Standard RS(15,9) code.

To increase the correction capability *t* by one, the number of parity symbols *p=2t* must increase by two. At this point, in order to transform the standard code into an Adaptive Reed Solomon code (A-RS code) there are two different strategies, to change the correction capability changed:

1. The length of the message can be kept constant, and the number of parity bytes is increased or decreased. In this case the code-word length must change [62]. For example, to obtain an increase the correction capability by 1 symbol (from t = 3 to t = 4) the length of one code-word should rise from 15 to 17, as shown in *Fig. 3.2*.

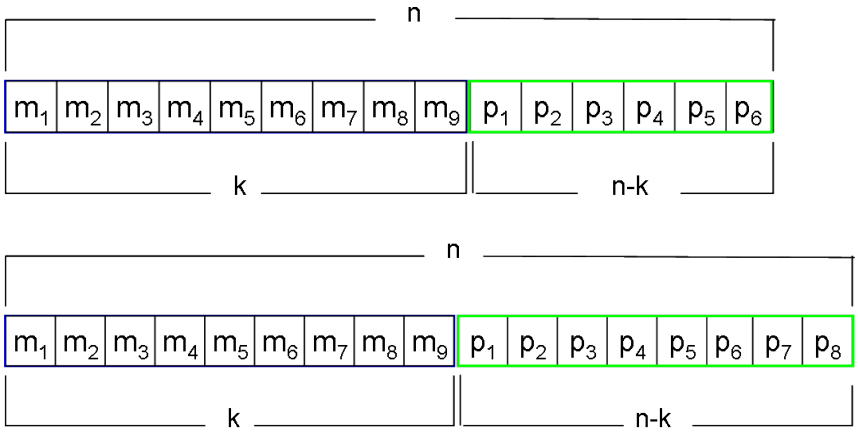


Fig.3.2 - The code-word displayed in the lower figure has a correction capability higher than that of the upper code-word, but is longer.

2. The length of the code-word is kept constant and the number of parity bytes is increased or decreased. In this case, the length of the message decreases or increases, respectively. For example, increase the correction capability by 1 symbol (from t = 3 to t = 4) the length of the message should decrease from 9 to 7, as shown in *Fig. 3.3*.

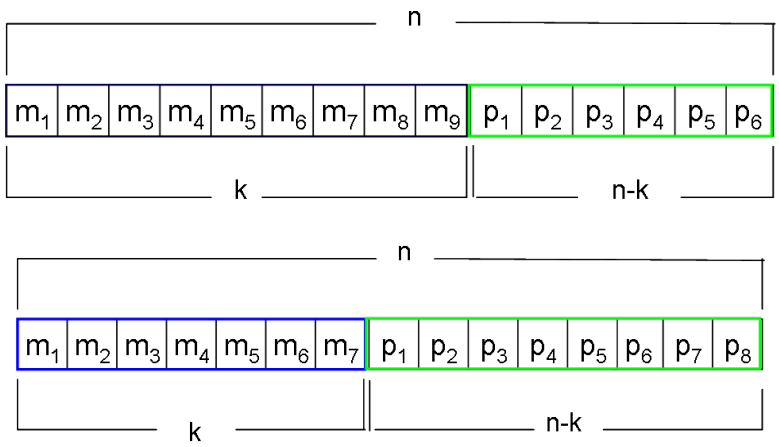


Fig 3.3 - The code-word displayed in the lower figure has a correction capability higher than that of the upper code-word, but its message part is shorter.

In this work, I used the second approach. That is to say, the length of the code-words is kept constant while changing the number of parity symbols. The proposed coding scheme may be interpreted as a form of spatial *unequal error protection* [63], applied statically, as the code rate profile obtained from the degradation function is not modified throughout the disc life.
This choice is made to allow the implementation of the A-RS code into burning devices without the need of significant hardware modifications. In fact, most codification systems are subjected to some constraints in terms of length of the code-words, according to ISO standard regulations.[CIT] for CD, DVD and BD.

As a simple example, we present in table 3.3 the A-RS(n,k) with n=15. The number of the messages symbols *k* can be *3,5,7,9,11 or 13* and the number of parity symbols is then one of *n-k = 12,10,8,6,4 or 2*. We thus have six different levels of A-RS coding reliability *p=2t = 1,2,3,4,5 or 6* with six different couples of *(k,n-k) = (3,12),(5,10),(7,8),(9,6),(11,4) or (13,2)* see *Table 3.3*.

Number of message symbols	Number of parity symbols		Number of correctable erroneous symbols
	n-k	2t=parity	
k=3	15-3	12	6
k=5	15-5	10	5
k=7	15-7	8	4
k=9	15-9	6	3
k=11	15-11	4	2
k=13	15-13	2	1

Table 3.3 – A-RS code (15,k) – The standard code with k=9 is evidenced in red.

To generate the Galois field elements for a A-RS(15,k) with $k = 3,5,..,13$ a unique Galois Field is sufficient.

The Galois Field is defined by $GF(p^m)$, where p is prime and m is integer.

For our RS(n,k) with $n=15$ we need a $GF(2^4)$ since $n=2^4-1$. Furthermore, "m" will be the number of bits for each A-RS symbol.

The number of elements in Galois Field must be $q= p^m =2^4$ (see Table 3.3).

Galois Field elements are generated from the primitive polynomial $p(x)$ of degree $m=4$

$p(x)=x^4+x+1$.

Let α be a root of p(x). This implies that $\alpha^4 + \alpha + 1 = 0$, or equivalently, $\alpha^4 = \alpha + 1$.

The field elements representation is in *Table 3.2*.

3.3.1 Polynomial Generator

A-RS(15,k)$_t$ code has six polynomial generator with $t= 1,2,3,4,5,6$:

$$g(x)_{n-2t} = \prod_{i=1}^{2t} (X - \alpha^i)$$

Therefore:

$$g(x)_{k=13} = (X-\alpha)(X-\alpha^2) \qquad t=1$$

$$g(x)_{k=11} = (X-\alpha)(X-\alpha^2)(X-\alpha^3)(X-\alpha^4) \qquad t=2$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$g(x)_{k=3} = (X-\alpha)(X-\alpha^2)(X-\alpha^3).....(X-\alpha^{12}) \qquad t=6$$

A-RS(15,k)$_t$ code has six different symbols in a codeword for each different value of $t= 1,2,3,4,5,6$:

$$m(X)_{n-2t} = m_0 + m_1 X^1 + ... + m_{n-2t-1} X^{n-2t-1}$$

Therefore:

with t=1 $m(X)_{k=13} = m_0 + m_1X + m_2X^2 + ... + m_{12}X^{12}$

$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots$

with t=6 $m(X)_{k=3} = m_0 + m_1X + m_2X^2$

3.3.2 Adaptive Reed Solomon Encoding

To generate the encoding in systematic form we have to shift a polynomial message *m(X)* into the rightmost *k* stages of a codeword register, and then append a parity polynomial, *p(X)*, by placing it in the leftmost *n-k* stages. Therefore, we multiply *m(X)* by X^{n-k}, thereby the message polynomial is positioned on the left part of the code-word, before of *n-k*.

To generate a parity symbols we have to divided the message $X^{n-k}m(X)_{n-2t}$ *by* $g(X)_{n-2t}$

$$p(X)_{n-2t} = X^{n-(n-2t)}m(X)_{n-2t} \bmod g(X)_{n-2t} \qquad \text{and } t=1,2,3,4,5,6$$

Therefore:

with *t=1* $p(X)_{k=13} = X^2 m(X)_{k=13} \bmod g(X)_{k=13}$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

with *t=6* $p(X)_{k=3} = X^{12} m(X)_{k=3} \bmod g(X)_{k=3}$

for each *t=1,2,3,4,5,6* we have *t* different code-words:

$$C(X)_t = p(X)_t + X^{n-2t}m(X)_t, \qquad \text{for } t=1,2,3,4,5,6$$

3.3.3Adaptive Reed Solomon Decoding

The A-RS decoder is the same as the RS standard decoder except for the number of syndrome equations, which changes according to the value of *t*.

The summary diagram of the encoding and decoding process of an A-RS code is shown in *Fig. 3.4*.

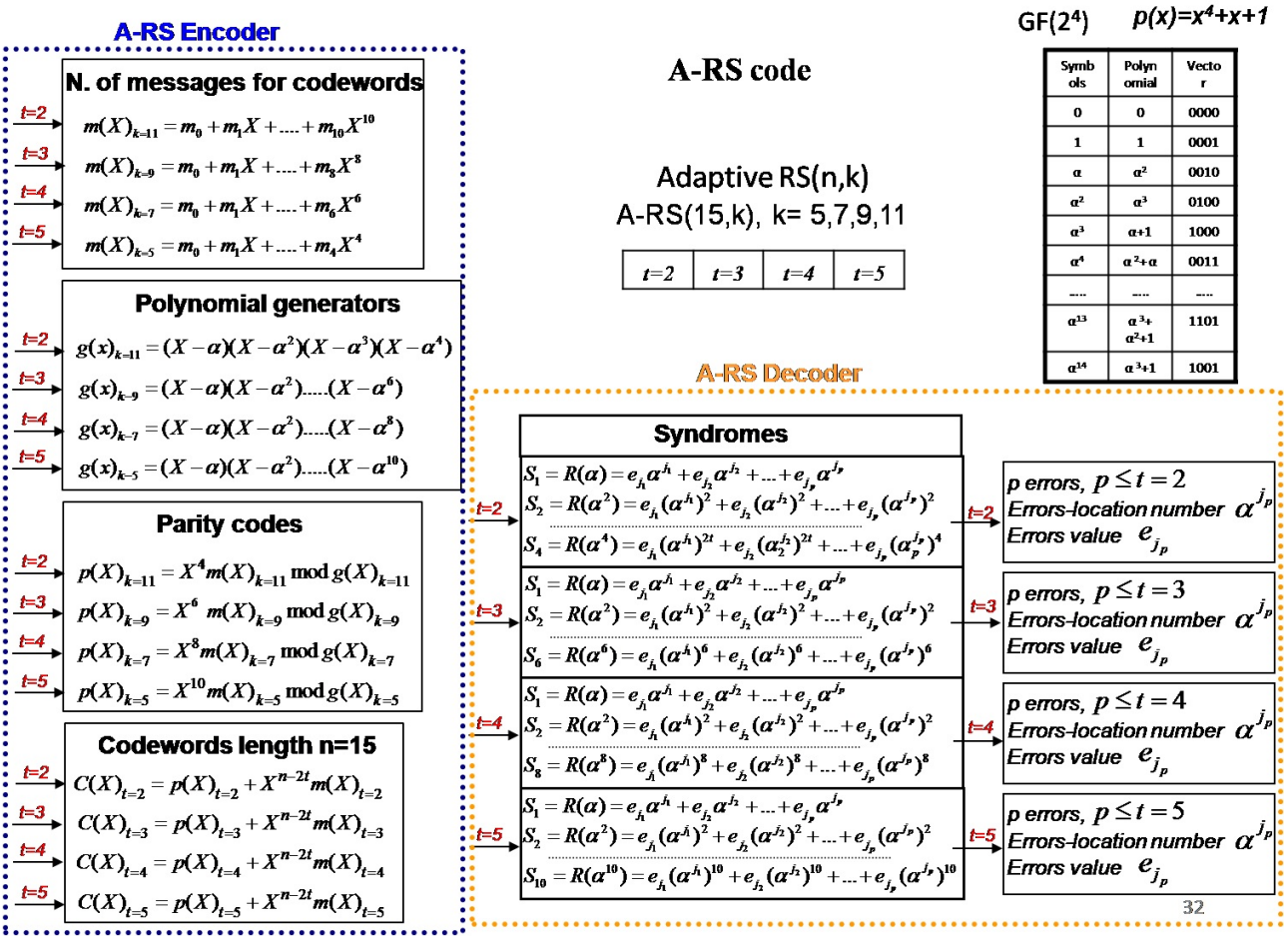


Fig. 3.4 – Adaptive-Reed Solomon (15,k).

CHAPTER 4

4. Our Contribution: a New Error Correction Approach for Optical Discs

In the ensuing chapter we will introduce a new system, whose application will redistribute *parity symbols* through an "intelligent" criteria, based on the information obtained from the degradation function and the analysis of the experimental tests.
As a starting point, we will analyse encoder systems and correction systems that are used for different supports – BDs, DVDs, and CD.
The information coming from the host reaches the optical disc coding system in the form of a *Frame* constituted by *2048 bytes*. After being formatted in a number of steps, several frames form a Data Block which is protected by a specific *Reed Solomon code (RS)*, depending on the optical disc type.
By analysing the standard correction system a new A-RS software will be defined, one for each type of optical disc. This A-RS software will simulate the behaviour of a hypothetical A-RS hardware that should be implemented onto the burning device. The A-RS software has to be defined so that its correction capability is comparable to that of the A-RS hardware.

On one hand, an A-RS encoder (software) codifies the *information data* in *code-words*, before they are burned onto the disc. On the other hand, an A-RS decoder translates the *code-words* recorded on the disc into data, while correcting possible errors.
In this chapter, the analysis of the previously obtained data from the degradation function will provide a parity file, that contains the *parity symbols*. These codes, along with information data, will be used by the A-RS code to generate new *code-words*. The new *code-words* will be contained into a binary file, then this file will be converted into an ISO image and so it will be possible to record it onto the discs.

4.1 Blu-Ray discs (BDs)

4.1.1 Data Encoding with Standard RS Code

The *BD* [14] coding system is reached by information data coming from the host *32 Frames*. To each of the *32 Frames*, *4 bytes of Error-Detection Code (EDC)* are added.
The *32 Frames* constitute the *Long Distance Code Block (LDC Block)* which adds extra protection codes to the data information. The *LDC Block* (*see Table 4.1*) is a two-dimensional array constituted by *304 columns*, *248 bytes* each, leading to a total of *75.392 bytes* (*65536* of which are

information data). In the *LDC Block* the information data are arranged in *304 column* of *216 bytes (k)*. To each of the *216 bytes* is applied an *RS(248,216)* which adds to each column *32 bytes* of parity (*p*). This code can correct up to *16 errors or erasures symbols (t)* in each *code-word* having a length of *248 bytes (n)*.

		←			304 columns			→
		code word 0	code word 1	:	code word *L*	:	code word 302	code word 303
↑	↑	$e_{0,0}$	$e_{0,1}$	:	$e_{0,L}$	:	$e_{0,302}$	$e_{0,303}$
	216	$e_{1,0}$	$e_{1,1}$	:	$e_{1,L}$	:	$e_{1,302}$	$e_{1,303}$
	rows	$e_{2,0}$	:	:	:	:	:	:
	with	:	:	:	:	:	:	:
1 LDC	data	:	:	:	:	:	:	:
code word	↓	$e_{215,0}$	$e_{215,1}$	:	$e_{215,L}$	:	$e_{215,302}$	$e_{215,303}$
= 248	↑	$p_{216,0}$	$p_{216,1}$	:	$p_{216,L}$	:	$p_{216,302}$	$p_{216,303}$
bytes	32	:	:	:	:	:	:	:
	rows	:	:	:	:	:	:	:
	with	:	:	:	:	:	:	:
	parity	:	:	:	:	:	:	:
↓	↓	$p_{247,0}$	$p_{247,1}$	:	$p_{247,L}$	:	$p_{247,302}$	$p_{247,303}$

Table 4.1 LDC Block structure.[14]

The *LDC* RS code is defined over the finite field $GF(2^8)$. The non-zero elements of the field $GF(2^8)$ are generated by primitive element α that is a root of the primitive polynomial *p(x)*:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1$$

The symbols of $GF(2^8)$ are represented by bytes, in polynomial-base representation, with $(\alpha^7, \alpha^6, \alpha^5, \cdots, \alpha^2, \alpha, 1)$, as basis. The root α is represented as: $\alpha = 00000010$. Each *code-word* can be represented by a polynomial of degree 247 where the highest degrees correspond to the information part of the vector and the lowest degrees correspond to the parity part of the vector (*RS in systematic form*).

Each polynomial is a multiple of the generator polynomial g(x).

$$g(x) = \prod_{i=0}^{31} (x - \alpha^i).$$

4.1.2 A New Approach: Data Encoding with A-RS Code

In order to carry out a simulation and to burn test discs by using the proposed new correction system, the data information are substituted by *code-words* (generated by an A-RS software code) before the burning task. For the RS code internal to the BD device (RS hardware) the "virtual" *code-words* (those generated by the A-RS code) are completely indistinguishable from "standard" *information data*.*

Since the standard *Data Block* consists of *32 Frames*, we have chosen to subdivide each *Frame* into *256 blocks (B)* constituted by *256 bytes* each (see *Fig. 4.1*).

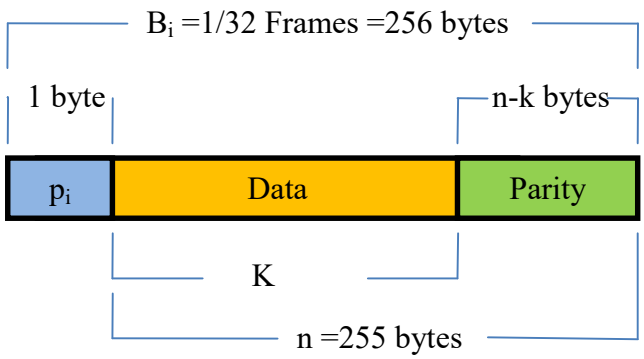


Fig. 4.1 - The smallest unit of the A-RS code is constituted by 256 bytes-lock Block (B), whose first byte encodes the parity value, while the code-word is encoded in the rest of the block.

In the first byte of each *B block* it is memorized the incremental *parity value* (p_i) is stored, and it will be used by the A-RS software for the decoding task. In the other *255 bytes* are stored the *code-words* generated by the correction code software A-RS(255,k_0) where k_0 (default bytes number of *information data*) must have a value such that the RS hardware and A-RS software are equivalent, in terms of correction capacity. The parameter which allows for the comparison of the two codes is named *Block Correction* $(BC)_{(\%)}$. $(BC)_{(\%)}$ is defined as *t/n*, where t represents the number of bytes that each *code-word* can correct, and where n represents the length of the *code-word* expressed as number of bytes. Since the *RS(248,216)* has a *BC=6.45%*, the best value of k_0 is *238*.. In this case the *ARS(255,238)* has a *BC=6.66%.*

The number of parity symbols assigned by the A-RS software to the *code-words* contained in each *Data Block* is calculated by an optimization algorithm which, starting from the errors in the *degradation function* of each *Data Block*, moves the parity bytes from the *Data Blocks* containing the smallest number of errors (adding further *k bytes* instead of them) to the *Data Blocks* having the highest number of errors (decreasing in this case the number of *k bytes*).

The *8 B Blocks* consisting of *256 bytes* will form a single Frame of *2048 bytes* and the 32 Frames will constitute a *Data Block* of *65536* bytes (*see Fig. 4.2*) .

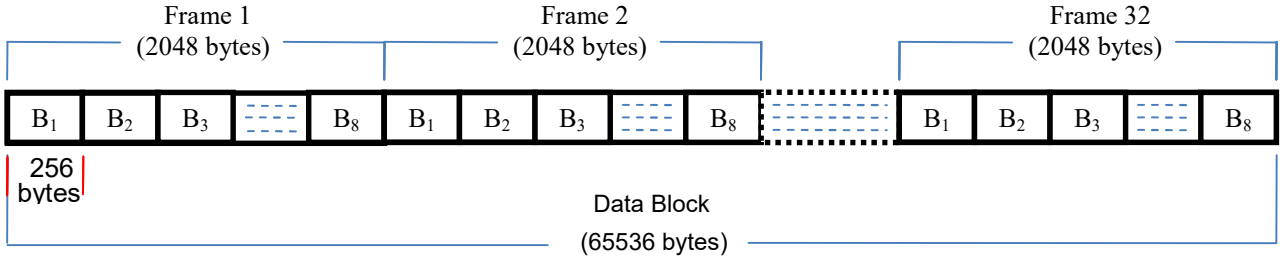


Fig. 4.2 – Data Blocks in a Blu-Ray disc.

Therefore the total number of *Data Blocks* burned to the disc is equal to *379541*. This will generate a file of *24.873.598.976 bytes* (see *Fig. 4.3*).

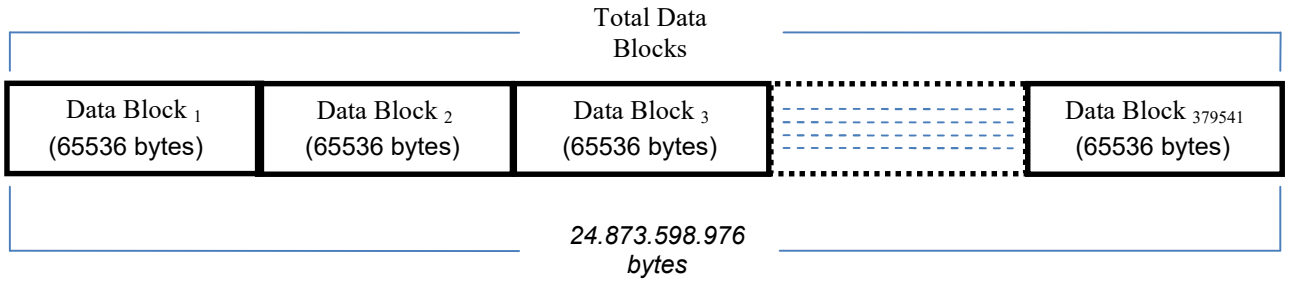


Fig. 4.3- Data Blocks in a Blu-Ray disc.

4.1.2.1 How to Pack the Binary Data File

An A-RS hardware allows for storage of *24.873.598.976 bytes* of total data information on the disc (*379541 Data Block* x *65536 bytes*) whereas, using an A-RS software, total number of the *bytes* of information data will decrease due to the substitution of some data with the *code-words*.

The total number of data *bytes* recorded by the A-RS encoder can still be calculated by assuming that $k=k_0$ for each *code-word*, even if k has in fact different values. The algorithm just substitutes data bytes with parity bytes (or vice-versa) in each *code-word*, and so the sum of all k still is equal to nk_0. This is due to the fact that for each $k=k_0$-2, there will always be a $k=k_0$+2, and so on. The result is a total of information data equal to *21.472.911.616* bytes (*221 bytes* x *256 B* x *379541 Data Block*). The number of bytes containing the parity value have to subtracted to the total amount of *information data*.

Eventually, the file of recordable data obtained through the new encoding will have *21.375.749.120 bytes* (*21.472.911.616 – 97.162.496*). Therefore starting from a file of random data of

21.375.749.120 bytes, the A-RS code will generate only *code-words* of *255 bytes,* with *k bytes* of data which can have:

- k equal to k_0 in the case of default parity=*34 bytes*,
- k smaller than k_0 if it is necessary to increase the parity in the *i-th Data Block*,
- k greater than k_0 if it is necessary to decrease the parity in the *j-th Data Block*).

Where i indicates any Data Block that needs a higher redundancy, while j indicates any Data Block that can afford a lower redundancy.

In any case the sum of all the $(k_{DBi} + k_{DBj})/DB_n = 221$, where DB_n = total number of *Data Blocks* of the whole disc.

The test *binary file* (*Fig.4.4*) that contains the *code-words* generated by the A-RS will be constituted by *24.873.598.976 bytes* which after the conversion to *ISO image* (*Fig.4.5*), have been burned onto the *BDs* discs.

4.1.2.2 How to Pack the Parity File

The *parity file* generated by the *degradation function* for the *BD* contains exactly *379541* incremental parity values p_i. Each of these p_i indicates how much the default parity p_0 must be increased (or decreased) in a *Data Block*. Since p_0 is the default parity of the A-RS code ($p_0 = n - k_0 = 34$), the p_i value (with $i=1,2,...,379541$) will be algebraically added to p_0 and the result will be applied to the A-RS code in order to generate the *code-words* with the defined correction capacity.

