Aston University

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional

École Nationale Supérieure des Télécommunications de Bretagne

Joint source-channel coding: application to speech coding

Dissertation by

Bertrand Mollinier Toublet

for the programme of

Master of Science in Telecommunications Technology

Aston University

Note

This is a joint research program led by the Cinvestav in Mexico and the ENST de Bretagne in France.

It is effectively led as a scientific cooperation exchange between France and Mexico, managed and funded by ECOS-Nord in France and ANUIES/CONACYT in Mexico: while the responsible teachers in the respective universities managed the project, a French student went to Mexico for four months from April 20th to August 20th, and a corresponding Mexican student went to the French university from September on.

Summary

While mainly a Matlab and C exercise in its realisation, this project is based on theories of both speech-optimised coding, also referred to as vocoding, which is an optimisation of standard audio signal coding for the specificity of speech, and joint source and channel coding, a novel theory of general signal coding, that goes in exactly the opposite direction as the one established by the Shannon theorem of separation.

Given the conditions of the project, I had to decide, jointly with Arturo Veloz and Jean-Marc Boucher, the two professors responsible for this project, and Fernando Villavicencio, the exchange student associated with the project, on a “close-to-real-life” application. Considerations of practicality and feasibility led us to decide to choose and include the vocoder developed by Fernando for his MSc project, as well as the North American CDMA standard IS-95A into our application. The application would support experimentation on joint coding over fading channels, as required in the project outline (refer to Appendix A).

This dissertation aims at reporting the work achieved during the project, and the results obtained. It is organised in three chapters. Chapter 1 will present background information and report some state-of-the-art in speech coding and joint source-channel coding. In turn, Chapter 2 will present in greater detail the components chosen in the project: Fernando Villavicencio’s TCENLP vocoder, and the IS-95A CDMA standard. Finally, Chapter 3 will present implementation details of the project, as well as the results we could obtain from our study.

Unfortunately, the implementation of a working system proved to take up most of the time resource allocated for the project so that while implementation details are numerous, results are few. While our attempt at a scheme of joint coding proved a failure, it taught the author a valuable lesson. Furthermore, the project leaves a working implementation that may be re-used later on for more ambitions projects.

Résumé

Bien que ce projet soit principalement un exercice de C et de Matlab dans sa forme, il est basé sur les deux théories du codage optimisé pour la parole, aussi connu comme le vocodage, qui se définit comme une optimisation du codage audio standard, adaptée aux spécificités de la parole, et du codage conjoint source-canal, une théorie émergeante de codage de données, qui va exactement a l'opposé de la ligne établie par le théorème de séparation de Shannon.

Étant données les conditions du projet, il a fallu décider, conjointement avec Arturo Veloz et Jean-Marc Boucher, les deux professeurs responsables du projet, et Fernando Villavicencio, l'étudiant mexicain associé au projet, d'une application proche de la réalité. Des considérations pratiques nous ont menés à choisir d'inclure dans notre application le vocodeur développé par Fernando dans le cadre de sa maîtrise, ainsi que le standard CDMA nord-américain IS-95A. L'application ainsi développée supporterait l'expérimentation de codage conjoint sur un canal de type Rayleigh, ainsi que requis dans la définition du projet.

Le but de cette dissertation est de rapporter le travail effectué au cours du projet, ainsi que de présenter les résultats obtenus. Elle est organisée en trois chapitres. Le premier présente les théories et l'état de l'art en matière de codage de parole et de codage conjoint. Le chapitre 2 présente pour sa part plus en détail les composants utilisés : le codeur TCENLP de Fernando, et le standard IS-95A. Finalement, le chapitre 3 présentera les détails de l'implémentation de notre application, ainsi que les résultats de nos analyses.

Malheureusement, l'implémentation d'une application qui tourne a consommé la plupart des ressources allouées au projet, si bien que les détails d'implémentation sont plus fournis que les résultats. Bien que notre essai de codage conjoint se soit avéré être un échec, l'expérience a permis à l'auteur de se débarrasser d'une idée préconçue. De plus, le projet a permis la réalisation d'une application qui pourra être réutilisée pour de futurs et plus ambitieux projets.