Each *byte* contained in the parity file is a relative even number since ±2 *parity bytes* increase or decrease the code correction capacity of *1 byte*.

The values contained in the *parity file* are inferred from the *degradation function* which contains the prediction of the errors produced in each *Data Block* of the *BD* after the accelerated aging process.

Since *304* errors in a *LDC Block* correspond to *1 error byte* in each *code-word,* this value has been used as a threshold value in order to increase or decrease the parity symbols of the A-RS code. Thus, in order to correct *304 errors* in a *LDC Block, 2 parity bytes* must be added to each of the *code-words* of the *LDC Block.*

These *2 parity bytes* will be taken from *Data Blocks* that have a number of errors below *304*. If there are not enough *Data Blocks* with less than *304* errors available, the parity will be taken from *Data Blocks* with a number of errors lower than *608*, then lower than *912*, and so on until reaching a threshold of *4256* (value that would leave just 2 bytes of residual capacity correction in each *Data Block*).

A small part of the binary file is shown as example in *Fig 4.4*.

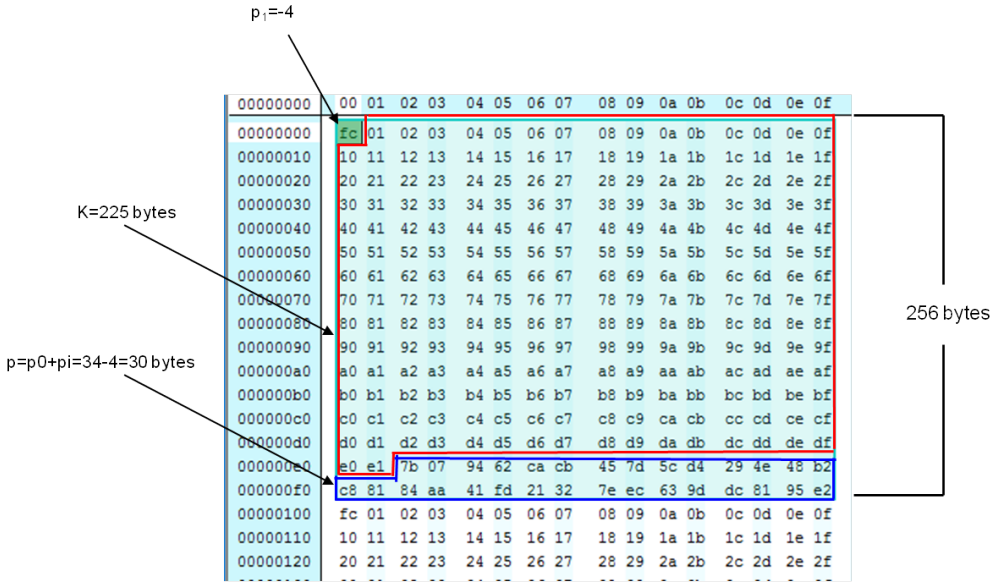


Fig. 4.4 – Part of the binary file of a Blu-Ray disc.

A section of the ISO file [64] of a Blu-Ray disc is shown in *Fig.4.5:*

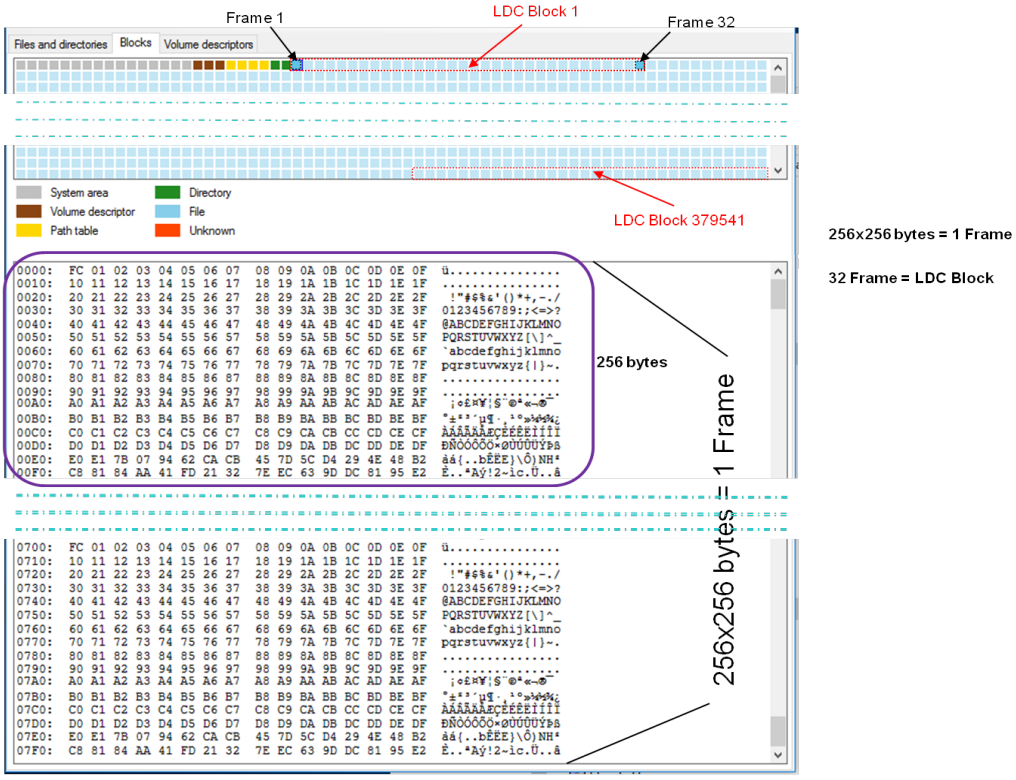


Fig. 4.5 – Part of the ISO file of a Blu-Ray disc.

4.2 Digital Versatile Discs (DVD$_s$): Data Encoding and Error Correction Code

4.2.1 Data Encoding with Standard RS Code

The *DVD* [65] coding system is reached by *information data* coming from the host *16 Data Frames* (*1 Data Frame=2048 bytes*).

To each of the 16 *Frames*, 4 bytes of the *Identification Data* (ID), 2 bytes of the *Error Detection Code* (IED), 4 bytes of the *Copyrigth Management Information* (CPR_MAI) and 4 bytes of the *Error Detection Code* (EDC) are added (*Table 4.2*).

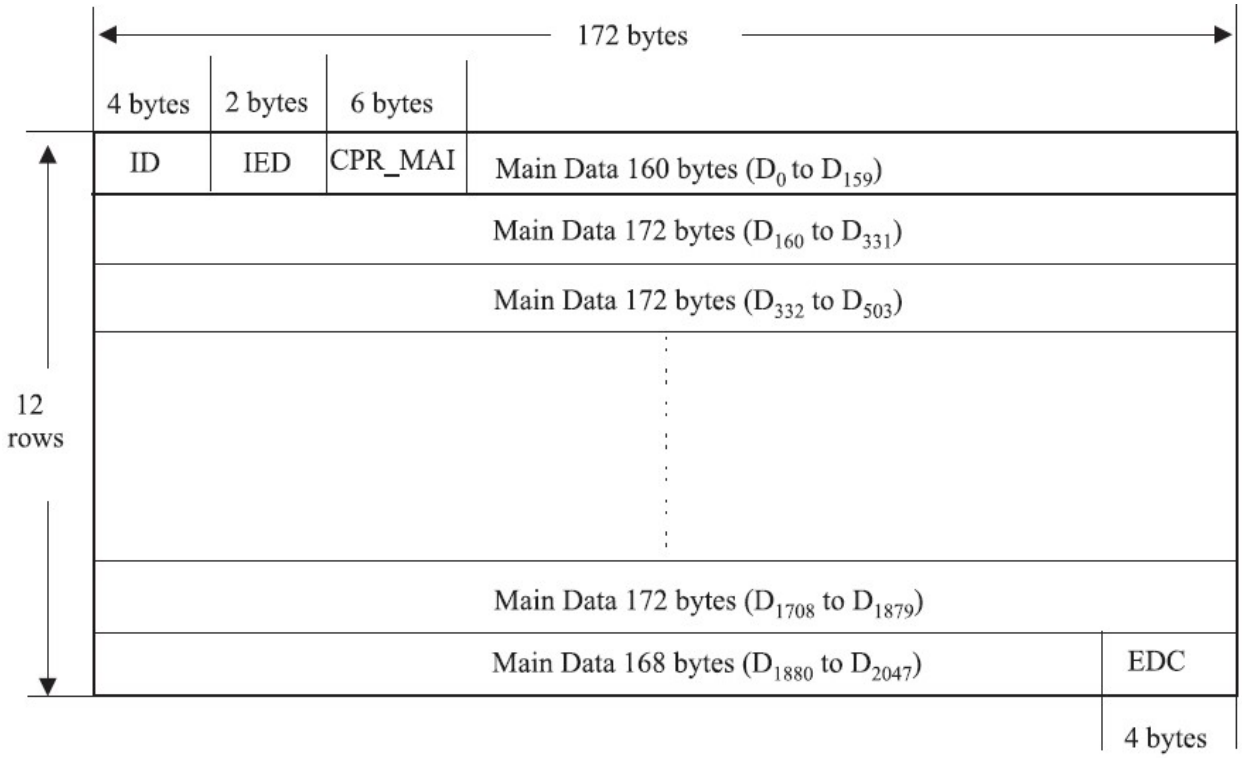


Table 4.2- DVD Data Frame.[65]

The 16 *Frames* constitute the *Error Correction Code Block* (*ECC Block*) witch adds extra protection codes to the *information data*. The *ECC Block* (*see Table 4.3*) is a two dimensional array constituted by *208 rows*, *182 bytes* each, leading to a total *37.856 bytes* (*32.768* of which are *information data*).

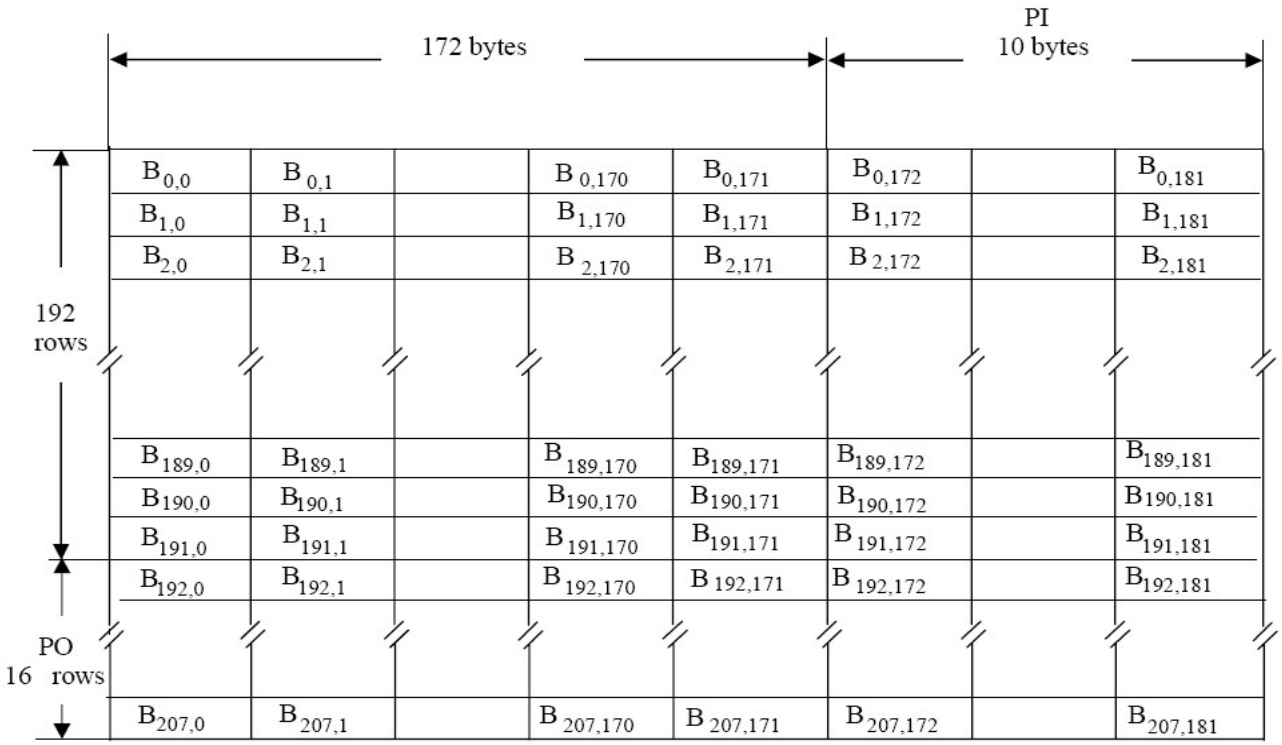


Table 4.3 - ECC Block.[65]

In the ECC Block the information data are arranged in *192 rows* of *172 bytes*.

To each of the *172 bytes*, denoted as *k*, is applied an *RS(182,172)* which adds *10 bytes of Parity Inner* (*PI*).

This code can correct up to *5 errors* or erasure symbols (denoted as *t*) in each *code-word* having a length of *182 bytes*, denoted as *n*.

An *RS(208,192)* adds, to each of 182 columns constituted by *192 bytes*, further 16 bytes of *Parity Outer* (*PO*). This code can correct up to 8 errors or erasure symbols in each *code-word* having a length of *208 bytes*.

The *PO* and *PI* shall be obtained as follows: in each of columns *j=0 to 171*, the *16 PO bytes* are defined by the remained polynomial to form the outer code *RS(208,192)*.

In each of *rows i=0 to 207*, the *10 PI bytes* are defined by the remainder polynomial to form the inner code *RS(182,172).*

4.2.2 A New Approach: Data Encoding with A-RS Code

In order to carry out a simulation and burn test discs using the proposed new correction system, the information data are substituted by *code-words* (generated by an A-RS software code) before the burning task. For the RS code internal to the *DVD device* (RS hardware) the 'virtual' *code-words* (those generated by the A-RS code) will continue to be treated just as data.

We decided to simulate only the *RS(182,172)* since it is the first correction code system that applies to the data in the *ECC code*. Since the standard *ECC Block* consists of *16 Frames*, we have chosen to subdivide *32768 bytes* of information data into *190 code-words* of length *172 bytes* (see *Fig. 4.6*).

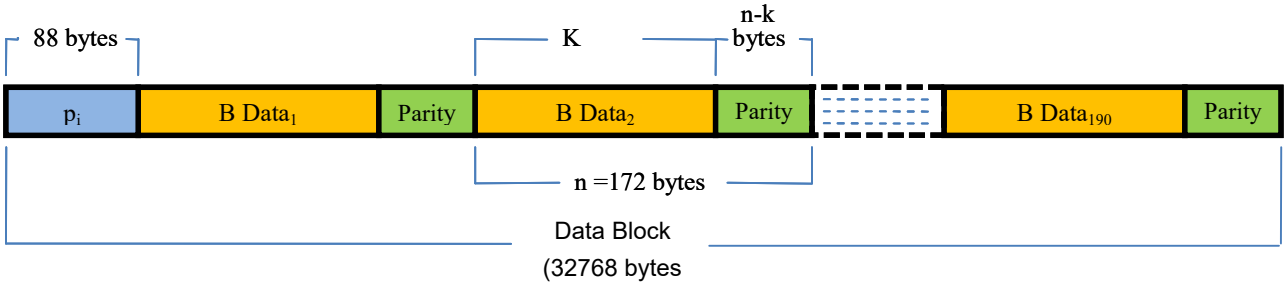


Fig. 4.6 - Data Blocks = 190 B Data (32.680) + p_i (88 bytes).

In the first *88 bytes* of each *Data Block* is memorized the incremental parity value (p_i) which will be used by the A-RS software during the decoding task. In the remaining part of the *Data Block* are stored the *190 code-words* generated by the correction code software A-RS(172,k_0) where k_0 (default bytes number of data information) must have a value such that the *RS hardware* and *A-RS software* are equivalent in terms of correction capacity. The parameter that allows for the comparison of the two codes is named *Block Correction* $(BC)_{(\%)} = t/n$ and, since the RS(182,172) has a BC=2.74%, the best value of k_0 is *162* indeed. In this case, the A-RS(172,162) has a BC=2.90%.

The number of parity symbols assigned by the A-RS software to the *code-words* contained in each *Data Block* is calculated by an optimization algorithm which, starting from the errors in the *degradation function* of each *Data Block*, substitutes the *k* parity bytes from the *Data Blocks* containing the smallest number of errors with *k bytes* in the *Data Blocks* having the highest number of errors.

Therefore the total number of *Data Blocks* burned to disc is *140996.* This will generate a file of *4.620.156.928 bytes* (see *Fig. 4.7*).

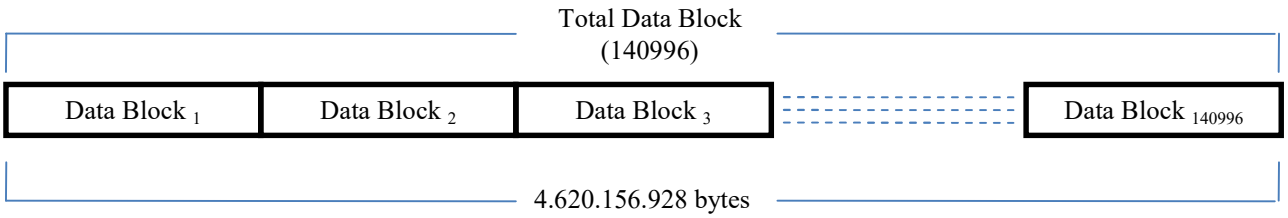


Fig. 4.7 – Data Blocks of a DVD disc.

4.2.2.1 How to Pack the Binary Data File

The A-RS hardware allows for the storage of 4.620.156.928 *bytes* of total data information on the disc (*140996 Data Blocks x 32768 bytes*) whereas, using the A-RS software, the number of *bytes* of the total information data will result decreased due to the substitution of data with the *code-words*.
The total number of data *bytes* recorded by the *ARS encoder* can be calculated by considering in all *code-words* $k=k_0$ (as the algorithm just shifts the parity from a data block to another). The result is a total of information data equal to 4.339.856.880 bytes (*30780 bytes* x *140996 Data Block*), to which the bytes containing the parity value must be subtracted.
Definitely, the file of the recordable data obtained by the new encoding will have *4.339.732.112 bytes* (*4339856880 -12.407.648*). Therefore, starting from a file of random data of *4.325.757.280 bytes,* the A-RS will generate only *code-words* of *172 bytes,* with *k bytes* of data which can have:

- k equal to k_0 (162 bytes) in the case of default parity=*10 bytes* (p_0),
- k smaller than k_0 if it is necessary to increase the parity in the *i-th Data Block*,
- k greater than k_0 if it is necessary to decrease the parity in the *j-th Data Block*).
  Where i indicates any Data Block that needs a higher redundancy, while j indicates any Data Block that can afford a lower redundacy.

In any case $(k_{DBi} + k_{DBj})/k_{DBn} = 162$, where k_{DBn} = total number of *Data Blocks* of the whole disc.
The test *binary file* that contains the *code-words* generated by the *ARS* will be constituted by *4.620.156.928 bytes* which, after the conversion to *ISO image*, have been burned onto the *DVDs* discs.

4.2.2.2 How to Pack the Parity File

The *parity file* generated by the *degradation function* for the *DVD* contains *140996* incremental parity values p_i. Each of these p_i indicates how much p_0 must be increased or decreased in a *Data Block*. Since p_0 is the default parity of the A-RS code ($p_0 = n - k_0 = 10$), the 'p_i' value (with *i=1,2,..., 140996)* will be algebraically added to p_0 and the result will be applied to the A-RS code in order to generate the *code-words* with the defined correction capacity.
Each *byte* contained in the *parity file* is a relative even number since ±2 *parity bytes* increase or decrease the code correction capacity of *1 byte*. The values contained in the *parity file* are calculated from the *degradation function* which contains the prediction of the errors produced in each *ECC Block (PI)* of the *DVD* after the accelerated aging process. Since *172* errors in a *ECC Block* correspond to *1 error byte* in each *PI code-word,* this value has been used as a threshold in order to

increase or decrease the parity symbols of the A-RS code. This means that, in order to "delete" *172 errors* in a *LDC Block, 2 parity bytes* must be added to each of the *code-words* of the *Data Block*. These 2 parity bytes will be taken from Data Blocks that have a number of errors lower than *172*.
If the amount of Data Blocks having a number of errors less *172* is not sufficient to add parity to the Data Blocks containing higher numbers of errors, the same algorithm will be reiterated by taking the parity from *Data Blocks* having a number of errors lower than *344,* than *516*, and so on, up to a maximum value of *688* (leaving 2 bytes of residual capacity correction in each *Data Block*).

4.3 Compact Discs Rom (CD-ROM): Data Encoding and Error Correction Code

4.3.1 Data Encoding with Standard RS Code

The *CD-Rom* [66] encoding system is reached by information data coming from the host in Frames (of *2048 bytes*). To each *Frame*, *12 bytes* for synchronization plus *4 bytes* for header are added (see Table 4.4) for a total of *2964 bytes*.

Sector: 2 352 bytes

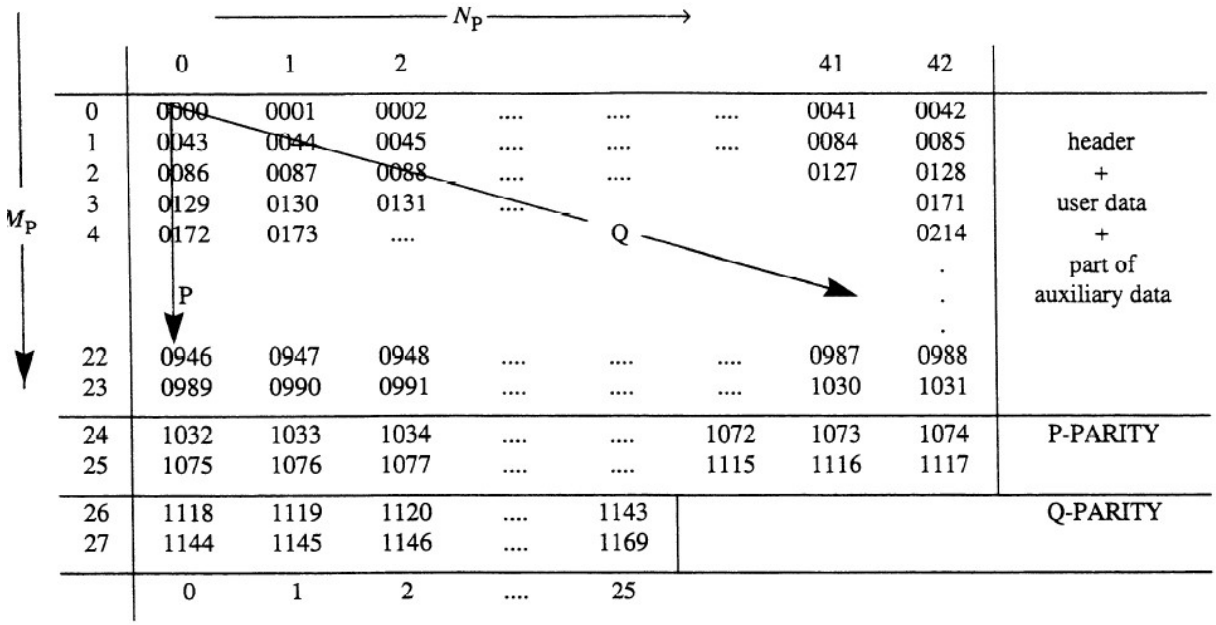
Sync	Header		User Data	EDC	Intermediate	P-Parity	Q-Parity
	Sector Address	Mode					
12 bytes	3 bytes	1 (01)-byte	2 048 bytes	4 bytes	8 bytes	172 bytes	104 bytes

Table 4.4 Sector of a Digital Data Track (mode 1).[66]

The *2064 bytes* are divided in two groups of *1032 bytes*. Each group constitute the *Error Correction Coding* (*RSPC*) which adds extra parity protection codes to the information data (see *Table 4.5*) .

In the RSPC the *information data* are arranged into *43 column* of *24 bytes* (*1032 bytes* of which *1024* are *information data*) . To each of the *24 bytes* is applied *an RS(26,24)*, named C_1, which adds *2 bytes* of parity (*P-Parity*). C_1 can correct up *1* error or erasure symbol in each *code-word* having length of *26 bytes*. Another *RS(45,43)*, named C_2, adds to each of the *26* diagonals of the matrix further *2 bytes* of parity (*Q-Parity*). C_2 can correct *1* error or erasure symbol in each code-word having a length of *45 bytes*.

The RSPC is a product code over $GF(2^8)$ producing *P-parity* and *Q-Parity bytes*. The $GF(2^8)$ field is generated by the primitive polynomial : $P(x) = x^8 + x^4 + x^3 + x^2 + 1$. The primitive element $\alpha = (00000010)$ where the right-most bit is the least significant bit.



M_P \ N_P	0	1	2	....	....	....	41	42	
0	0000	0001	0002	....	....	....	0041	0042	header + user data + part of auxiliary data
1	0043	0044	0045	....	....	....	0084	0085	
2	0086	0087	0088	....	....		0127	0128	
3	0129	0130	0131	....				0171	
4	0172	0173	....		Q			0214	
	P							.	
								.	
								.	
22	0946	0947	0948	....	....	....	0987	0988	
23	0989	0990	0991	....	....	....	1030	1031	
24	1032	1033	1034	....	....	1072	1073	1074	P-PARITY
25	1075	1076	1077	....	....	1115	1116	1117	
26	1118	1119	1120	....	1143				Q-PARITY
27	1144	1145	1146	....	1169				
	0	1	2	....	25				

Table 4.5 - Error Correction Coding.

4.3.2 A New Approach: Data Encoding with A-RS Code

In order to carry out a burn test simulation using the proposed correction system, the information data are substituted by *code-words* (generated by an A-RS software code) before the burning task. For the RS code internal to the *CD device* (RS hardware) the *virtual code-words* (those generated by the A-RS code) will continue to be treated just as data.

The *ISO 18927* defines the Block Error Rate (BLER) as the number of erroneous blocks per second measured at the input of the C_1*-decoder* at the standard (1X) data rate. The total number of blocks per second in input to the C_1*-decoder* is *2048 bytes/block* × *(75 blocks/s)* = 153600 bytes/s of *information data*.

IEC 60908:1999 states that the *BLER* averaged over any *10s* should be less than $3x10^{-2}$ *blocks/s*. Thus, the maximum value of *BLER* is $7350 \times 3x10^{-2} = 220$ *block per second*.

Furthermore, following the International Standard the data are considered to have reached end-of-life when the *BLER*, measured as erroneous blocks per second, exceeds *220* anywhere on the disc. A *BLER* of *220* is an arbitrary level chosen as a predictor of the onset of uncorrectable errors and thereby of the end-of-life.

Since the evaluation of the errors is made on *153600 bytes* (for convenience we consider *153660* the number bytes in the *Data Block*) we have chosen to subdivide the *Data Blocks* in *6400 code-words* of *24 bytes* (see *Fig.s 4.8 – 4.9*).

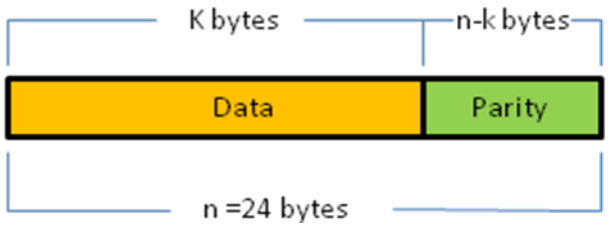


Fig. 4.8 - RS(24,k_0) code-word.

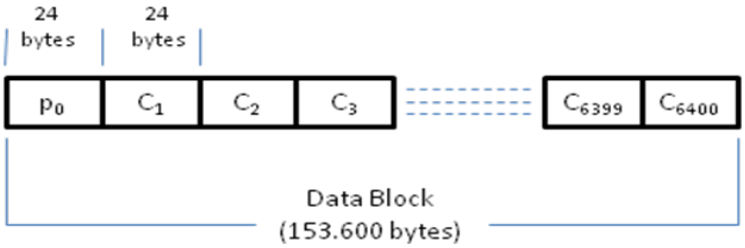


Fig. 4.9 - Data Block of a CD-ROM.

In the first *code-words* (*Fig. 4.9*) is stored the incremental parity value (p_i) which will be used by the A-RS Software for the decoding task. The other code-words are generated by the correction code software A-RS(24,k_0) where k_0 (default number of bytes of the information data) must have a value such that the *RS hardware* and A-RS software are equivalent, in terms of correction capacity. The parameter which allows for the comparison of the two codes is named *Block Correction* $(BC)_{(\%)}=t/n$.

Since C_1 error can be indicated by C1= *E11, E21, E31* (where E11 is the number of single errors per second in the C1 Decoder, *E21* is the number of double errors per second in the C1 Decoder and and *E31* is the number of triple or more errors per second in the C1 Decoder) and since each *RS hardware* can correct just *1* erroneous symbol, we have chosen, in this simulation, to consider only the C_1 code with 2 errors correction capability. In this case the *RS(26,24) hardware* becomes an *RS(26,22)* and has a BC=15.38%. Thus the best value for A-RS software with *n=24* is *RS(24, 20)* which has a *BC=16.66%.*

The number of parity symbols assigned by the A-RS software to the *code-words* contained in each *Data Block* is calculated starting from the errors in the *degradation function* of each *Data Block.*

The missing *k bytes* in the code-word are taken from empty space of the disc (decreasing the total capacity for the data).

Therefore the total number of *Data Blocks* burned in the disc is *4440.* This will generate a file of *681.984.000 bytes* decreased by the total number of bytes used for the extra parity (see *Fig. 4.10*).

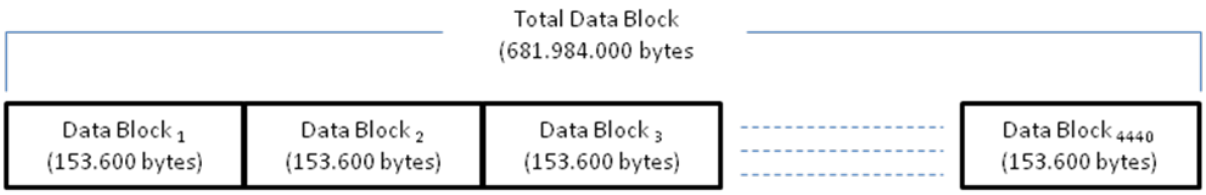


Fig. 4.10 –Schematisation of all Data Blocks of a CD-ROM.

4.3.2.1 How to Pack the Binary Data File

Synthesizing the above discussion, the A-RS hardware allows for the storage of *681.984.000 bytes* of total information data on the disc (*4440 Data Blocks x 153.600 bytes*) whereas using the *ARS software,* the number of *bytes* of the total information data will result decreased due to the substitution of data with the *code-words*.

Since the A-RS(24,k_0) has just *4 parity bytes* in each *code-words*, we have chosen to increase the parity in the Blocks (which have a higher number of errors) taking the extra bytes from the empty space of the disc. Thus, the total number of *information data* bytes, before the burning task, has to be calculated by including the extra parity bytes.

In any case the A-RS will generate only code-words of *24 bytes* with *k bytes*. The *k bytes* can be:

- equal to k_0 (*162 bytes*) in the case of default parity=*10 bytes* (p_0),
- smaller than k_0 if it is necessary to increase the parity in another *Data Block*,
- greater than k_0 if it is necessary to decrease the parity in another *Data Block*.

Thus the test binary file that contains the *code-words* generated by the A-RS is ready to be recorder onto the CD-R disc,s after the *ISO image* conversion.

4.3.2.2 How to Pack the Parity File

The *parity file* generated by the *degradation function* for the *CD* contains *4441* incremental parity values p_i. Each of these p_i indicates how much p_0 must be increased or decreased in a *Data Block*.

Since p_0 is the default parity of the A-RS code ($p_0 = n - k_0 = 10$), the 'p_i' value (with $i=1,2,\ldots,$ 140996*)* will be algebraically added to p_0 and the result will be applied to the A-RS code in order to generate the *code-words* with the defined correction capacity.

Each *byte* contained in the parity file is a relative even number since ±2 *parity bytes* increase or decrease the code correction capacity of *1 byte*. The values contained in the *parity file* are calculated from the *degradation function* which contains the prediction of the errors produced in each *RSPC Block* of the *CD* after the accelerated aging process. Since *50* errors in a *RSPC Block* correspond about to ¼ of the *max BLER*) this value has been used as a threshold in order to increase or decrease the parity symbols of the A-RS code. This means that, in order to correct 50 BLER errors in a *LDC Block, 2 parity bytes* must be added to each of the *code-words* in the *Data Block*.

CHAPTER 5

5. Tests and Data Analyses

This chapter is devoted to the investigation of the evolution of the distribution of errors in optical discs. For this purpose a large number of different types of optical discs will undergo an accelerated aging cycle. A failure in the reading process is produced when the reflection of the incident laser beam is not information enough to discriminate correctly between land areas (areas of maximum reflection) and pit areas (areas of minimum reflection). We will show how the degradation of the materials composing the supports induces the creation of particular zones with higher error rate which we call *critical zones*. These degraded areas produce errors in some Data Blocks. The Data Blocks of the discs are numbered as shown in *Fig.5.1*.

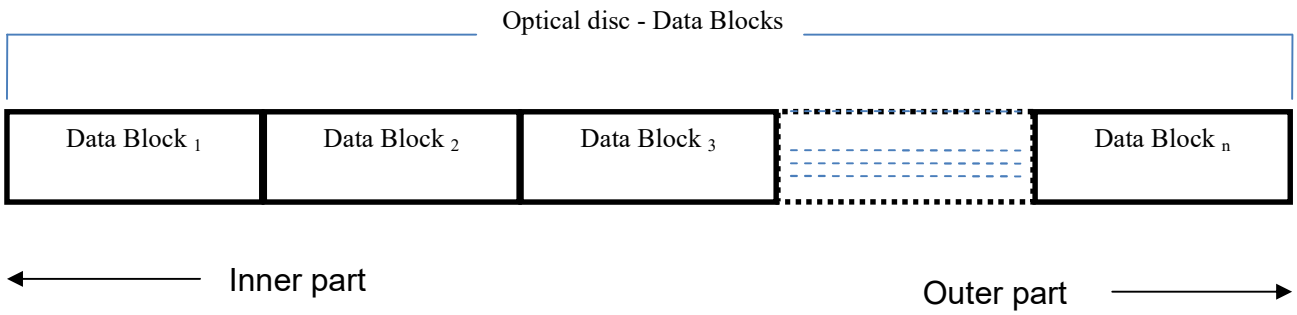


Fig. 5.1 - Data Blocks of an Optical disc..

Different types of recordable discs have been subjected to the tests: *48 Blu-Ray recordable discs (BD-R discs),* brand 1, *48 Blu-Ray recordable discs (BD-R discs),* brand 2, and 48 *Recordable Compact Discs (CD-R),* brand 3. For comprehension purposes, in the following chapter all the recordable discs will be addressed to as discs.
For each type of disc the errors have been read by the ATM *Automatic Testing Machine*, before and after an accelerated aging stage the climatic chamber.

The ATM allows to execute automatic tests without dust, and by controlling the temperature and the humidity of the environment so that all the burning stages are performed in comparable conditions.

In this chapter the files containing the number of errors found before and after having performed the accelerated aging stage will be illustrated and it will be possible to observe that:

- Before the aging, the discs present a really low number of errors (much lower than the correction capability of the correction code). At this stage, errors are uniformly distributed on the surface of the disc.
- After the aging process, on the contrary, errors clearly accumulate in a specific area of the discs. In this area, located in the external part of the disc, the errors increase exponentially. This behaviour characterized a statistically relevant amount of discs (more than 90%). In the remainder, we will refer to the area in which errors accumulate after the accelerated aging as the *critical area*. Symmetrically, the region of the surface of the disc where the distribution of the errors remains uniform will be named *safe area*.

Tests performed before the accelerated aging process indicate that the standard correction code system can correct errors that were created in the burning process. In fact, in BDs the maximum number of errors is equal to 600, and for CDs it is equal to 25. In both cases the values are sensibly lower than the correction limit, respectively equal to 4864 and 220.
Tests performed after the accelerated aging process indicate that the standard correction code system is not able to correct all errors. It has been determined that BDs have, in some areas, a number of errors higher than 9000, and that CDs have a number of errors higher than 800.

5.1 Blu-Ray Discs - Brand 1

5.1.1 Before Accelerated Aging

Before proceeding with the general analysis that was performed on all discs, the analysis of the errors on a single BD disc brand 1 will be illustrated as example.
In *Fig 5.2* it is possible to notice that the errors are uniformly spread over the whole surface of the optical disc – *i.e.* there are no particularly affected areas. The maximum number of errors (Max Error) that was individuated in a single Data Blocks is equal to 290.

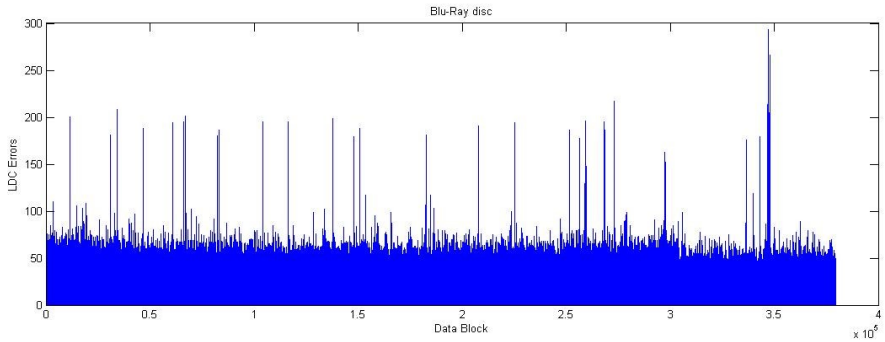


Fig. 5.2 - Blu-Ray, brand 1- Data Block errors before accelerate aging- max error<300.

In *Fig 5.3* it is possible to see that, after the burning process, only a small percentage of Data Blocks – 21.33% – contains at least one error.

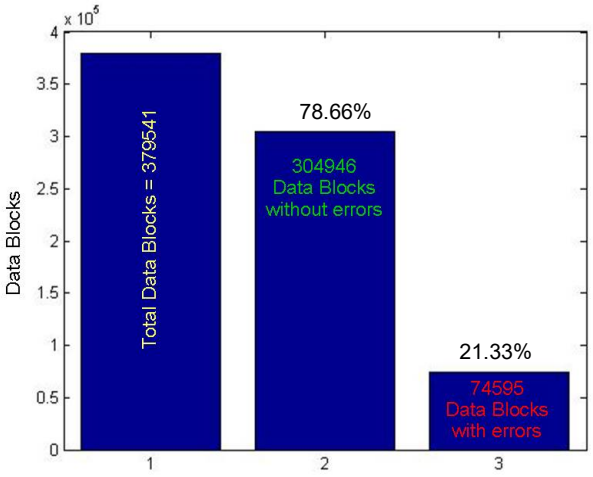


Fig. 5.3 - Blu-Ray disc, brand 1 - only 21.33% Data Blocks has at least one error.

Therefore, in this disc the 78.66% of the Data Blocks does not have errors, and the correction code still has a residual capability equal to the 100%. In the remaining Data Blocks the number of errors belong to the range between 1 and 290, and the correction code has a residual capability equal to the 94% at least.
This is important since the higher is the residual capability, the better is the life expectancy of an optical disc.

In *Fig. 5.4a* the histogram of the errors and in *Fig. 5.4b* the best fit is the *lognormal probability density function* [67] are shown.

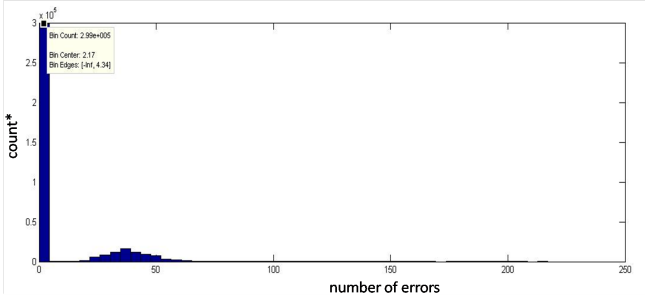


Fig. 5.4a - Blu-Ray disc 1, brand 1. Data Blocks errors distribution.
count=count of Data Blocks for each bin.*

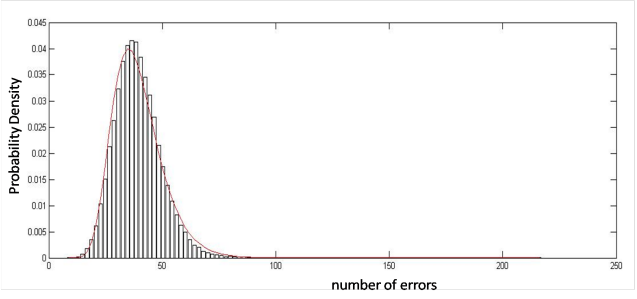


Fig. 5.4b - Blu-Ray disc 1 brand 1. Lognormal distribution.
(without the Data Blocks = 0 errors).

The lognormal distribution should be used to characterize the failure rate distribution. For example, the lognormal distribution fits many applications in the corrosion of thin metal films. It is likely to be the best distribution model for cases in which the dominant failure mechanism relies on chemical reactions.

The errors that are created during the burning task are linked to physical-chemical processes that involve the dye. The dye is burned in order to create pit-zones and the burning occurs at *circa* 200°C, thus is a chemical-physical transformation. Therefore, when the dye is burned it is not possible to completely avoid the generation of a number of errors. The distribution of these errors follows a lognormal distribution, which generally related to physical-chemical processes. Furthermore, this lognormal distribution is also used for the analysis of failure data of CD-R [9]

For all BD-R discs brand 1 we have calculated three different cumulative error files:

1. $File_{mean}$: contains the errors calculated from the mean errors of each Data Block and for each type of disc
2. $File_{max}$: contains the errors calculated on the maximum
3. $File_{mode}$: contains the errors calculated on the mode

5.1.1.1 Mean Errors File ($File_{mean}$)

The errors mean file is calculated as:

$$File_{mean} = Data\_Block_{1\_mean}, Data\_Block_{2\_mean}, \ldots Data\_Block_{379541\_mean}$$

where:

$$DataBlock_{1mean} = mean(DataBlock_1Disc_1, DataBlock_1Disc_2, \ldots, DataBlock_1Disc_n)$$
$$DataBlock_{2mean} = mean(DataBlock_2Disc_1, DataBlock_2Disc_2, \ldots, DataBlock_2Disc_n)$$
$$DataBlock_{3mean} = mean(DataBlock_3Disc_1, DataBlock_3Disc_2, \ldots, DataBlock_3D_n)$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$
$$DB_{379541mean} = mean(DataBlock_{379541}Disc_1, DataBlock_{379541}Disc_2, \ldots, DataBlock_{379541}D_n)$$

$$mean(A) = \frac{1}{N}\sum_{i=1}^{N} A_i \,, \qquad \text{with} \quad N = \text{total number of Data Blocks; } A_i = i^{th} \text{ Data Block error}$$

In *Fig. 5.5* the block diagram to obtain the $File_{mean}$ is shown.

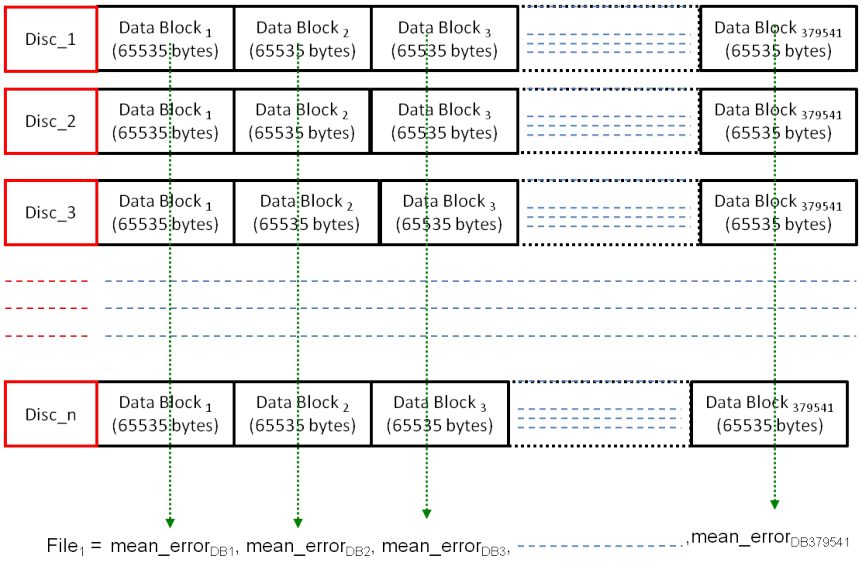


Fig. 5.5 - Blu-Ray discs 1 brand. Mean errors.

In this case the maximum value of all the mean errors is smaller than 250 (see *Fig. 5.6*).

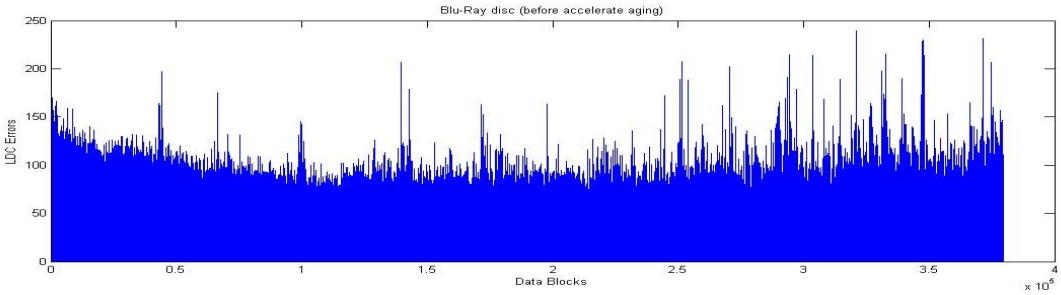


Fig. 5.6 - Blu-Ray discs brand 1. Mean errors.

In *Fig. 5.7a* the histogram of the mean errors is shown, and in *Fig. 5.7b* it is possible to notice that the best fit is again the *lognormal probability density function*.

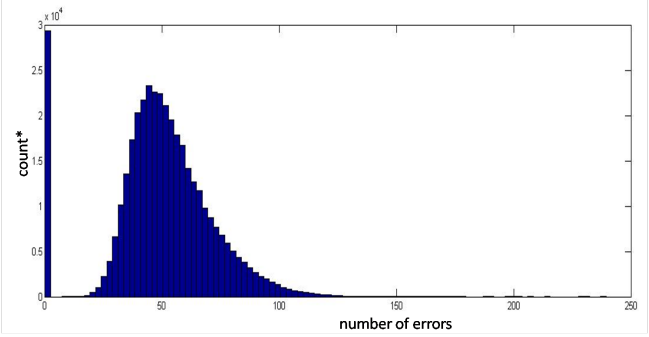


Fig. 5.7a - Blu-Ray discs brand 1. Data Blocks errors distribution.