Introducción

Aunque principalmente un ejercicio de C y Matlab en su realización, este proyecto esta basado en ambas teorías de codificación optimizada para la voz, o más bien vocoding, que consiste en una optimización de la codificación estándar de señal audio para la especificidad de la voz, y de codificación conjunta de fuente y canal, una teoría nueva de codificación de señal arbitraria, que va en el rumbo opuesto del establecido por el teorema de separación de Shannon.

En relación con las condiciones del proyecto, tuve que decidir, conjuntamente con los profesores Arturo Veloz y Jean-Marc Boucher, encargados del proyecto, y Fernando Villavicencio, el estudiante mexicano asociado al proyecto, de una aplicación "realista". Decidimos elegir e incluir el vocoder TCENLP desarrollado por Fernando para su proyecto de Maestría, tal como el estándar CDMA Norte-Americano IS-95A en nuestra aplicación. La aplicación soportaría la experimentación con codificación conjunta sobre canales de tipo Rayleigh, como especificado en el proyecto (en Appendix A.)

Está disertación tiene como objetivo presentar el trabajo hecho durante el proyecto, y los resultados obtenidos. Esta organizada en tres capítulos. Capítulo 1 presentara la teoría y el estado del arte en codificación de voz y codificación conjunta. A su vez, el capítulo 2 presentara mas detalles de los componentes elegidos en el proyecto: el vocoder TCENLP de Fernando y el estándar CDMA IS-95A. Finalmente, el capítulo 3 presentara detalles de implementación del proyecto, como resultados que pudimos obtener de nuestros estudios.

Desgraciadamente, la implementación de un sistema que funciona ocupó la mayor parte de los recursos del proyecto, así que aunque detalles de implementación son numerosos, resultados son pocos. Aunque el esquema de codificación conjunta que intentemos implementar resultó en un fracaso, el autor aprendió una lección valuable. Más bien, el proyecto deja una implementación que funciona y que podrá ser reutilizada en proyectos futuros.

Acknowledgements

This project, besides my coming to the lab and doing some work from time to time, owes much to the effort of many, that either helped me, pushed me or supported me in one way or another before, during and after the stay in Guadalajara. Many thanks to them all.

First of all, thank you to Jean-Marc Boucher, at ENST de Bretagne, for accepting me into this project. I sure hope he will not be disappointed in the work achieved during the project. I also wish to thank los doctores Arturo Veloz y Deni Torres Román for the welcome at the Cinvestav, the friendly support, and for making material considerations so much easier for me.

My fellow co-workers also deserve their fair share of acknowledgement: Fernando Villavicencio, for all the interesting insights on the Mexican way of life – table dance not being the least of all –, Miguel Angel Alonso Arévalo, for letting me discover the not-so-subtle taste of beef's virility, Javier, Alex, Chuy, Edgar and Verdin at the lab for the Mexican (and Mayan) vocabulary lessons, the nights at the pool and the unforgettable concert at la Concha Acústica.

I also need to thank Lydie and Delphine at ECOS-Nord for their strong and very welcome support and help in obtaining the Mexican visa, as well as an anonymous lady at the Mexican Consulate for allowing the bending of the – otherwise extremely impractical – opening times of the Consulate.

Maria and Manuel de Manjarrez deserve all my affection for the warm welcome at their home during our four months stay and for giving me the feeling of being at home ten thousand kilometres away from it.

Finally, I'd like to thank my Bebbe for the good times always.