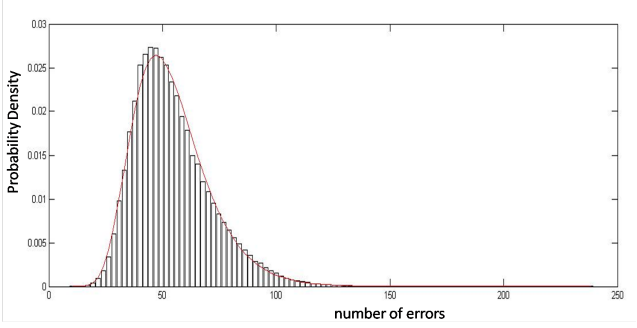


Fig. 5.7b Blu-Ray discs, brand 1. Lognormal distribution.

count=count of Data Blocks for each bin* *(without the Data Blocks with 0 errors)*

5.1.1.2 Maximum Errors File ($File_{max}$)

The maximum errors file is calculated as follows:

$$File_{max} = Data\_Block_{1\_max}, Data\_Block_{2\_max}, \ldots Data\_Block_{379541\_max}$$

where:

$$DataBlock_{1max} = max(DataBlock_1Disc_1, DataBlock_1Disc_2, \ldots, DataBlock_1Disc_n)$$

$$DataBlock_{2max} = max(DataBlock_2Disc_1, DataBlock_2Disc_2, \ldots, DataBlock_2Disc_n)$$

$$DataBlock_{3max} = max(DataBlock_3Disc_1, DataBlock_3Disc_2, \ldots, DataBlock_3Disc_n)$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$DB_{379541max} = max(DataBlock_{379541}Disc_1, DataBlock_{379541}Disc_2, \ldots, DataBlock_{379541}Disc_n)$$

In *Fig. 5.8* the block diagram to obtain the $File_{max}$ is shown.

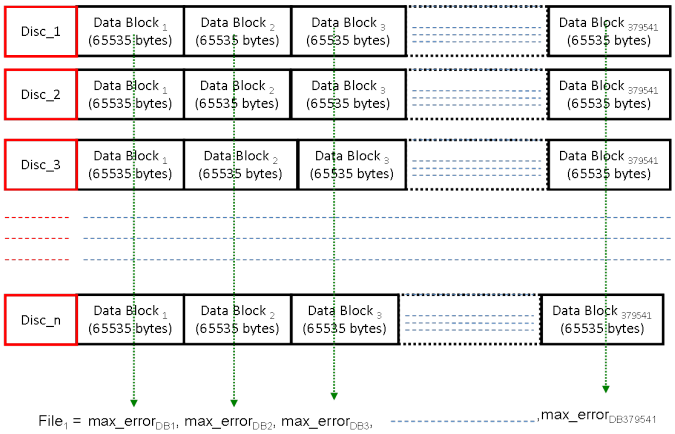


Fig. 5.8 - Blu-Ray discs 1 brand. Maximum errors.

In this case the maximum number of errors is smaller than 600 (see *Fig. 5.9*).

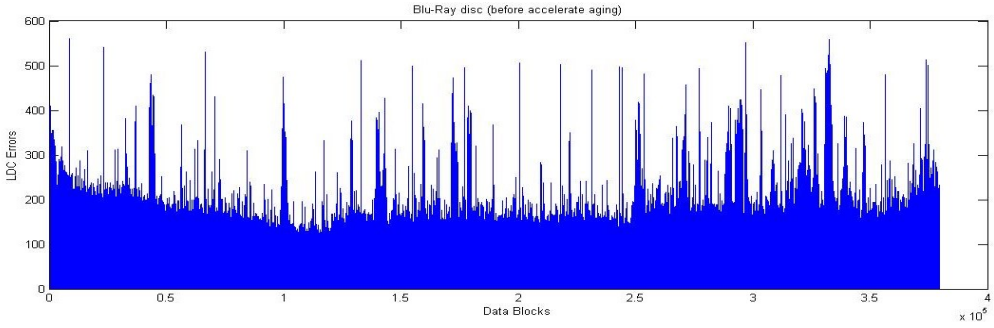


Fig. 5.9 – Blu-Ray discs, brand 1. Errors calculated on the maximum value.

In *Fig. 5.10a* the histogram of the maximum errors file and in *Fig. 5.10b* the best fit is again the *lognormal probability density function* are shown.

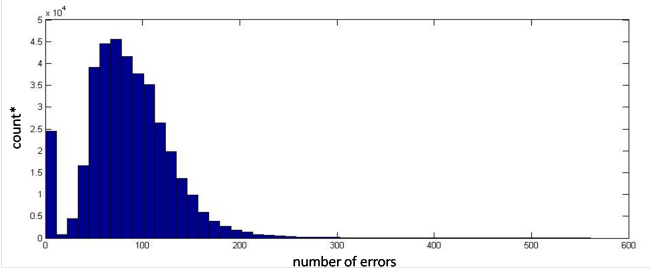


Fig. 5.10a - Data Blocks errors distribution..
count=count of Data Blocks for each bin.*

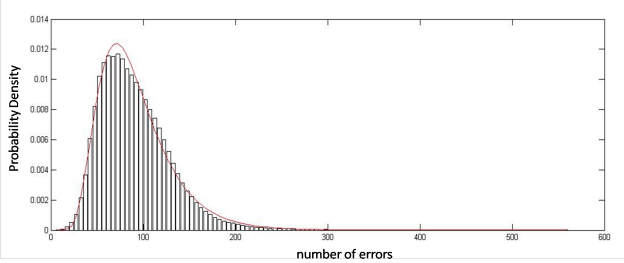


Fig. 5.10b - Lognormal distribution.
(without the Data Blocks with 0 errors)

5.1.1.3 Mode Errors File (*File*$_{mode}$)

The mode errors file is calculated as

:

$File_{mode} = Data\_Block_{1\_mode}, Data\_Block_{2\_mode}, \ldots Data\_Block_{379541\_mode}$

where:

$DataBlock_{1mode} = mode(DataBlock_1Disc_1, DataBlock_1Disc_2, \ldots, DataBlock_1Disc_n)$

$DataBlock_{2mode} = mode(DataBlock_2Disc_1, DataBlock_2Disc_2, \ldots, DataBlock_2Disc_n)$

$DataBlock_{3mode} = mode(DataBlock_3Disc_1, DataBlock_3Disc_2, \ldots, DataBlock_3Disc_n)$

$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$

$DB_{379541mode} = mode(DataBlock_{379541}Disc_1, DataBlock_{379541}Disc_2, \ldots, DataBlock_{379541}Disc_n)$

$$mode = l + \left(\frac{fs}{f_p + f_s} \times c \right)$$

with:

l = lower limit of the modal class

fs = the frequency of the class succeeding the modal class

fp = the frequency of the class preceding the modal class

c = width of the class interval

In *Fig. 5.11* the block diagram used to obtain the $File_{mode}$ is shown.

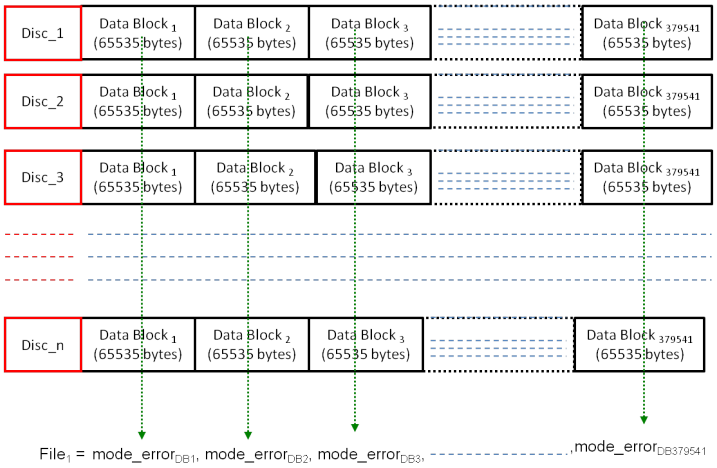


Fig. 5.11 - Blu-Ray discs 1 brand. Errors calculated on the mode.

The maximum errors in the plot $File_{mode}$ is smaller than 250 (see *Fig. 5.12*).

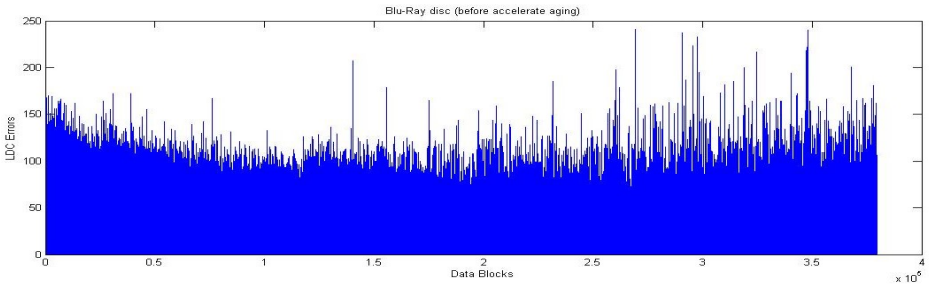


Fig. 5.12 – Blu-Ray discs, brand 1. Mode errors.

In *Fig. 5.13a* the histogram of the mode errors file and in *Fig. 5.13b* the best fit is the *lognormal probability density function* are shown.

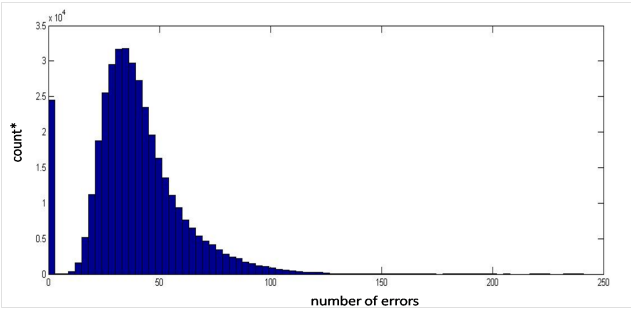


Fig. 5.13a - Data Blocks errors distribution..
count=count of Data Blocks for each bin.*

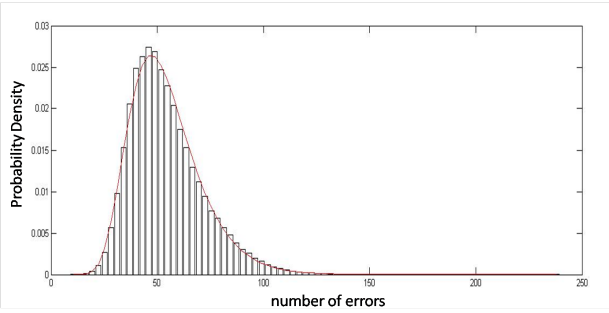


Fig. 5.13b - Lognormal distribution.
(without the data Block with 0 errors)

5.1.2 After Accelerate Aging

In *Fig. 5.14* the amount of errors in the Blu-Ray disc 1 of brand 1 are illustrated. Up to the middle of the disc the errors are uniformly distributed and in the remaining part they increase exponentially.

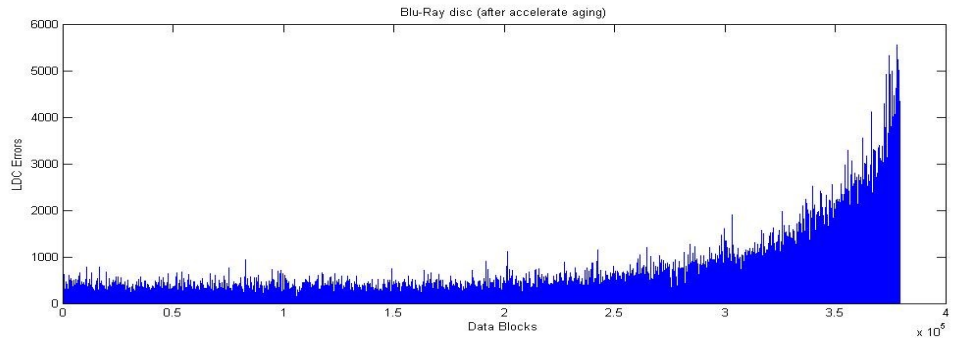


Fig. 5.14 Blu-Ray disc1 brand 1. Data Block errors after accelerate aging.

In *Fig. 5.15* the experimental data are adequately reproduced by the exponential function [68]

$y = A \cdot e^{kx}$ where A = value at the start and k is a constant.

The exponential function is used to describe several physical phenomena like the radioactive decay and the avalanche effects inside a dielectric material [69].

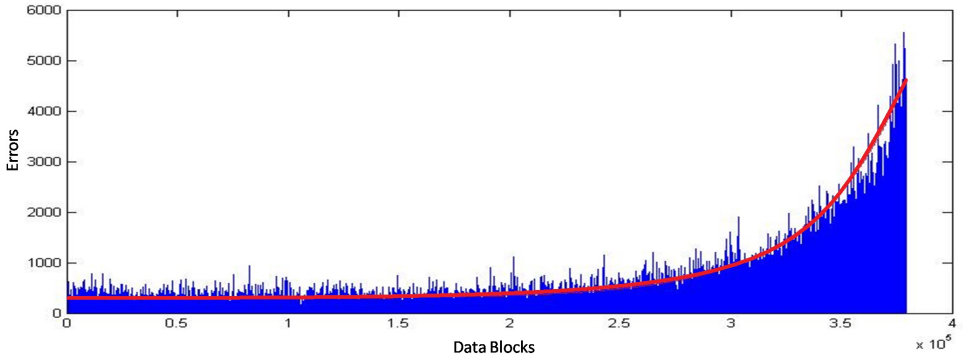


Fig. 5.15 Blu-Ray disc1 brand 1- after accelerate aging - fit of errors.

In *Fig. 5.16a* the histogram of the errors is shown. In *Fig. 5.16b* the histogram of the errors without the Data Blocks with errors equal to zero is shown.

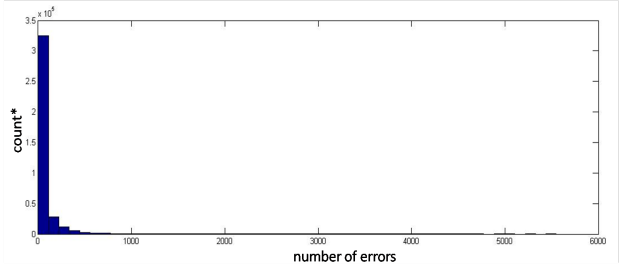


Fig. 5.16a Blu-Ray disc1 brand 1- histogram. count=count of Data Blocks for each bin.*

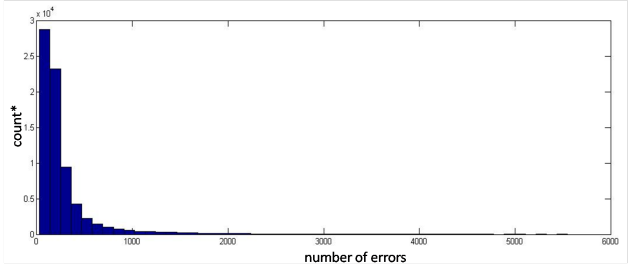


Fig. 5.16b 14a Blu-Ray disc1 brand 1- histogram. (without the data Block = 0 errors)

In *Fig. 5.17* the best fit is no more the *lognormal function distribution* but the *generalized extreme value distribution* [70].

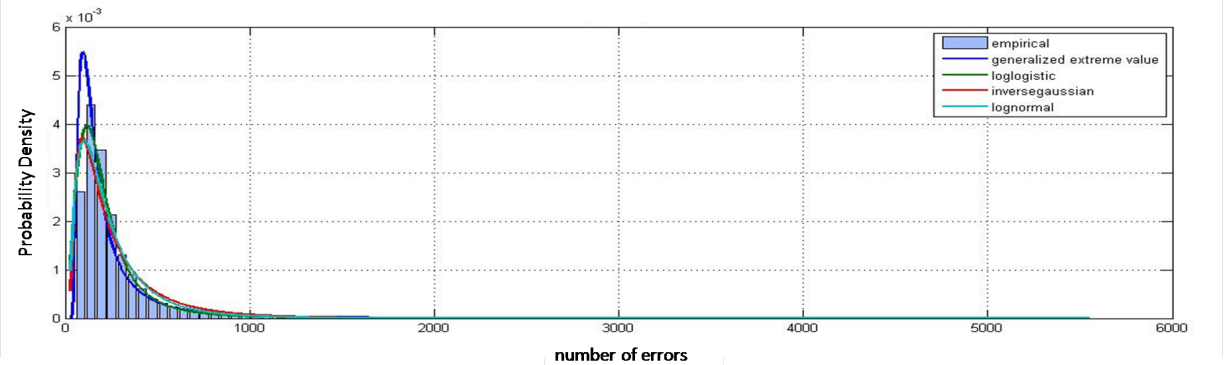


Fig. 5.17 Blu-Ray disc, brand 1- fit generalized extreme distribution function.

The extreme value distribution is appropriate for modeling the smallest value from a distribution whose tails decays exponentially fast. It can also model the largest value from a distribution, such as the normal or exponential distributions, by using the negative of the original values.
In *Fig. 5.18* the errors of the Blu-Ray disc of brand 1 before and after accelerate aging are shown.

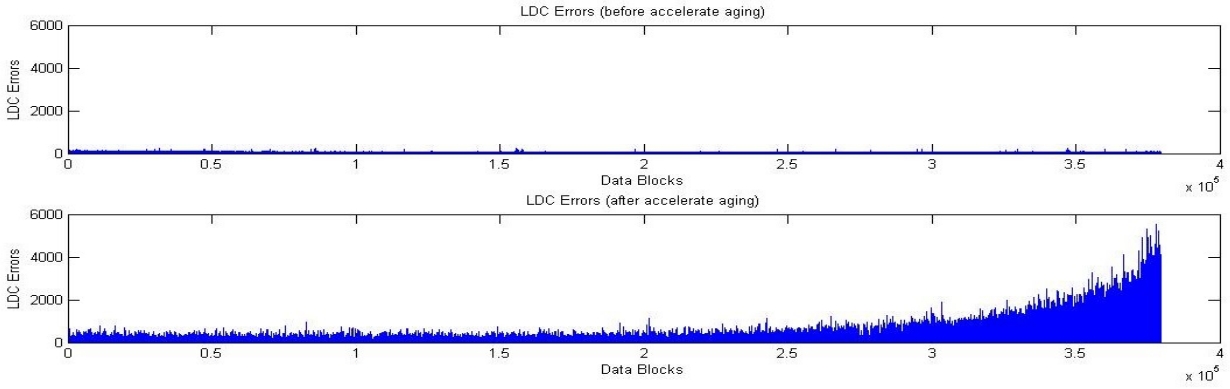


Fig. 5.18 – Blu-Ray disc1 brand 1- errors before and after accelerate aging.

In *Table 5.2* a comparison of some statistical parameters of the plots in Fig. 5.18 is reported.

	max error	mean	Standard Deviation
before accelerate aging	217	39.3	11.3
after accelerate aging	5550	278.5	365.2

Tabella 5.2 – comparison of selected statistical parameters. The maximum number of errors, the mean number of errors and the corresponding standard deviations are reported, before and after the aging processes.

The maximum error before aging, equal to 217, increases after aging to 5550. This last value indicates that the disc has reached the end life.

The mean of the errors before aging, 39.3, becomes 278.5 and the standard deviation [71] increase from 11.3 to 365.2.

Standard deviation: $S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} | A_i - \mu |^2}$

The standard deviation indicates that before aging the error dispersion is very low (the error distribution is uniform) but notably increases after aging.

For all BD-R discs brand 1 we have calculated three cumulative error files:

4. $File_{mean}$: it contains the errors calculated on the mean
5. $File_{max}$: it contains the errors calculated on the maximum
6. $File_{mode}$: it contains the errors calculated on the mode

Since the procedure to calculate this three files has been detailed in the previous paragraphs, I will discuss only the results.

5.1.2.1 Mean Errors File ($File_{mean}$)

In this case the errors mean is bigger than 5000 (see *Fig. 5.19*).

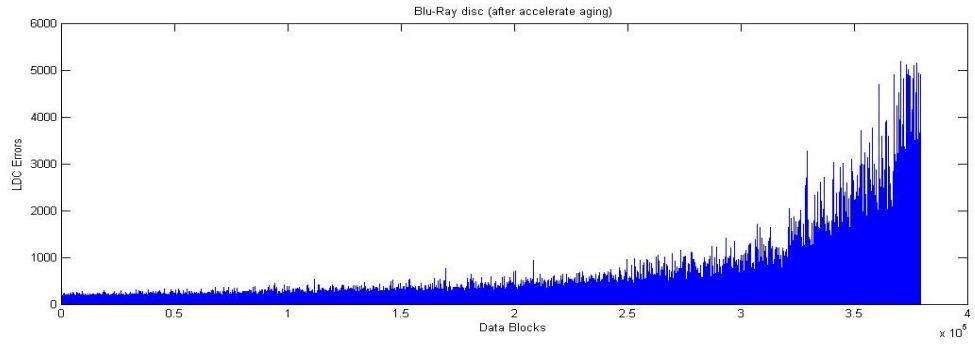


Fig. 5.19 Blu-Ray discs brand 1 - errors calculated on the mean.

In *Fig. 5.20a* the histogram of the errors and in *Fig. 5.20b* the histogram of the errors without the Data Blocks errors equal to zero are shown

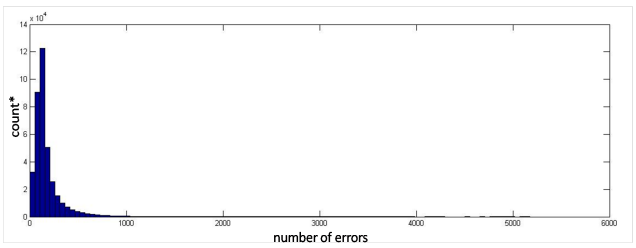


Fig. 5.20a Blu-Ray brand 1- histogram. count=count of Data Blocks for each bin.*

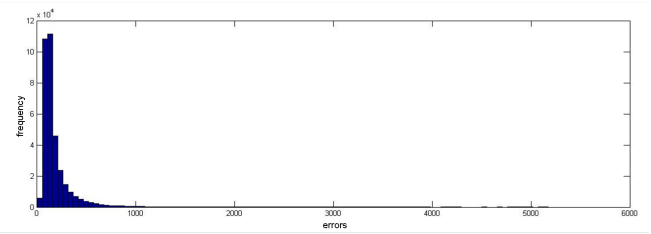


Fig. 5.20b Blu-Ray disc1 brand 1- histogram. (without the data Block = 0 errors)

In *Fig. 5.21* we observe that the best fit is again the *generalized extreme value distribution*.

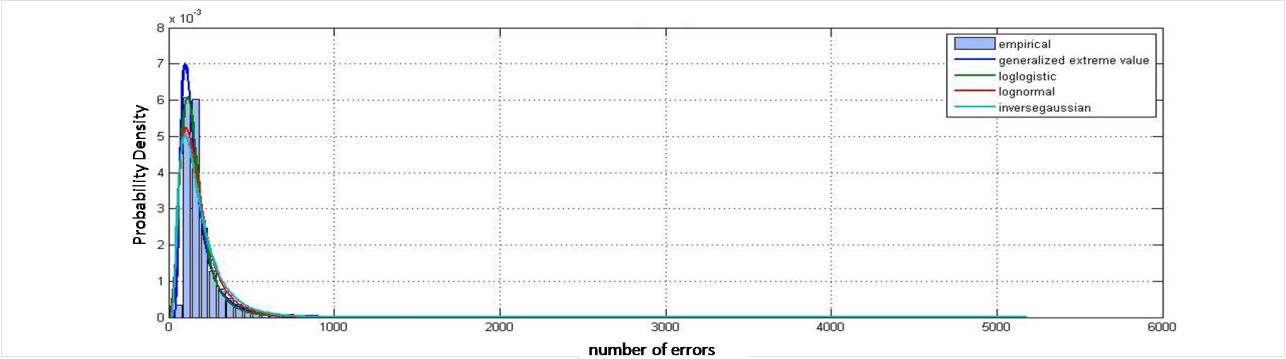


Fig. 5.21 - Blu-Ray discs, brand 1- fit generalized extreme distribution function..

In *Fig 5.22* the raw errors of Blu-Ray disc brand 1 before and after accelerate aging is shown.

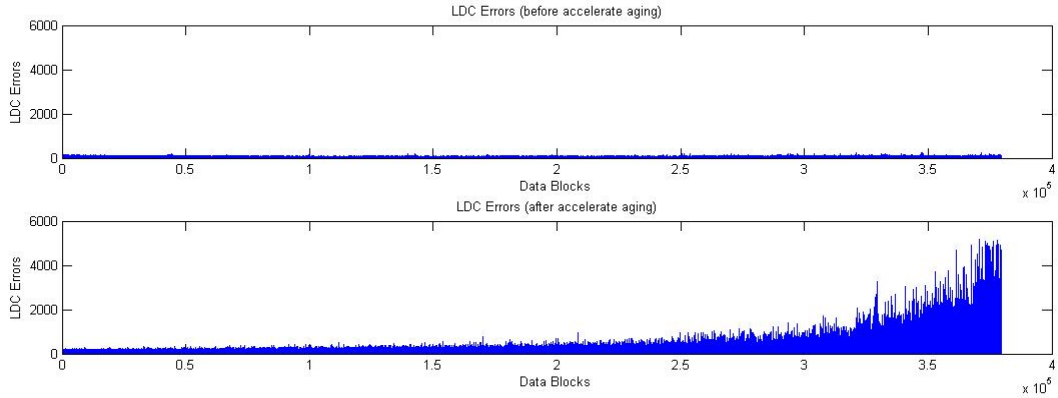


Figura 5.22 - Blu-Ray discs brand 1 - errors before and after accelerate aging.

In Table 5.3 there are some statistical parameters calculated on the two plots

	max errors	mean	Standard Deviation
before accelerate aging	200	50.1115	21.9431
after accelerate aging	5144	177.1787	202.1842

Table 5.3- Selected statistical parameters calculated from the two plots.

5.1.2.2 Maximum Errors File (*File$_{max}$*)

In *Fig. 5.23* the plot calculated on the maximum errors is reported. The maximum errors is 10000.

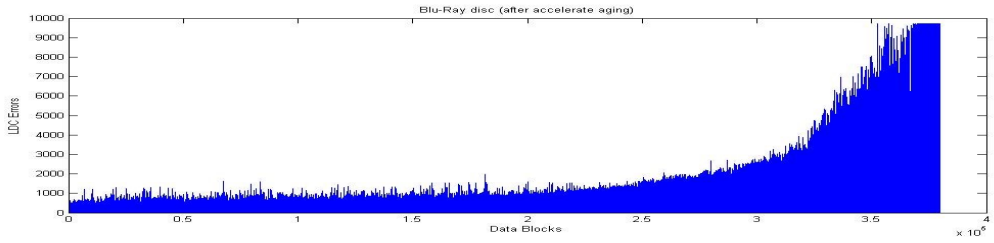


Fig. 5.23 - Blu-Ray discs, brand 1 - errors calculated on the maximum errors.

In *Fig. 5.24a* the histogram of the errors and in *Fig. 5.24b* the histogram of the errors without the Data Blocks errors equal to zero.

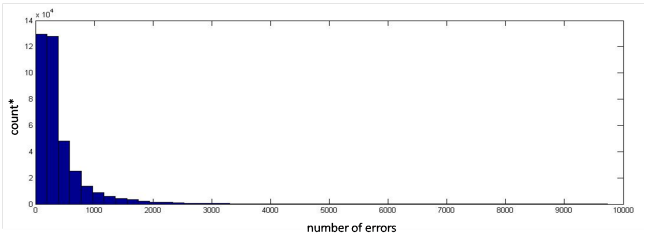


Fig. 5.24a Blu-Ray brand 1- histogram.
count=count of Data Blocks for each bin.*

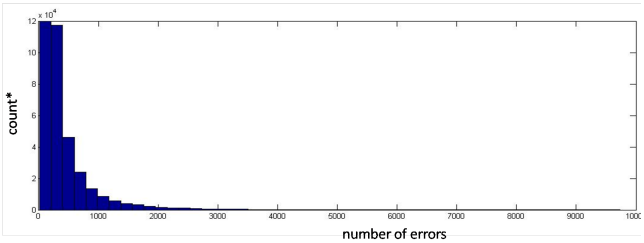


Fig. 4.24b Blu-Ray brand 1- histogram.
(without the data Block = 0 errors)

In *Fig. 5.25* we observe that the best fit is again the *generalized extreme value distribution*.

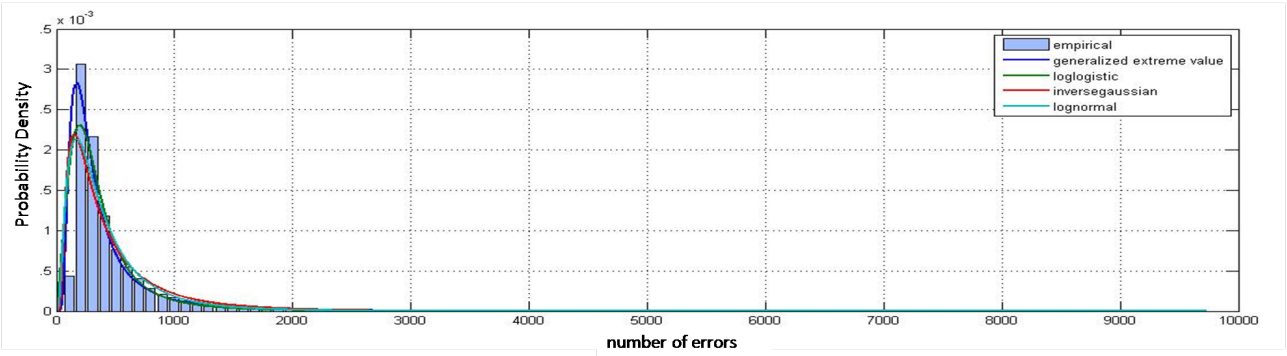


Fig. 5.25 Blu-Ray discs, brand 1- fit generalized extreme distribution function.

In *Fig. 5.26* the errors of Blu-Ray disc brand 1 before and after accelerate aging.

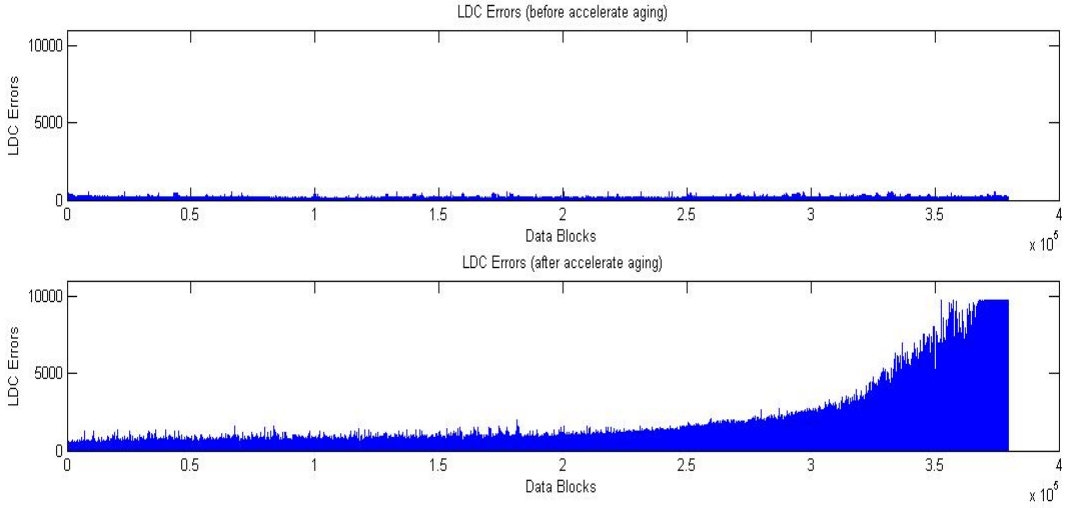


Fig.5.26 - Blu-Ray discs, brand 1 - errors before and after accelerate aging.

In *Table 5.4* there are some statistical parameters calculated on the two plots

	mean errors	*mean*	*Standard Deviation*
before accelerate aging	*423*	*85.8732*	*44.5075*
after accelerate aging	*9738*	*455.5537*	*710.1434*

Table 5.4- some statistical parameters calculated on the two plots.

5.1.2.3 Mode Errors File (*File$_{mode}$*)

In *Fig. 5.27* the plot calculated on the maximum errors is shown. The maximum errors is 10000.

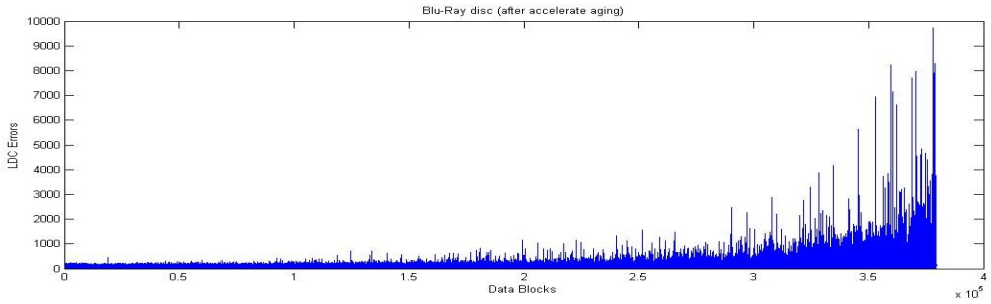


Fig. 5.27 - Blu-Ray discs, brand 1 - errors calculated on the mode.

In *Fig. 5.28a* the histogram of the errors and in *Fig. 5.28b* the histogram of the errors without the Data Blocks with errors equal to zero.

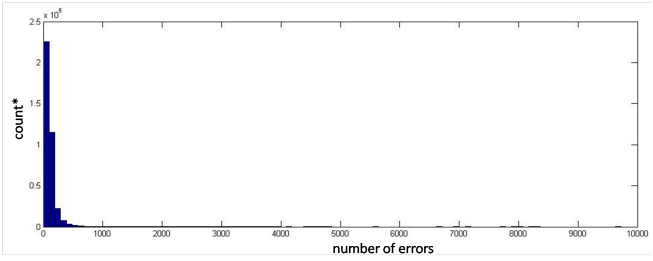


Fig. 5.28a Blu-Ray discs, brand 1- histogram. count=count of Data Blocks for each bin.*

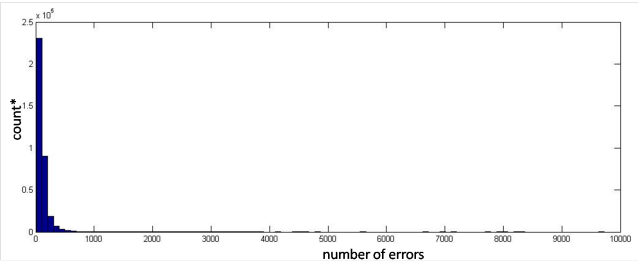


Fig. 5.28b Blu-Ray discs, brand 1- histogram. (without the Data Block = 0 errors)

In *Fig. 5.29* we observe that the best fit is again the *generalized extreme value distribution*.

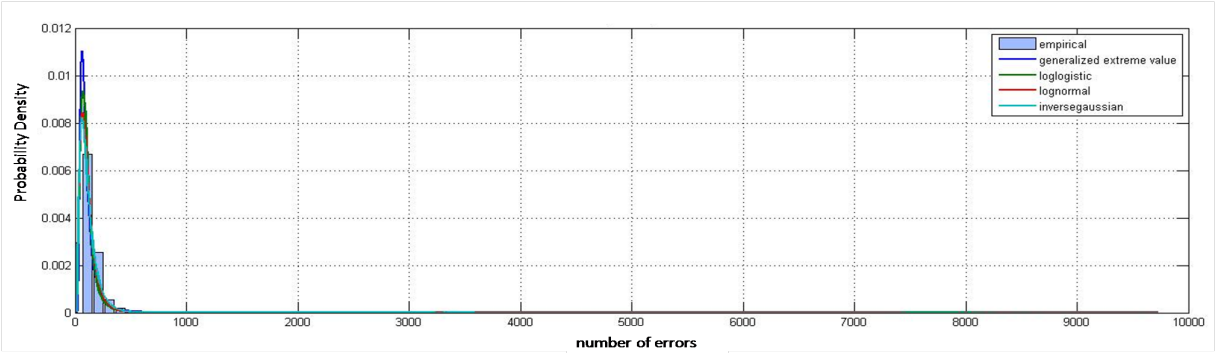


Fig. 5.29 Blu-Ray discs brand 1- fit generalized extreme distribution function.

In *Fig. 5.30* the errors of Blu-Ray disc brand 1 before and after accelerate aging.

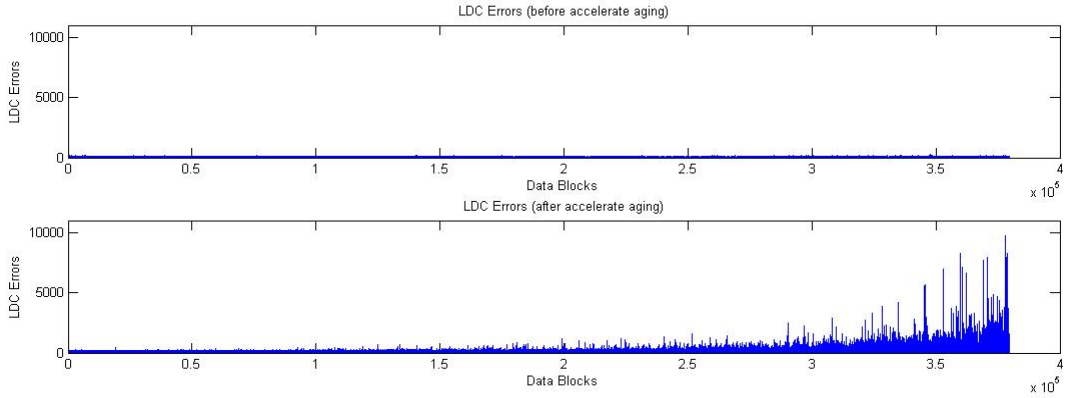


Fig. 5.30 - Blu-Ray discs, brand 1 - errors before and after accelerate aging.

In *Table 5.4* there are some statistical parameters calculated on the two plots

	mean errors	mean	Standard Deviation
before accelerate aging	289	39.8255	20.2978
after accelerate aging	9000	110.2054	126.7373

Table 5.4- some statistical parameters calculated on the two plots.

For BDs brand 1, the analysis performed on the data has highlighted the presence of degradation phenomena (that, as shown in the previous sections, are linked to the nature of the dye). These phenomena affect the life expectancies of optical discs, if the standard correction code is used. The fact that errors are concentrated in specific areas of the discs will be the starting point for the development of a new, adaptive correction code system. With this adaptive code it will be possible to overcome these degradation effects and to improve the life expectancies of optical discs.

5.2 Blu-Ray - brand 2

After the accelerate aging, 36 out of 48 BD-R discs brand 2 became so damaged that they could not be analyzed. The defect in the polycarbonate layer is clearly visible in Fig. 5.31.

Unfortunately, also the remaining discs were partially damaged. In the damaged zone high error peaks are observed. On some disc the peak errors already present before the aging process slightly increase after the aging (see *Fig. 5.32*).

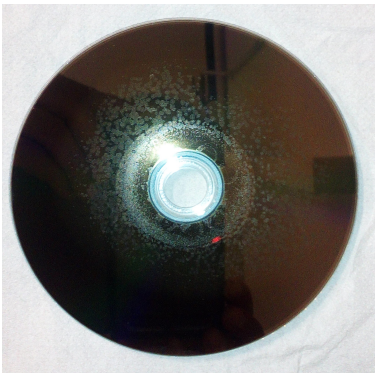


Fig. 5.31 - Damaged Blu-Ray disc brand 2

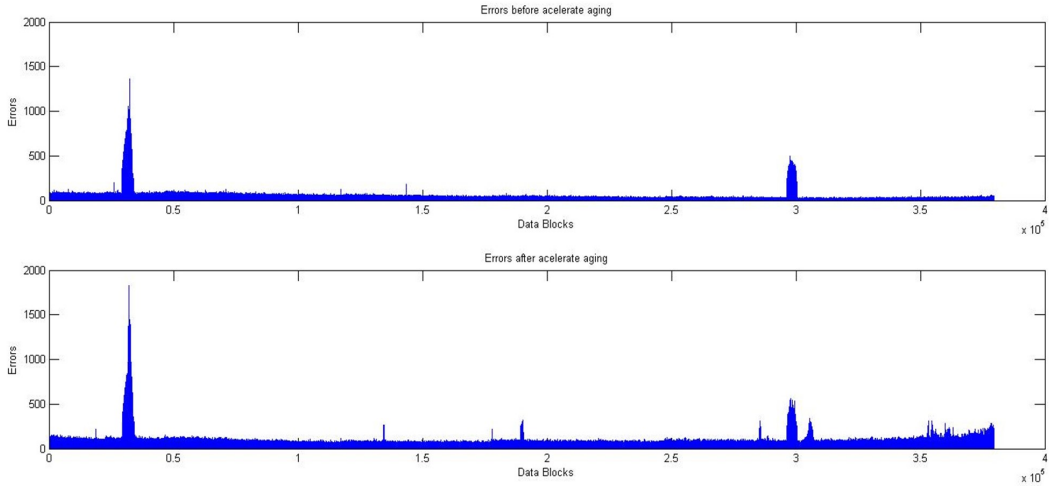


Fig. 5.32 Blu-Ray disc, brand 2 - errors before and after accelerate aging.

Differently, for some discs the high error peaks appear only after the aging (*see Fig. 5.33*).

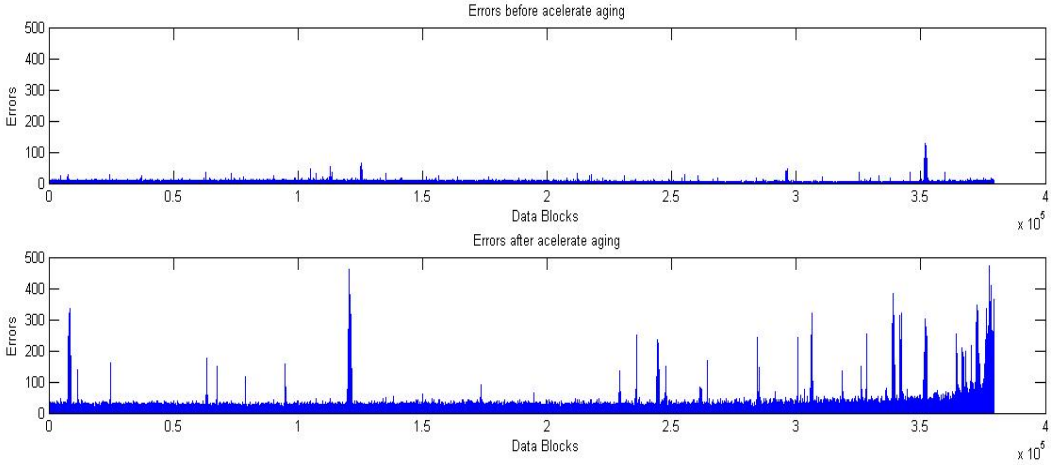


Fig. 5.33 Blu-Ray disc, brand 2- errors before and after accelerate aging.

Since any disc of different brand presented the same behaviour we believe that the BD-R discs of brand 2 were made with defective material.

Both *Figs. 5.32 - 5.33*, evidence that the errors outside the anomalous peaks have a distribution similar to that observed for BD-R discs brand 1.

5.3 CD-R - brand 3

5.3.1 Before accelerate aging.

The plots of the analysis for the CD-R discs brand 3 are illustrated below.

5. 3.1.1 Mean Errors File (*File*$_{mean}$)

In this case the maximum value of all the mean errors is smaller than 12 (see *Fig. 5.34*).

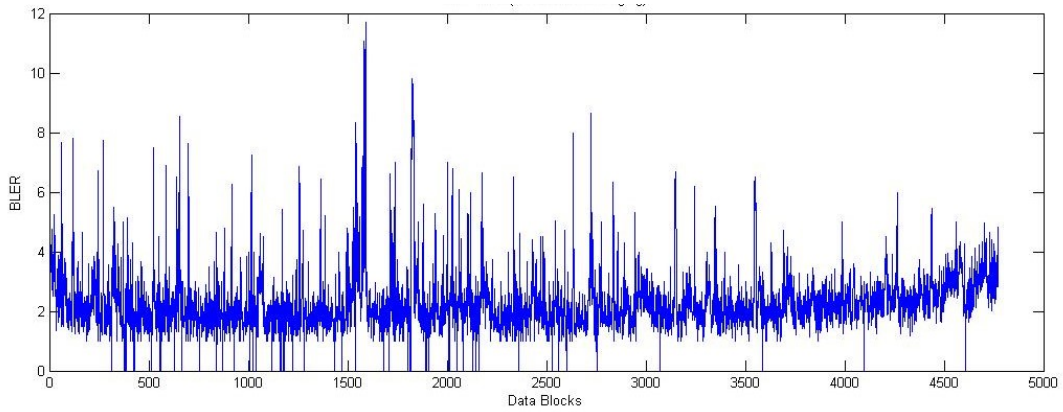


Fig. 5.34 CD-R discs brand 1 – errors calculate on the mean before accelerate aging.

In *Fig. 5.35a* the histogram of the mean errors is shown, and in *Fig. 5.35b* it is possible to notice that the best fit is again the *lognormal probability density function*.

.

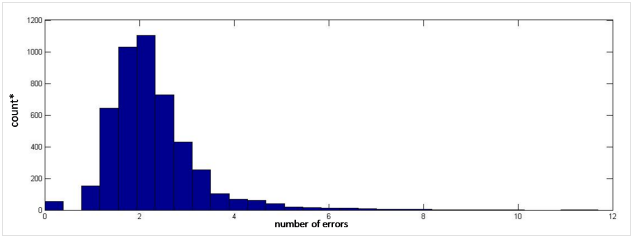


Fig. 5.35a CD-R discs brand 3 - Data Blocks errors distribution.

count=count of Data Blocks for each bin.*

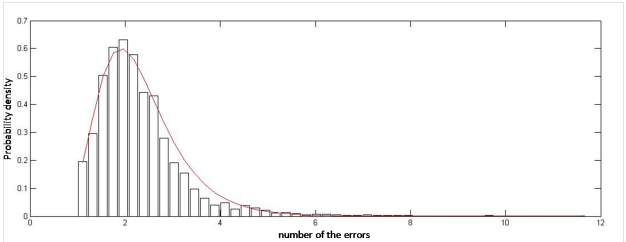


Fig. 5.35b CD-R discs brand 3 – lognormal distribution.

(without the Data Blocks=0)

5.3.1.2 Maximum Errors File (*File*$_{max}$)

The maximum value of all the maximum errors is smaller than 25 (see *Fig. 5.36*).

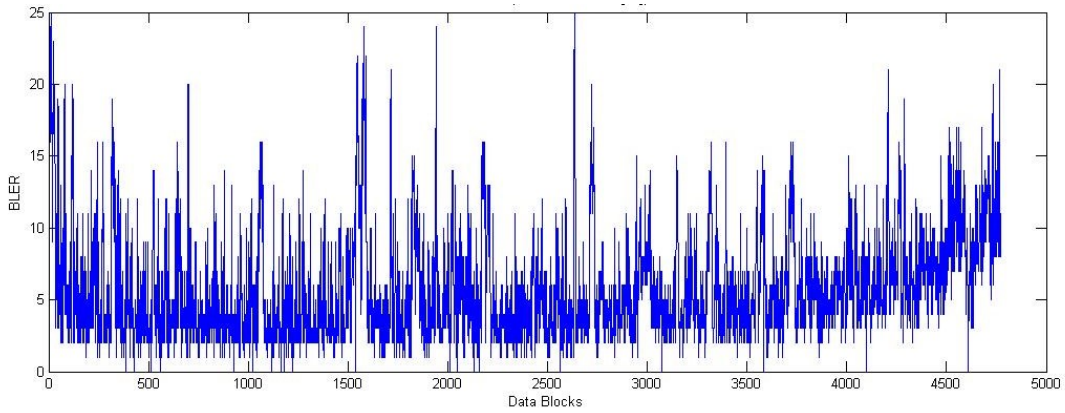


Fig. 5.36 CD-R discs brand 1 – errors calculate on the maximum errors - before accelerate aging.