Table of contents

Table of figures

Glossary of Symbols and Abbreviations

AWGN:	Additive White Gaussian Noise
BER:	Bit Error Rate
BPSK:	Binary Phase-Shift Keying
BSA:	Binary Switching Algorithm
CDMA:	Code Division Multiple Access
COSQ:	Channel-Optimized Scalar Quantisation
COVQ:	Channel-Optimized Vectorial Quantisation
DoD:	The US Department of Defense
IA:	Index Assignment
LPC:	Linear Prediction Coding or Linear Prediction Coder
LTP:	Long Term Prediction or Long Term Predictor
MORVQ:	Modulation Organized Vector Quantisation
PN:	Pseudo-random Noise
SAA:	Simulated Annealing Algorithm
SOHC:	Self-Organized Hyper Cube
SOM:	Self-Organized Map
STP:	Short Term Prediction or Short Term Predictor
TCENLP:	Trained Code-Excited Non-Linear Predictor
Vocoder:	Voice Coder

Chapter 1 Background

While mainly a Matlab and C exercise in its realisation, this project is based on theories of both speech-optimised coding, also referred to as vocoding, which is an optimisation of standard audio signal coding for the specificity of speech, and joint source and channel coding, a novel theory of general signal coding that goes in exactly the opposite direction as the one established by the Shannon separation theorem.

In this chapter, both theories are summed up and presented, in order to give the reader a minimum background about the subject of this dissertation.

1.1 Vocoding

According to Scott McCloud, all forms of media are motivated by our inability to directly communicate from mind to mind. The idea is then that a thought of our mind be translated to the physical world, travel through it, hopefully as untouched as possible, to be received and perceived as a thought. Speech is one of the many media[1] conveying this indirect, imperfect and apparently not perfectible mind-to-mind communication [7].

1.1.1 The speech chain

Among all types of audio data encoding, voice coding stands apart for the simple reason that voice signals are produced by a very specific source: the human vocal organs. Consequently, careful analysis of those organs, and the way voice signals are produced, might help design adapted and optimised coding schemes thus

[1] Note that the medium analyzed by Scott McCloud in his very good analysis [7] happens to be the comics. Of course this is irrelevant to this dissertation, but I thought his proposition interesting enough to be mentioned here.

drawing better results – as far as encoding efficiency and quality are concerned – than on more general audio signals.

Let us first take a look at an example of a speech chain, explained in detail in [1], as illustrated below. We can identify as in other contexts, e.g. the OSI network stack, several interacting layers. The top, most abstract layer would be the linguistic layer, taking place in the brain, where the message the speaker wants to convey is formed, the sentence prepared, the words ordered.

The next level is the physiological level: based on the information prepared in the brain, several parts of the body work in conjunction in order to produce a sensible sound that will be emitted at the mouth. The sound waves then propagate through the aerial medium (at least in most common cases) to the listener's ear. This is the acoustic level.