In *Fig. 5.37a* the histogram of the mean errors is shown, and in *Fig. 5.37b* it is possible to notice that the best fit is again the *lognormal probability density function*.

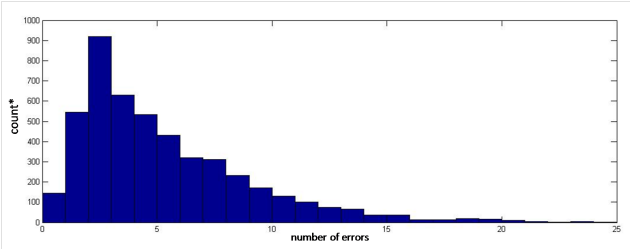


Fig. 5.37a CD-R discs brand 3 - Data Blocks errors distribution.
count=count of Data Blocks for each bin.*

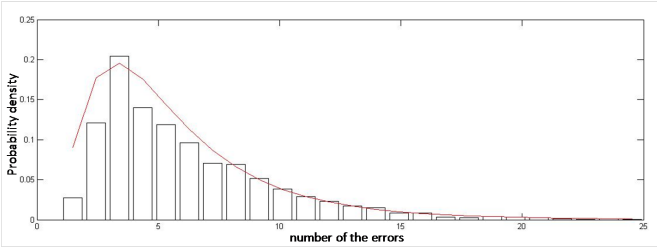


Fig. 5.37b CD-R discs brand 3 – lognormal distribution.
(withot the Data Blocks=0)

5. 3.1.3 Mode Errors File (*File*$_{mode}$)

The maximum value of all the maximum errors is smaller than 12 (see *Fig. 5.38*).

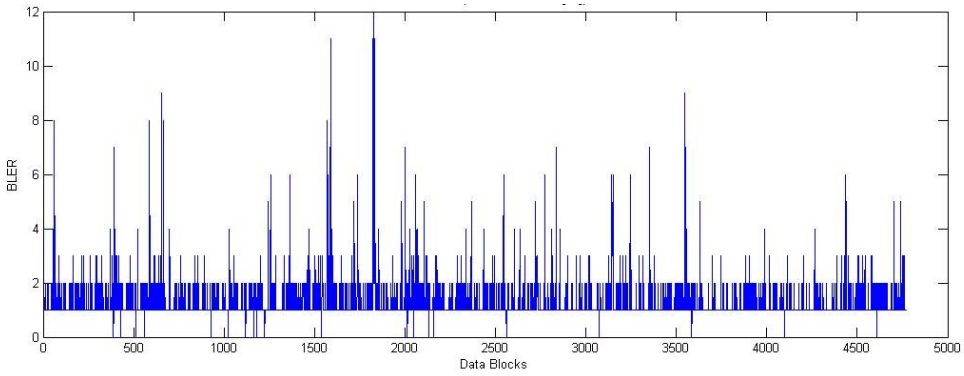


Fig. 5.38 CD-R discs brand 1 – errors calculate on the mode errors - before accelerate aging.

In *Fig. 5.39a* and *Fig. 5.39b* the histograms of the mode errors is shown.

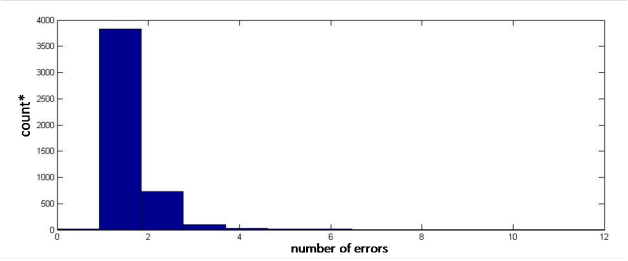


Fig. 5.39a CD-R brand 3 - Data Blocks errors distribution.
count=count of Data Blocks for each bin.*

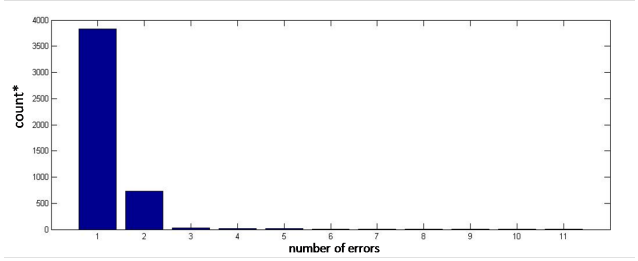


Fig. 5.39a CD-R brand 3 - Data Blocks errors distribution.
(withot the Data Blocks=0)

5.3.2 After accelerate aging

The plots of the analysis for the CD-R discs brand 3 are illustrated below.

5.3.2.1 Mean Error File (*File*$_{mean}$)

The maximum value of all the mean errors is smaller than 40 (see *Fig. 5.40*).

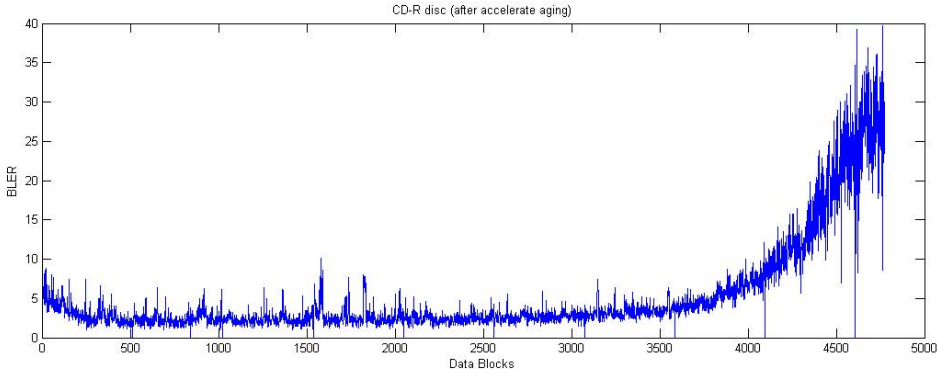


Fig. 5.40 CD-R discs brand 3 – errors calculate on the mean after accelerate aging.

In *Fig. 5.41* the best fit is *generalized extreme value distribution* .

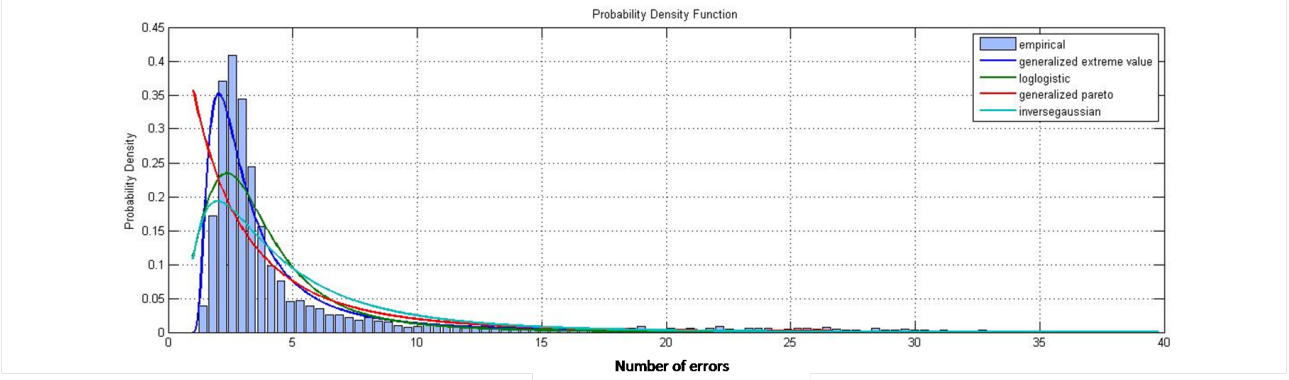


Fig. 5.41 CD-R discs, brand 3 – fit generalized extreme distribution function.

In *Fig. 5.42* the errors of the CD-R discs of brand 3 before and after accelerate aging are shown.

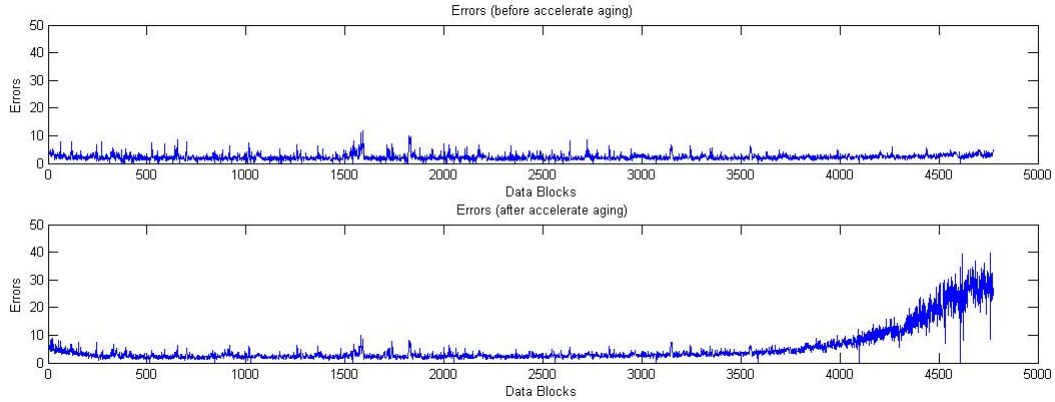


Fig. 5.42 – CD-R discs brand 3 - errors before and after accelerate aging.

5.3.2.2 Maximum Errors File (*File$_{max}$*)

The maximum value of all the maximum errors is smaller than 800 (see *Fig. 5.43*).

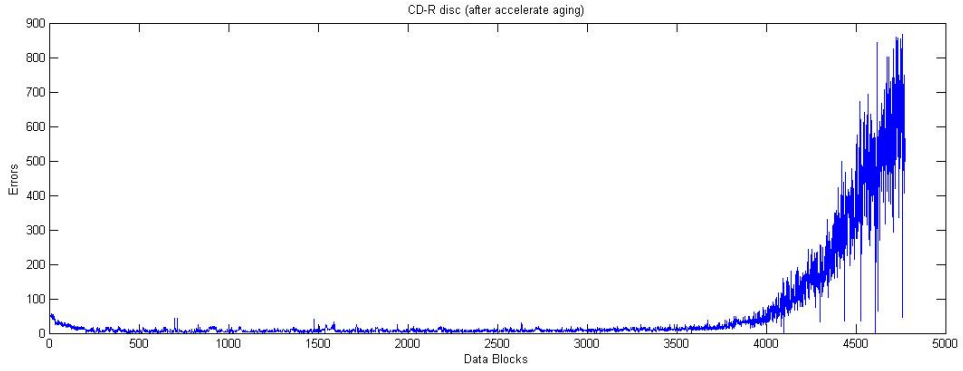


Fig. 5.42 CD-R discs, brand 3 – errors calculate on the maximum value - after accelerate aging.

In *Fig. 5.44* the histogram of the mode errors is shown.

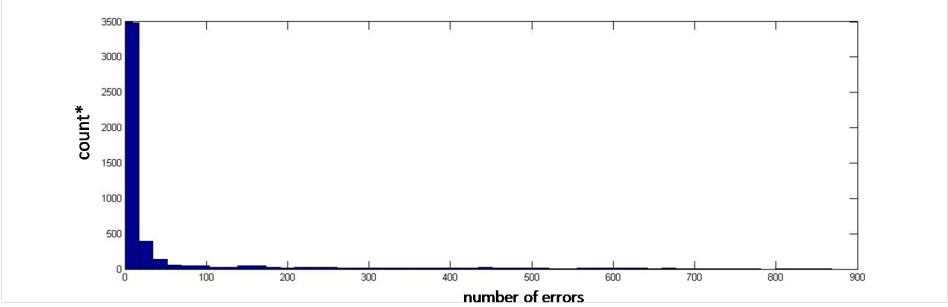


Fig. 5.44 CD-R discs brand 3 - Data Blocks errors distribution.

count=count of Data Blocks for each bin.*

In *Fig. 5.45* the best fit is *generalized extreme value distribution.*

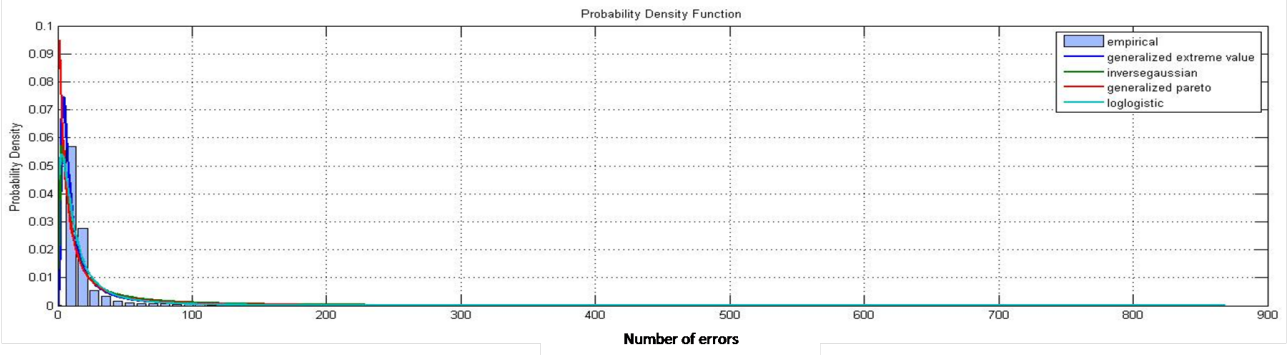


Fig. 5.45 CD-R discs, brand 3 – fit generalized extreme distribution function.

In *Fig.5.46* the errors of the CD-R discs of brand 3 before and after accelerate aging are shown.

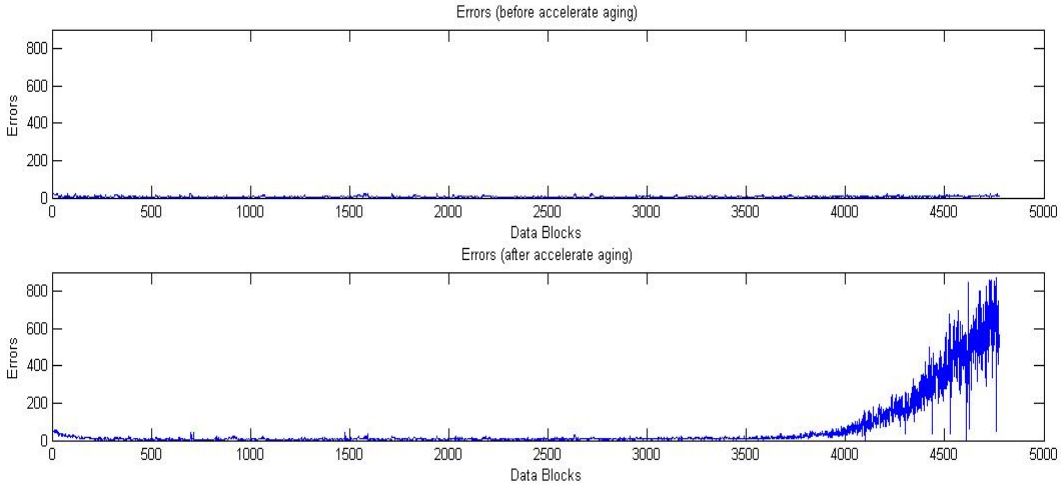


Fig. 5.46 – CD-R discs brand 3 - errors before and after accelerate aging.

The analysis that were performed on the CD-R confirm the concentration of degradation phenomena in the outer areas of the discs. Therefore it is possible to affirm that the degradation processes are comparable for all the types and all brands of optical discs – DVDs, BDs, CD-R. Also for this type of support it would be possible to contrast the degradation by implementing a new, adaptive correction code system.

CHAPTER 6

6. How to implement an Adaptive RS code: computing the degradation function

In this chapter we will discuss how to provide a proper fitting of the experimental data sets obtained in Chapter 5, where we computed the errors degradation function. In the remainder, we will refer to the functions fitting the data as *degradation functions*. By using this functions, and also considering the discussion in Chapter 4, it will be possible to generate, for each type of disc, a *parity file*. The A-RS code will use this file before the burning process to counteract the degradation effects on the information data.

We will discuss three types of degradation function computed on the mean ($File_{mean}$), local maximum ($File_{max}$), and the mode ($File_{mode}$) of the error data sets.

6.1 Blu-Ray discs – brand 1

6.1.1 Degradation function calculated from the $File_{mean}$

From two plots (see *Fig. 6.1*), calculated on the mean value of the errors (before and after the accelerated aging), it is possible to calculate the increment of the degradation, from the different number of errors before and after the aging process, as follows.

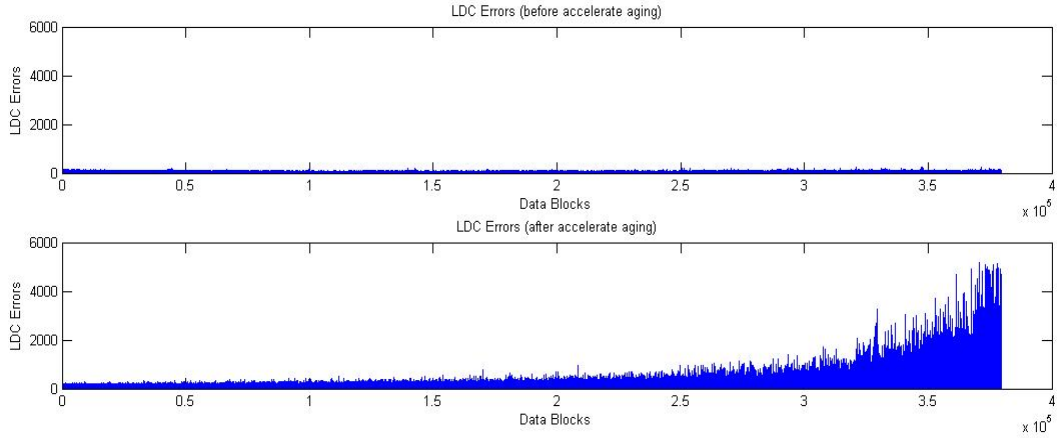


Fig. 6.1 - Blu-Ray brand 1 – Mean errors before (up) and after (down) the accelerated aging process.

The degradation increment is given:

$\Delta = \sum_{i=1}^{n} (EA_i - EB_i)$, where EA=errors after aging and EB= errors before aging

The fit [72, 73] of Δ is the polynomial function of degree 6 (see *Fig. 6.2*) that follows:

$p(x) = p_1 x^6 + p_2 x^5 + p_3 x^4 + p_4 x^3 + p_5 x^2 + p_6$

In this case I have used a polynomial function instead of an exponential function (that had been employed in the previous example regarding a single Blu-Ray optical disc) in order to avoid unnecessary complications. In fact, polynomial functions are more adaptable to various data sets than exponential functions, and their application is more straightforward.

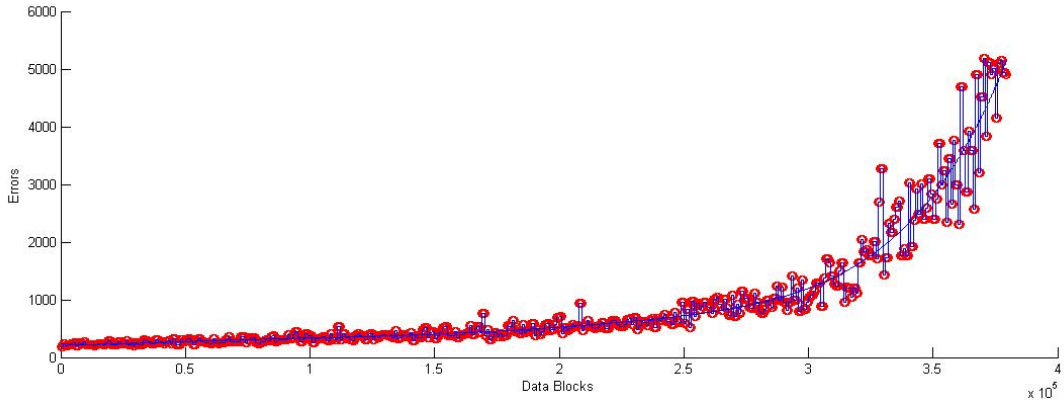


Fig. 6.2 – Blu-Ray discs brand 1 – the best polynomial fit.

6.1.1.1 Fit Tests

To verify that the polynomial of degree 6 is the best choice we will use two methods:

- Residuals Sum of Squared (RSS)
- Coefficient of determination (R^2)

The Residuals Sum of Squared (SSR) [74] is the sum of the squares of residuals (deviations predicted from actual empirical values of data). It is a measure of the discrepancy between the data and an estimation model. A small SSR indicates a tight fit of the model to the data. It is defined as:

$$SSR = \sum_{i=1}^{N} [p_i - y_i]^2$$

where:

p_i = predicted values

y_i = data values

Calculating SSR for the three polynomials of degree 5, 6 and 7, we have that SSR_5= 2.1668, SSR_6=2.1447, SSR_7=2.1443.

Considering polynomial functions, the optimal value of SSR is obtained by polynomial of the 6 because $SSR_6 < SSR_5$, and because the polynomial function of the 7 would not result in a significant reduction of SSR.

The coefficient of determination (R^2) [75] is a statistical measure of how close the data are to the fitted regression line. It indicates the proportionate amount of variation in the dependent variable that is predictable from the independent one. The more the value R^2 approaches 1, the more the prediction is accurate. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}[p_i - y_i]^2}{\sum_{i=1}^{N}\left[y_i - \bar{y}\right]^2}$$

where:

p_i = predicted values

y_i = data values

$\bar{y}$ = mean value

Calculating R^2 for the three polynomials of degree 5, 6 e 7 we have that $R^2(5)=0.9481$, $R^2(6)=0.9487$, $R^2(7)=0.9487$.

Also in this case, the best value of SSR is obtained by polynomial of the 6 .

6.1.2 Degradation function calculated from the File$_{max}$

From two plots (see *Fig. 6.3*), calculated on the mean value of the errors (before and after the accelerated aging), now we calculate the degradation increment.

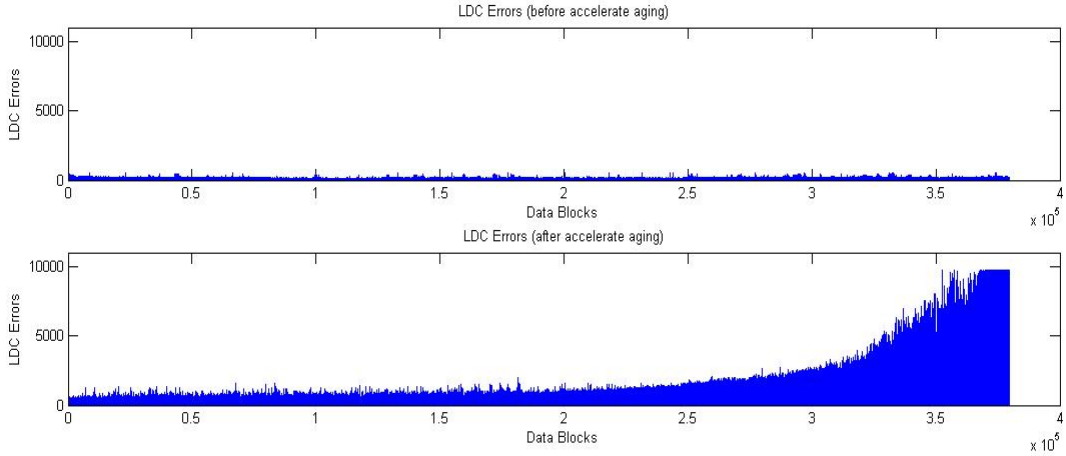


Fig. 6.3- Blu-Ray discs, brand 1 - errors calculated on the maximum errors.

The degradation increment is given as follows:

$\Delta = \sum_{i=1}^{n}(EA_i - EB_i)$, where: *EA*=errors after aging, and: *EB*=errors before aging

The fit of Δ is the polynomial function of degree 6 (see *Fig. 6.4*).

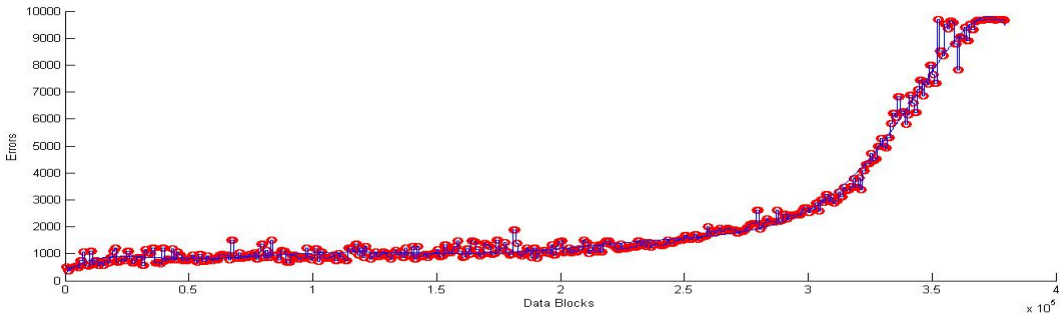


Fig. 6.4 – Blu-Ray discs brand 1 – the best polynomial fit.

6.1.2.1 Fit Tests

To verify that the polynomial of degree 6 is the best choice we will use two methods:

- Residuals Sum of Squared (RSS)
- Coefficient of determination (R^2)

Calculating SSR for the three polynomials of degree 7, 8 and 9, we have that SSR_7= 3.1202, SSR_8=2.3934, SSR_9=2.3934.

In this case the best value of SSR is obtained by polynomial of the 8° because $SSR_8 < SSR_7$, and because the polynomial function of the 9 would not result in a significant reduction of SSR.

Calculating R^2 for the three polynomials of degree 5, 6 e 7 we have that $R^2(7)$= 0.9863, $R^2(8)$=0.9895, $R^2(9)$=0.9895.

In this case again, the best value of SSR is obtained by polynomial of the 8 degree.

6.1.3 Degradation Function Calculated from the $File_{mode}$

From two plots (see *Fig. 6.5*), calculated on the mode value of the errors (before and after the accelerated aging), now we calculate the degradation increment.

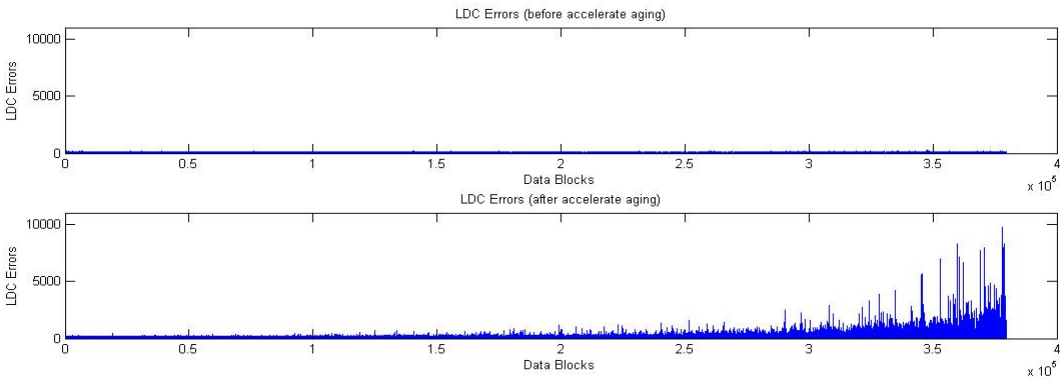


Fig. 6.5 - Blu-Ray discs, brand 1 - errors before and after accelerate aging..

The degradation increment is given:

$\Delta = \sum_{i=1}^{n} (EA_i - EB_i)$, where: EA=errors after aging and: EB= errors before aging

The fit of Δ is the polynomial function [22] of degree 7 (see *Fig. 6.6*).

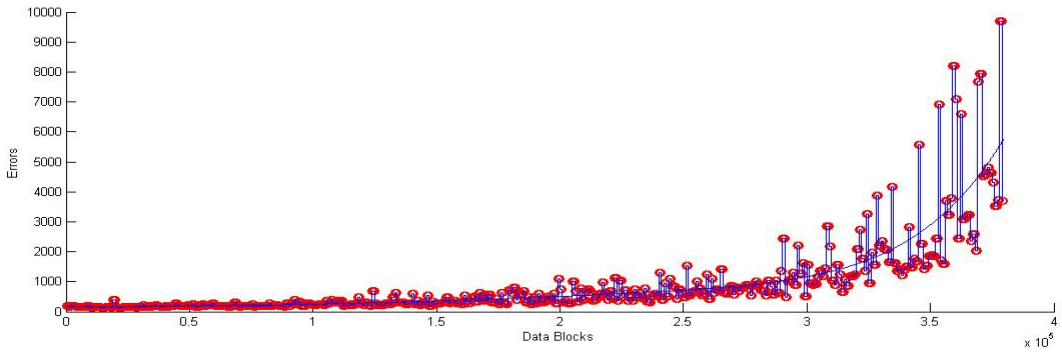


Fig. 6.6 – Blu-Ray discs brand 1 – the best polynomial fit.

6.1.3.1 Fit Tests

To verify that the polynomial of degree 7 is the best choice we will use two methods:

- Residuals Sum of Squared (RSS)
- Coefficient of determination (R^2)

Calculating SSR for the three polynomials of degree 6, 7 and 8 we have that SSR_6= 1.7751, SSR_7=1.7749, and SSR_8=1.7749.
In this case the best value of SSR is obtained by polynomial of the 7 degree because $SSR_7 < SSR_6$, and because the polynomial function of the 8 would not result in a significant reduction of SSR.

Calculating R^2 for the three polynomials of degree 6, 7 e 8 we have that
$R^2(6)$=0.7248, R^2 (7)=0.7252, $R^2(8)$=0.7252.
In this case again, the best value of SSR is obtained by polynomial of the 7.

6.2 CD-R discs – brand 3

6.2.1 Degradation function calculated from the $File_{mean}$

From two plots (see *Fig. 6.7*), calculated on the mean value of the errors (before and after the accelerated aging), now we calculate the degradation increment.

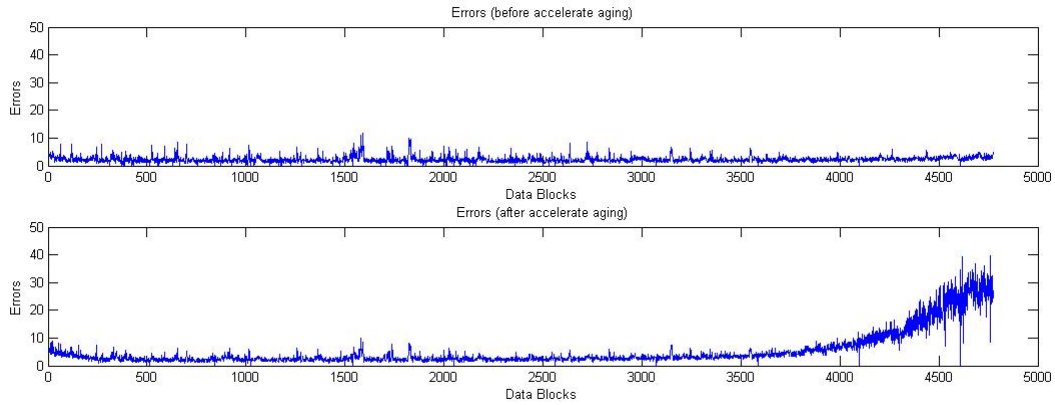


Fig. 6.7 - CD-R discs, brand 3 - errors before and after accelerate aging.

The degradation increment is given as follows:

$\Delta = \sum_{i=1}^{n} (EA_i - EB_i)$, where: EA=errors after aging and: EB= errors before aging

The fit of Δ is the polynomial function of degree 6 (see fig. 6.8).

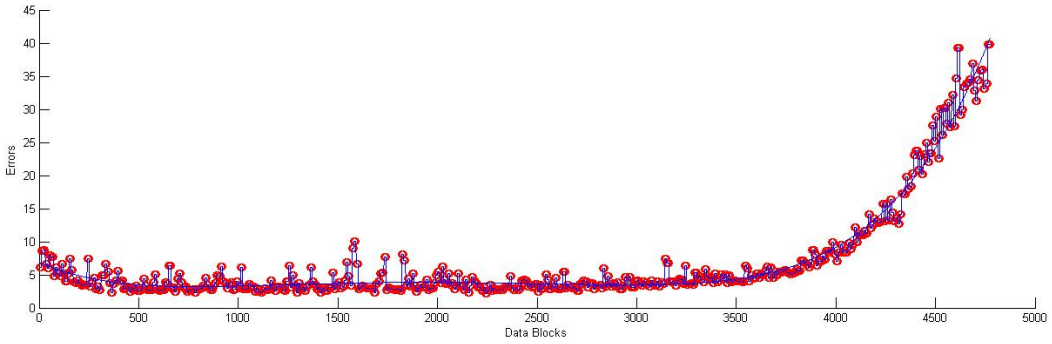


Fig. 6.8 – CD-R discs brand3 – the best polynomial fit.

6.2.1.1 Fit Tests

To verify that the polynomial of degree 6 is the best choice we will use two methods:

- Residuals Sum of Squared (RSS)
- Coefficient of determination (R^2)

Calculating SSR for the three polynomials of degree 5, 6 e 7 we have that SSR_5= 8.3040, SSR_6=7.521, SSR_7=7.7347.

In this case the best value of SSR is obtained by polynomial of the 6 because $SSR_6 < SSR_5$, and $SSR_7 > SSR_6$.

Calculating R^2 for the three polynomials of degree 5, 6 e 7 we have that $R^2(5)=0.9689$, $R^2(6)=0.9710$, $R^2(7)=0.9710$.

In this case again, the best value of SSR is obtained by polynomial of the 6 .

6.2.2 Degradation Function Calculated from the $File_{max}$

From two plots (see *Fig. 6.9*), calculated on the maximum value of the errors (before and after the accelerated aging), now we calculate the degradation increment.

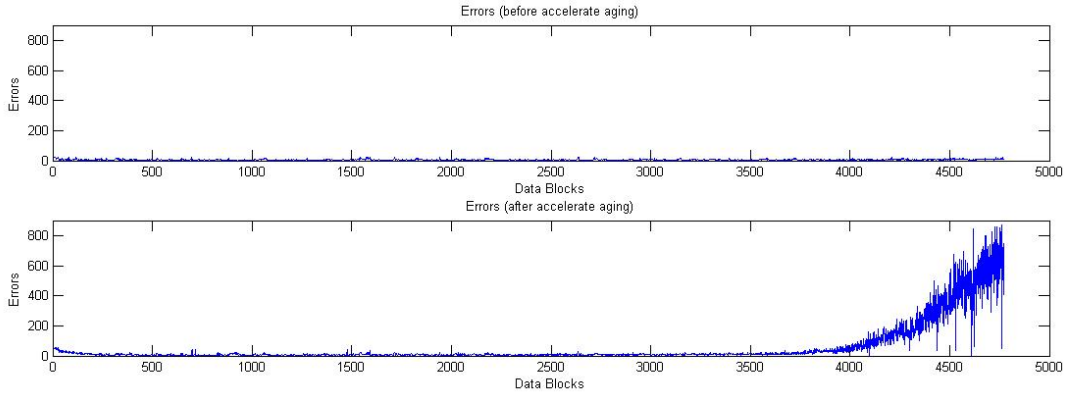


Fig. 6.9 – CD-R brand 3 - errors before and after accelerate aging.

The degradation increment is given:

$\Delta = \sum_{i=1}^{n} (EA_i - EB_i)$, where EA=errors after aging and EB= errors before aging

The fit of Δ is the polynomial function of degree 5 (see *Fig. 6.10*).

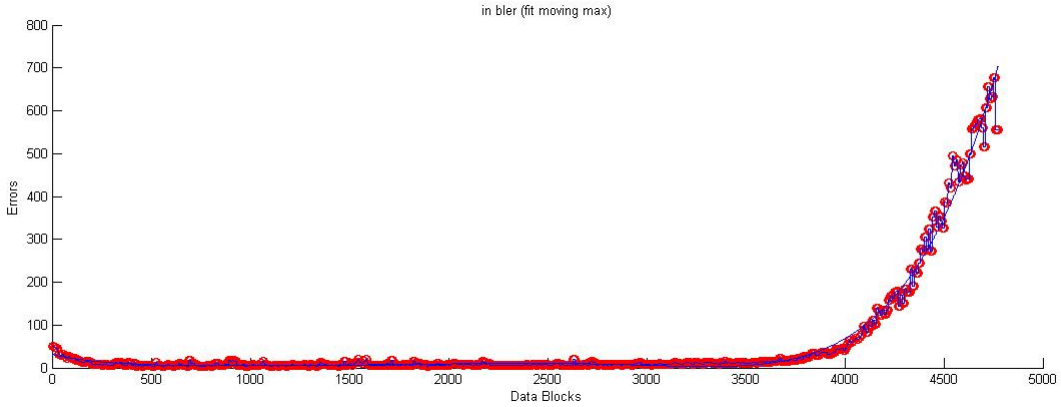


Fig. 6.10 – CD-R discs brand 3 – the best polynomial fit.

6.2.2.1 Fit Tests

To verify that the polynomial of degree 6 is the best choice we will use two methods:

- Residuals Sum of Squared (RSS)

- Coefficient of determination (R^2)

Calculating SSR for the three polynomials of degree 5, 6 e 7 we have that SSR_5= 4.884, SSR_6=2.5795, SSR_7= 2.5793.

In this case the best value of SSR is obtained by polynomial of the 6 because $SSR_6 < SSR_7$, and because the polynomial function of the 7 would not result in a significant reduction of SSR.

Calculating R^2 for the three polynomials of degree 5, 6 e 7 we have that $R^2(5)$=0.9689, $R^2(6)$=0.9710, $R^2(7)$=0.9710.
In this case again, the best value of SSR is obtained by polynomial of the 6.

CHAPTER 7

7. How to Implement an Adaptive RS Code: Parity Redistribution Algorithm and Data Analysis

By using the degradation function, it will be possible to generate a parity file for BDs and CD-R. For the two types of optical discs the algorithms will be operating differently. For BDs the parity algorithm will reposition parity from safe areas to critical areas. On the contrary, for CD-R the parity will be taken from empty areas, thus decreasing the disc capacity.

7.1 Blu-Ray Discs - brand 1

7.1.1 Testing the Parity Algorithm on the File$_{mean}$

In *Fig 7.1a* the plot of the degradation function is shown. This function predicts where are the Data Blocks (after aging) with high number of errors .

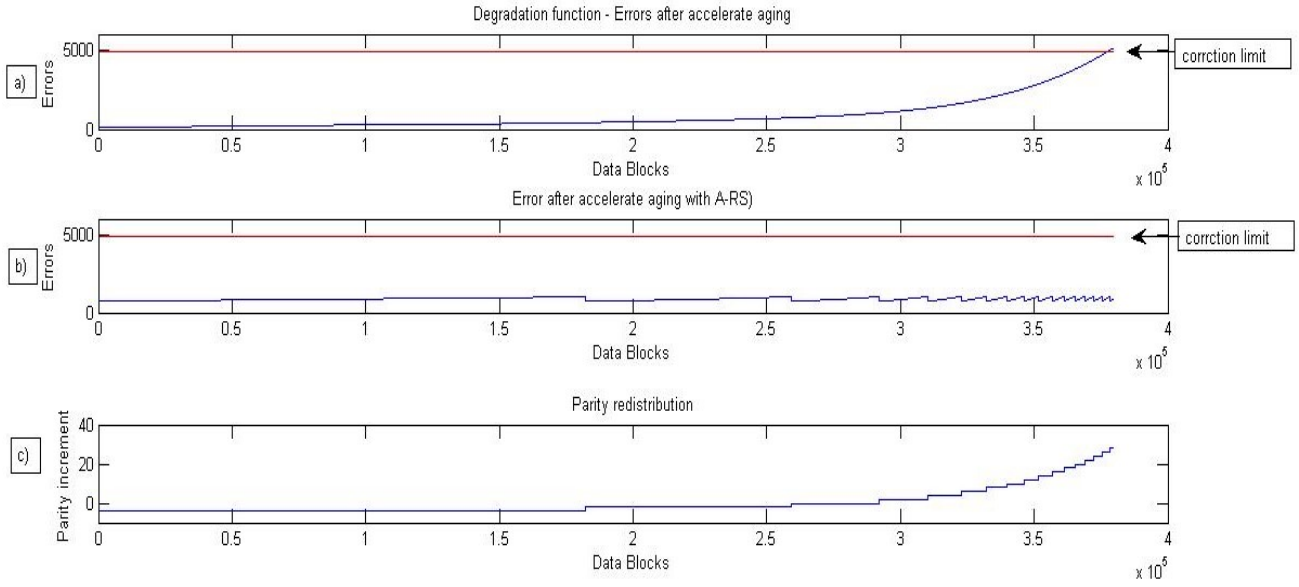


Fig. 7.1 – Blu-Ray discs - brand 1. (a) degradation function (prediction errors), (b) errors after the redistribution of the parity symbols, (c) parity vector calculated by the degradation function.

The parity vector (*Fig. 7.1c*) assigns the amount of parity to each Data Block in according to the errors value indicated by the degradation function. During the burning process, the A-RS will use this information to assigns, to each Data Blocks, the appropriate parity value.

The difference between plots of the *Fig.7.1a* and in *Fig.7.1b* indicates the reliability difference between a standard correction system and our new proposed correction system. An aged disc with an error distribution like the one shown in *Fig. 7.1a*, has already reached the end of life (since the

some errors are greater than *4864*), the old disc, with an errors distribution like *Fig. 7.1b*, it will still have an high expectation of life (since the maximum error, on the whole disc is less than *1000*).

7.1.2 Testing the Parity Algorithm on the File$_{max}$

In *Fig.7.2* it is shown that by using the adaptive parity redistribution it is possible to obtain an uniform the ratio between number of errors and parity symbols. Moreover, the residual parity capability for each *Data Block* is much lower than the maximum correction limit. It is also worth to mention that in this case, the worst-case scenario was analysed: The data we used are obtained from the disc presenting the maximal amount of errors, among the ones we analysed. In this original data, in most cases the number of errors was too high to be corrected by a standard *RS-code*, yet the new adaptive code is able to correct all of them.

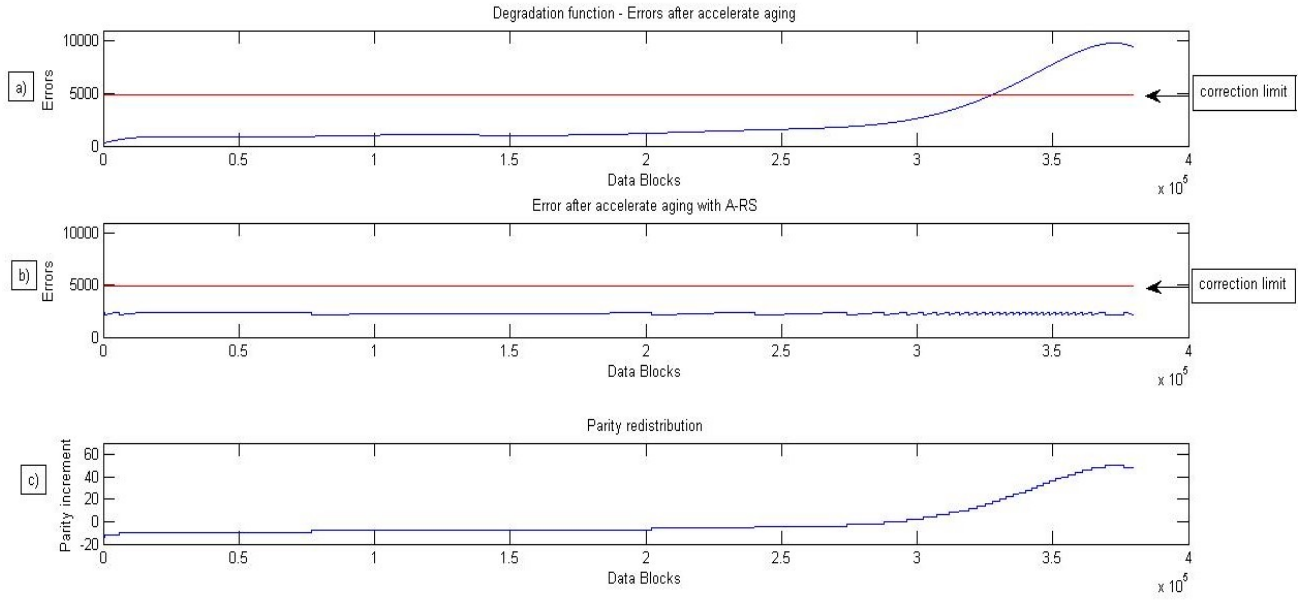


Fig. 7.2 – Blu-Ray discs brand 1. (a) degradation function (prediction errors), (b) errors after the redistribution of the parity symbols, (c) parity vector calculated by the degradation function.

7.1.3 Testing the Parity Algorithm on the File$_{mode}$

In *Fig 7.3* it is also possible to notice that the same results can be obtain by considering the errors calculated from the mode errors.

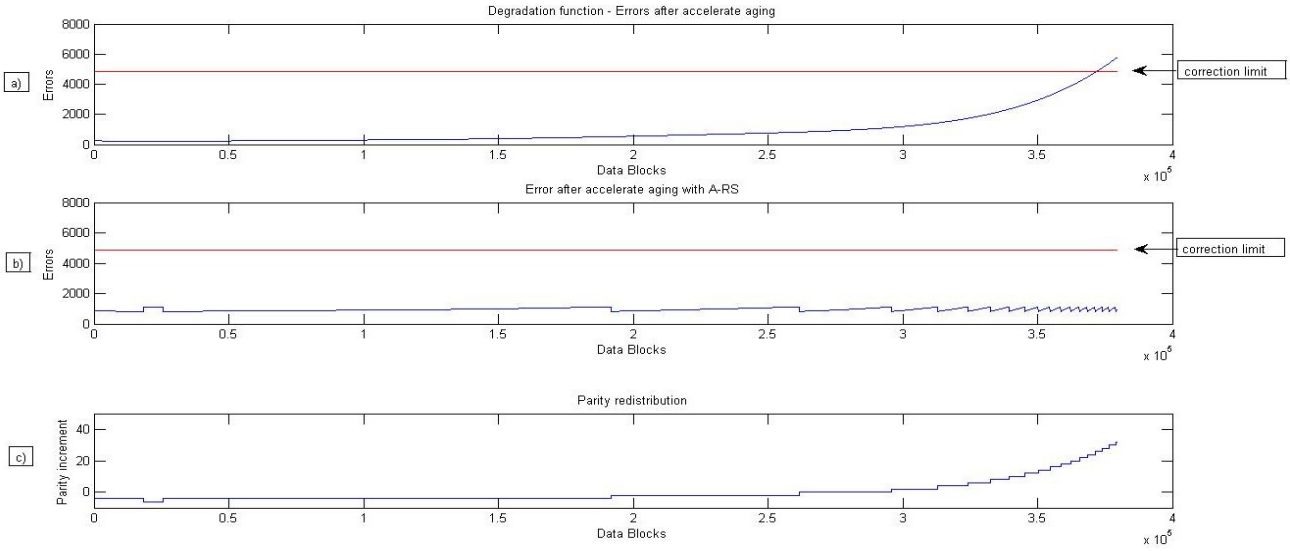


Fig. 7.3 – Blu-Ray discs brand 1. (a) degradation function (prediction errors), (b) errors after the redistribution of the parity symbols, (c) parity vector calculated by the degradation function.