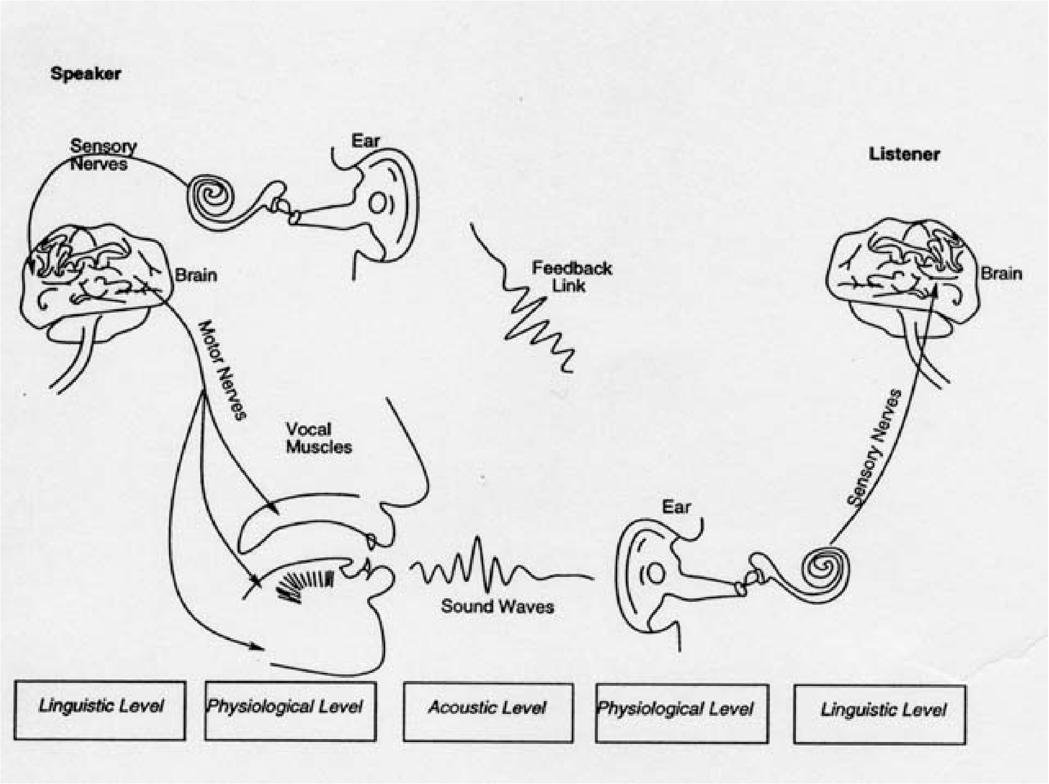

Figure 1-1 The speech chain (from [1])

Upon reaching the listener's ear, the acoustic waves stimulate the listener's auditory organs, which in turn translate the acoustic excitation into a message for the listener's brain. This operation is considered to take place at the physiological level again. Finally, the nervous message from the listener's ear reaches the listener's brain, where they are decoded into sensible information, at the physiological level.

Notice that the acoustic waves emitted by the speaker also reach his own ears, where they are also decoded through the physiological and linguistic layers. This works as a feedback link to allow the speaker for a better control over the sounds emitted. As a consequence, it is no surprise that deaf people have trouble obtaining a clear pronunciation.

1.1.2 Speech production

Careful analysis of the physiological production of speech allows for satisfying means of artificial speech synthesis.

The first step in speech production is the contraction of the lungs with the help of the diaphragm, creating a flow of air through the vocal tract. The glottis is the first organ the flow of air goes through. There, depending on the sound uttered, the flow may or may not be made a quasi-periodic signal. If it is, the utterance is said to be voiced. In that case, the base frequency of the signal is called the pitch. It is linked to the general perception of the speaker's tone of voice. It will be generally higher in frequency for children and women, and lower for men. Typical values for the pitch range from a hundred hertz to a few hundred hertz. If the flow of air is left untouched, the utterance is said to be non-voiced. Also some utterances consist of a mix of voiced and non-voiced sounds, in variable proportions.

Next the flow reaches the velum. When it is open, acoustic coupling with the nasal cavity occurs. When the coupling occurs, the utterance is said to be nasal: as an example, consider the sound of the consonants 'm' and 'n' in the English language.