7.2 CD-R discs – Brand 3

7.2.1 Testing the parity algorithm on the $File_{max}$

When considering *CD-R* optical discs it is not possible to apply the redistribution algorithm without, at the same time, diminishing the overall memory capability of the disc. This inconvenience is due to the reduced length of the code-words, that therefore have a limited number of parity symbols. In order to increase the redundancy in the critical areas, the algorithm cannot relocate parity symbols from safe areas to critical areas. Instead, it takes data-free space from the disc. However, the loss in terms of memory capability is negligible. Compare *Figs 7.3a - 7.3b*.

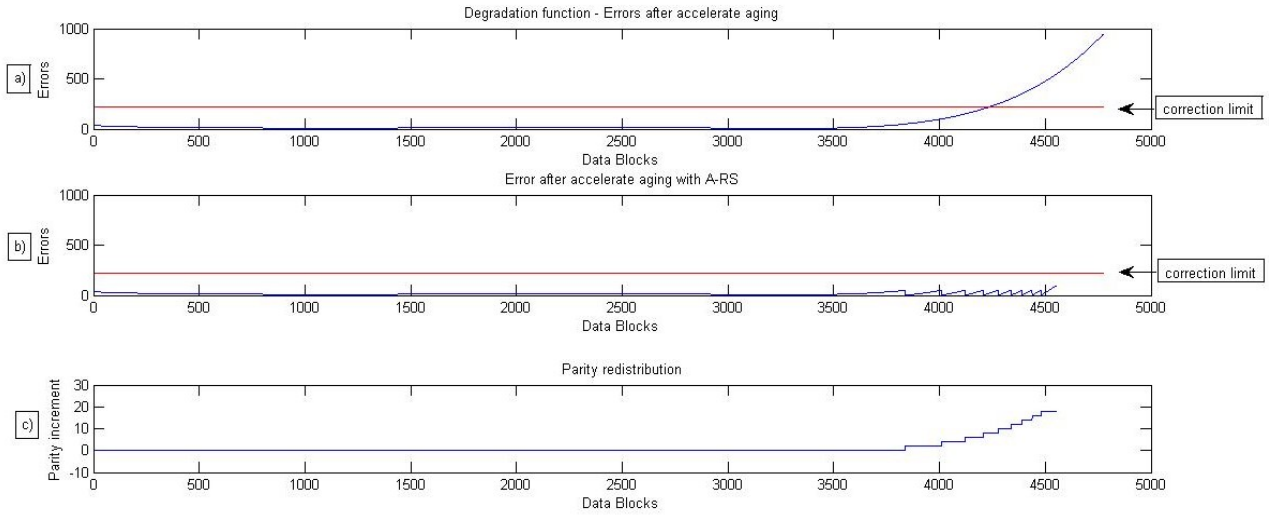


Fig. 7.4 – CD-R discs brand 3. (a) degradation function (prediction errors), (b) errors after the redistribution of the parity symbols, (c) parity vector calculated by the degradation function..

CHAPTER 8

8. Future developments

In the time left after the completion of the thesis project, we tried to investigate what physical or dynamical phenomena could possibly generate this safe and critical regions on the disc. It was possible to observe that these zones are not randomly distributed on the disc, but follow a specific pattern. In particular, the distribution of the errors rises exponentially in the external zones of the discs.

In order to understand this behaviour, burnable discs were compared to factory discs and it emerged that the latter did not form critical zones in their external areas. Due to the fact that the only difference between these two types of optical discs is the presence or the absence of a layer of dye - present in burnable discs and absent in factory discs - the dye itself was addressed as one of the cause of this behaviour.

By focusing on the role of the dye, some interesting conclusions can be drawn when considering the coating methods. In particular, the spin coating is a procedure used to deposit uniform thin films to flat substrates. Usually a small amount of coating material is applied on the center of the substrate. The substrate is then spinned at high speed, to spread the coating material by centrifugal force. Rotation is continued while the fluid spins off the edges of the substrate, until the desired thickness of the film is achieved. The applied solvent is usually volatile, and simultaneously evaporates. So, the higher the angular speed of spinning, the thinner the film. The thickness of the film also depends on the viscosity and concentration of the solution and the solvent.

However, when this method is applied to optical discs it emerges that the thickness of the layer of dye over the polycarbonate matrix is rather anisotropic [76]. In *Fig. 8.1* it is shown that the thickness of the layer is minimal in the central zone of the discs, and then it increases in the external zones. Moreover, it is possible to notice that the thickness follows a trend which is similar to the graph of the inferred degradation function (*Fig. 8.2*).

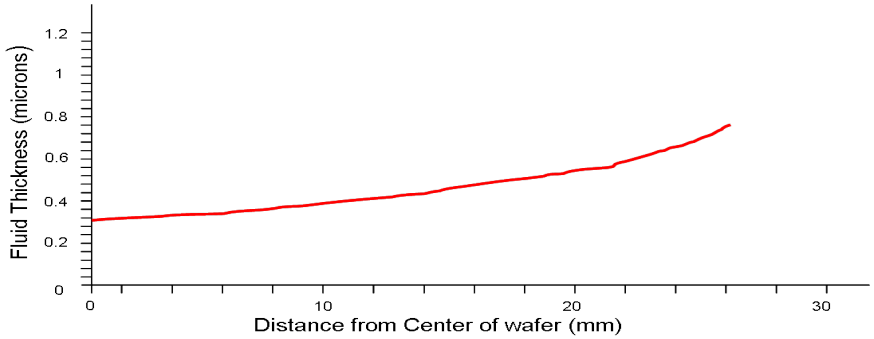


Fig. 8.1- Plot showing the thickness of the fluid as a function of the distance from the center.

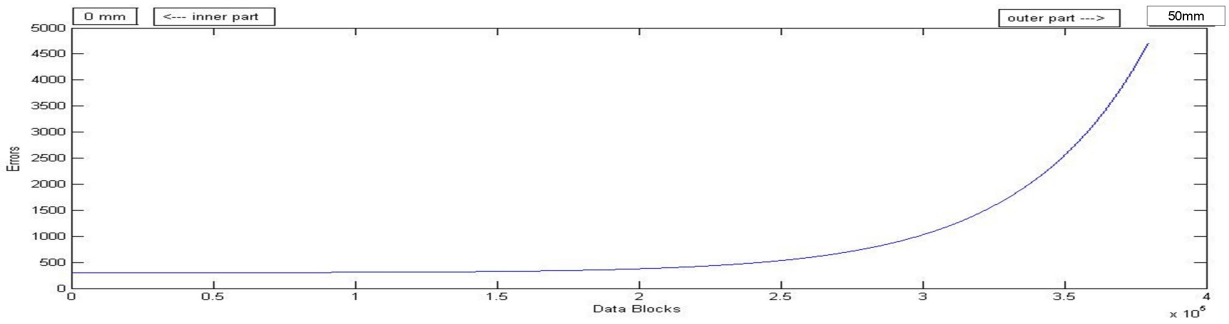


Fig. 8.2- Errors of the BD_s

At this point, we speculated about the correlation between these two trends. The literature [77, 78, 79] indicates that, during the first step of the burning process the power of the laser beam is tuned. The tuning is performed in the central area of the disc, that is in fact the area of minimum dye thickness, as well as the area with the minimum number of errors. Once the power of the laser beam is tuned, it is able to burn the central areas of the discs completely. Therefore we consider highly likely that the laser will no more be able to completely burn the dye when it moves to the outer zones, where the dye layer is thicker.

The exponentially increasing number of errors that was individuated in the critical areas could therefore be linked to the partially burned dye, which then degrades through time. In the safe areas there are less errors, probably because the dye can be completely burned from the polycarbonate surface of the disc, and therefore less decomposition processes will occur.

In the future it would be interesting to further investigate the relation between spin coating, dye thickness, and number of errors. If the existence of a correlation between errors and spin coating is confirmed by further and dedicated analysis, two different technological solutions might be devised:

1- The spin coating technique could be substituted by an alternative methodology, capable to provide a more uniform distribution of the dye on the surface of the discs. However, the development of such technology might require a sensible investment in terms of time and economical resources.
2- The firmware of the burning device could be modified to vary the power of the laser source throughout the burning, in a way chosen to compensate to the trend individuated by the degradation function.

CHAPTER 9

9. Discussion and conclusions

Nowadays, optical discs are still a great device for archiving data storage. Furthermore, they are of great interest for institutions working towards the protection of artistic and cultural heritages, as aforementioned.

Optical discs present many advantages when compared to other types of data storages. For example, Hard Disks (HDs) and Solid State Disks (SSDs) are not bare supports and are affected by electromagnetic fields, among other interferences. On the contrary, optical discs are bare supports, since they have neither moving mechanical parts nor electronic circuits, are not affected by electromagnetic field or by moderate water exposures. As Henry Ford said: "*All that is not there, it does not break*". At the end of this work, it is still not possible to ascertain the real life expectancy of optical discs, yet it is possible to say that it can be extended, by applying the new methodology that we propose.

The adaptive A-RS code is able to protect the critical zones of the optical disc by displacing parity symbols from the safe zones, without affecting the latter. This redistribution of parity symbols allows to increase the reliability and the life-span of the support.

The achievement of these results has been a complex endeavour, by starting from the development of the necessary machineries, and leading to the implementation of the new adaptive system itself. First, the climatic chamber was developed and perfected. The most important feature of the chamber is the set of automatisms that allows to control vital parameters such as humidity and temperature, allowing to gather data by following a clean and reproducible protocol. With this apparatus it was possible to accelerate the aging process for optical discs. In order to reproduce this process with a high degree of accuracy, a complex system was tuned and the necessary software was developed as well.

Second, an automatic testing machine (ATM) has been realised. As well as the climatic chamber, the ATM allowed to operate by controlling the environment, in terms of humidity, temperature, and dust. The combination of these controls allowed to perform the burning and testing tasks in consistent conditions and ensure the reproducibility of the experiences. A robotic system was set up to automatically perform the burning and testing processes. Moreover, the automatic system allowed to avoid any possible contamination of the optical discs.

By using these equipments, it was possible to collect the large data set the allowed us to infer the graph of the degradation function. Once defined, the degradation function was used to identify the optimal parity redistribution on which the adaptive Reed Solomon code is built.
This step allowed a better employment of the parity symbols, that were moved from safe zones to critical ones. Ultimately, we show how this strategy permits to improve the life expectancy of optical discs, without decreasing their capacity

To conclude, according to the proposed A-RS scheme, different areas of the optical discs are protected by different amounts of redundancy, according to their different needs. However, it might be possible to consider as each area was protected by different standard RS codes that had the same length but, at the same time, different dimensions. Due to this, it is important to appropriately evaluate the implementation of both the encoder and of the decoder. On one hand, multiple encoders and decoders may be used. On the other hand, it may be possible to adopt a certain degree of compatibility. However, in both cases there would be some serious problems concerning the complexity of the solutions. Nonetheless, the second case would probably be better, since rate compatibility is usually intended as the capability of an encoder/decoder system to support different code rates. Similar codes are already employed in wireless communication systems, and moreover in the case of optical discs we possess *a priori* knowledge of the position of the *critical* areas of the discs.

REFERENCES

[1] Bradley, Kevin. Risk Associated with the Use of Recordable CDs and DVDs as Reliable Storage Media in Archival Collections: Strategies and Alternatives. UNESCO, 2006.

[2] Choy, Sarah CC, Crofts, Nicholas, Fischer, Robert, et al. “Guidelines for the selection of digital heritage for long term preservation”, UNESCO, 2016.

[3] P.H. Siegel and J.K. Wolf, “Modulation and coding for information storage” ,IEEE Commun. Mag., no. 12, pp. 68-86, Dec. 1991.

[4] S. Lin and D.J. Costello, Error Control Coding Fundamentals and Applications. Prentice-Hall Inc., 1983, Chaps. 4, 6.

[5] S.B. Wicker, Error Control Systems for Digital Communication and Storage. Prentice-Hall International Inc., 1995, Chap. 16.

[6] I.S. Reed and G. Solomon, “Polynomial codes over certain finite fields”, J. Soc. Indust. Appl. Math., vol. 8, pp. 300-304, 1960.

[7] S. Lin and D.J. Costello, Error Control Coding Fundamentals and Applications. Prentice-Hall Inc., 1983, Chap. 2.

[8] K.A.S. Immink, P.H. Siegel, and J.K. Wolf, “Codes for digital recorders,”IEEE Trans. Inform. Theory, vol. 44, no. 6, pp. 2260-2299, Oct. 1998.

[9] Kunej, Drago. "Instability and Vulnerability of CD-r carriers to Sunlight." Audio Engineering Society Conference: 20th International Conference: Archiving, Restoration, and New Methods of Recording. Audio Engineering Society, 2001

[10] Slattery, Oliver, et al. "Stability Comparison of Recordable Optical Discs-A study of error rates in harsh conditions." Journal of Research-National Institute of Standards and Technology 109.5 (2004): 517.

[11] Van Ooijen, P. MA, et al. "Accessibility of Data Backup on CD-R after 8 to 11 years." Journal of digital imaging 23.1 (2010): 95-99.

[12] Hasan, M. Anwarul, and Vijay K. Bhargava. "Architecture for a low complexity rate-adaptive Reed-Solomon encoder." IEEE Transactions on Computers 44.7 (1995): 938-942.

[13] Chen, Zheng Xiong, et al. "A programmable Reed-Solomon codec processor." Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium on. Vol. 1. IEEE, 2001.

[14] ISO 30190, “Information technology –Digitally recorded media for information interchange and storage -120 mm Single Layer (25,0 Gbytes per disk) and Dual Layer (50,0 Gbytes per disk) BD Recordable disk”, 2013-07-01.

[15] Turner, Shawn M. Guidelines for developing ITS data archiving systems. No. Report 2127-3. 2001.

[16] Van Bogart, J. W. C., and P. Wetzel. "Mag tape life expectancy 10-30 years." Online: http://cool. conservation-us. org/bytopic/electronic-records/electronic-storage-media/bogart. html (1995).

[17] Bate, Geoffrey. "The limits to magnetic recording." Compcon Spring'88. Thirty-Third IEEE Computer Society International Conference, Digest of Papers. IEEE, 1988.

[18] Pinheiro, Eduardo, Wolf-Dietrich Weber, and Luiz André Barroso. "Failure Trends in a Large Disk Drive Population." FAST. Vol. 7. 2007.

[19] Schroeder, Bianca, and Garth A. Gibson. "Disk failures in the real world: What does an mttf of 1, 000, 000 hours mean to you?." FAST. Vol. 7. 2007.

[20] Schroeder, Bianca, Raghav Lagisetty, and Arif Merchant. "Flash Reliability in Production: The Expected and the Unexpected." FAST. 2016.

[21] Hutsell, Woody, Jamon Bowen, and Neal Ekker. "Flash solid-state disk reliability." Texas Memory Systems White Paper (2008).

[22] ISO/IEC 10995, “Information technology - Digitally recorded media for information interchange and storage - Test method for the estimation of the archival lifetime of optical media” 2011-06-15

[23] ISO/IEC 18927, “Imaging materials - Recordable compact disc systems - Method for estimating the life expectancy based on the effects of temperature and relative humidity” 2013-03-01.

[24] Optical Disc Archive Generation 2 WHITE PAPER http://www.sonypro-latin.com/pro/lang/ls/mx/support/brochures/1237494629249

[25] Panasonic to commercialize Facebook's Blu-ray cold storage systems http://www.pcworld.com/article/3019413/panasonic-to-commercialize-facebooks-blu-ray-cold-storage-systems.html

[26] Rao, Thammavarapu RN, and Eiji Fujiwara. "Error control coding for computer systems." Prentice-Hall Inc. (1989).

[27] Chang, Hsie-Chia, C. Bernard Shung, and Chen-Yi Lee. "A Reed-Solomon product-code (RS-PC) decoder chip for DVD applications." IEEE Journal of Solid-State Circuits 36.2 (2001): 229-238.

[28] Cho, Yongwoo, et al. "Proactive Reed-Solomon bypass (PRSB): a technique for real-time multimedia processing in 3G cellular broadcast networks." Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on. IEEE, 2005.

[29] Wicker, Stephen B., and Vijay K. Bhargava. Reed-Solomon codes and their applications. John Wiley & Sons, 1999.

[30] Arnal, Fabrice. Optimisation de la fiabilité pour des communications multipoints par satellite géostationnaire. Diss. Télécom ParisTech, 2004.

[31] ETSI, EN. "300 744 V1. 6.1 (2009-01),“Digital video broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television,”." Standards Inst.(ETSI), Valbonne, France (2001).

[32] Stylianakis, V., and S. Toptchiyski. "A Reed-Solomon coding/decoding structure for an ADSL modem." Electronics, Circuits and Systems, 1999. Proceedings of ICECS'99. The 6th IEEE International Conference on. Vol. 1. IEEE, 1999.

[33] Alliance, HomePlug Powerline. "Medium Interface Specification. Release 0.5." (2000).

[34] Council, D. A. "DAVIC 1.3 Specification Part 8: Lower Layer Protocols and Physical Interfaces, 1997."

[35] Advanced Television Systems Committee. "ATSC Standard: Digital Television Standard (A/53), Revision D, Including Amendment No. 1." Doc A D 53 (2005).

[36] Luby, Michael, et al. Forward error correction (FEC) building block. No. RFC 3452. 2002.

[37] Song, Mook Kyou, et al. "Architecture for decoding adaptive Reed-Solomon codes with variable block length." IEEE Transactions on Consumer Electronics 48.3 (2002): 631-637.

[38] Almulhem, A., F. El-Guibaly, and T. A. Gulliver. "Adaptive error correction for ATM communications using Reed-Solomon codes." Southeastcon'96. Bringing Together Education, Science and Technology., Proceedings of the IEEE. IEEE, 1996.

[39] Singh, Ranjita. "VHDL Implementation of Reed-Solomon Encoder-Decoder for WiMax Network." (2014).

[40] Chaari, Lamia, et al. "A reconfigurable FEC system based on Reed-Solomon codec for DVB and 802.16 network." WSEAS transactions on circuits and systems 8.8 (2009): 729-744.

[41] Uno, Arduino. "“Arduino– ArduinoBoardUno." (2015).

[42] Badamasi, Yusuf Abdullahi. "The working principle of an Arduino." Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on. IEEE, 2014.

[43] Bertrand, Ennio. "Soffio (Breath). Interactive Poetry and Words Installation." ESSEM@ AI* IA. 2013.

[44] Marriner, Paul. "Temperature and humidity control." Gazette 13.1 (1980): 12-17.

[45] Rowe, David Michael, ed. CRC handbook of thermoelectrics. CRC press, 1995.

[46] Wood, Lawrence A. "The use of dew-point temperature in humidity calculations." J Res Nat Bur Stand 74c (1970): 117-122.

[47] Ogata, Katsuhiko, and Yanjuan Yang. "Modern control engineering." (2010).

[48] Ang, Kiam Heong, Gregory Chong, and Yun Li. "PID control system analysis, design, and technology." IEEE transactions on control systems technology13.4 (2005): 559-576.

[49] Seyedreza Fattahzadeh. “Implementing of an Arduino based Temperature controller with PID algorithm” (2015).

[50] Honguh, Yoshinori. "Analysis of optical disk readout signal deterioration caused by dust on the substrate." Optical Review 2.1 (1995): 14-19

[51] Vembu, Sridhar. "System and method for optimized power control." U.S. Patent No. 6,259,928. 10 Jul. 2001.

[52] Roche, Emmanuel, and Yves Schabes. Finite-state language processing. MIT press, 1997.

[53] Fransson, Tobias. "Simulators for formal languages, automata and theory of computation with focus on JFLAP." (2013).

[54] Jacobson, Nathan. Lectures in Abstract Algebra: III. Theory of Fields and Galois Theory. Vol. 32. Springer Science & Business Media, 2012.

[55] Birkhoff, G. and Mac Lane, S. "Insolvability of Quintic Equations." §15.8 in A Survey of Modern Algebra, 5th ed. New York: Macmillan, pp. 418-421, 1996.

[56] J. Westall, J. Martin. “An Introduction to Galois Fields and Reed-Solomon Coding” Clemson University (2010)

[57] Lidl, Rudolf, and Harald Niederreiter. Introduction to finite fields and their applications. Cambridge university press, 1994.

[58] Sankaran, Jagadeesh. "Reed Solomon decoder: TMS320C64x implementation." Application Report SPRA686, Texas Instruments (2000).

[59] Geisel, William A. "Tutorial on Reed-Solomon error correction coding." (1990) - NASA.

[60] Lim, Raymond S. "A decoding procedure for the Reed-Solomon codes." (1978) - NASA.

[61] Sklar, Bernard. Digital communications. Vol. 2. Upper Saddle River: Prentice Hall, 2001.

[62] Song, Mook Kyou, et al. "Architecture for decoding adaptive Reed-Solomon codes with variable block length." IEEE Transactions on Consumer Electronics 48.3 (2002): 631-637.

[63] Horn, Uwe, et al. "Robust internet video transmission based on scalable coding and unequal error protection." Signal Processing: Image Communication 15.1 (1999): 77-94.

[64] ISO/IEC 9660, “Information technology – Volume and file structure of CD_ROM for information interchange”, 1999-01-14.

[65] ISO/IEC 1448, “Information technology – “120 mm DVD – Read-only disk”, 2002-04-15.

[66] ISO/IEC 10149, “Information technology – “Data interchange on read-only 120 mm optical data disks (CD-Rom), 1995-07-15.

[67] Johnson, Norman L., Samuel Kotz, and N. Balakrishnan. "Lognormal distributions. " Continuous univariate distributions 1 (1970).

[68] Meadows, Donella H., Dennis L. Meadows, Jørgen Randers, and William W. Behrens III. (1972) The Limits to Growth. New York: University Books. ISBN 0-87663-165-0

[69] Hazewinkel, Michiel, ed. (2001), "Exponential function", Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4

[70] Coles, Stuart (2001). An Introduction to Statistical Modeling of Extreme Values. Springer - Verlag. ISBN 1-85233-459-2.

[71] Bland, J. Martin, and Douglas G. Altman. "Statistics notes: measurement error" Bmj 313.7059 (1996): 744.

[72] Sandra Lach Arlinghaus, PHB Practical Handbook of Curve Fitting. CRC Press, 1994

[73] William M. Kolb. Curve Fitting for Programmable Calculators. Syntec, Incorporated, 1984.

[74] Draper, N.R.; Smith, H. (1998). Applied Regression Analysis (3rd ed.). John Wiley. ISBN 0-471-17082-8

[75] Anderson-Sprecher, Richard. "Model comparisons and R 2." The American Statistician 48.2 (1994): 113-117.

[76] Toolan, Daniel TW, and Jonathan R. Howse. "Development of in situ studies of spin coated polymer films." Journal of Materials Chemistry C 1.4 (2013): 603-616.

[77] ECMA-394, “Recordable Compact Disc Systems CD-R Multi-Speed”, 2010-12.

[78] ISO/IEC 23912, “Information technology – 80 mm (1,46 Gbytes per side) and 120 mm (4,70 Gbytes per side) DVD Recordable Disk (DVD-R)”, 2005.

[79] ISO/IEC 30190, “Information technology – Digitally record media for information interchange and storage 120 mm Single layer (25,0 Gbytes per disk) and Dual Layer (50,0 Gbytes per disk) BD Recordable disk”, 2016.