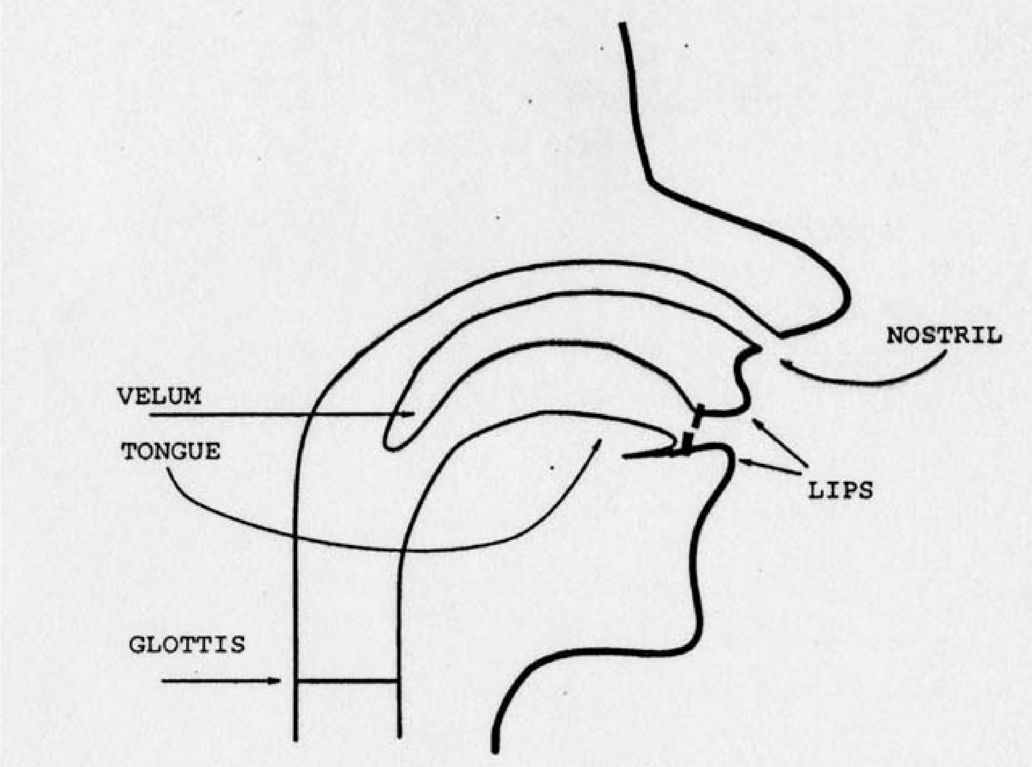

Figure 1-2 The primary articulators of the vocal tract (from [1])

Finally, the tongue, and lips provide fine shaping of the air flow in order to produce a wide range of sounds.

For speech coding purposes, the physiological production of speech is separated into two steps. The first one is excitation, which will either consist of white noise, corresponding to non-voiced sounds, or a periodic signal at an adequate frequency, corresponding to voiced sounds, or a mix of both. The other one is the vocal tract, which may be considered like a filter shaping the excitation. For the needs of vocoding, the vocal tract can be considered as a time varying filter.

Thus encoding will consist in extracting the three basic pieces of information from the speech source: the voicing information, indicating the relative amount of voiced and non-voiced signals in the excitation, the pitch information, if the voiced signal is present in the excitation, and the vocal tract filter parameters.

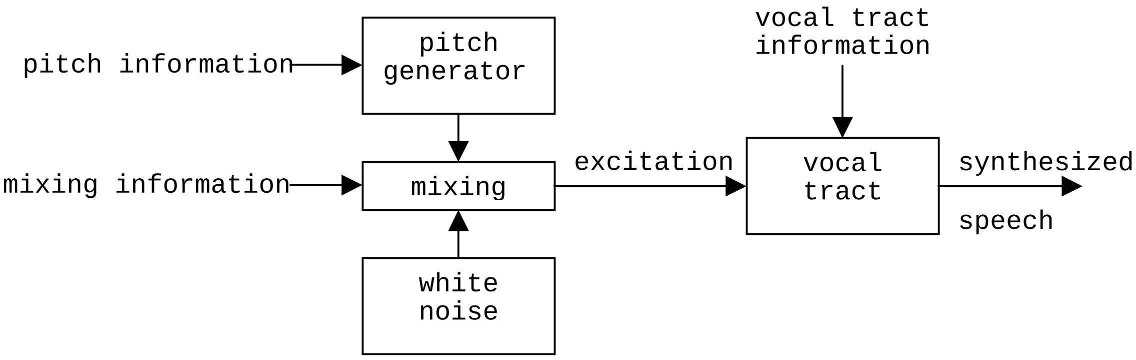

Figure 1-3 The source-filter model

Most vocoders are based on the source-filter model, the only differences being in the practical implementation choices of the different components. For example, the mixing stage is commonly designed as a hard decision between voiced and non-voiced. While allowing for greater simplicity both at analysis and synthesis stages, this approach leads to slightly reduced performance as far as the quality of the synthesised signal is concerned.

1.2 Joint source and channel coding

1.2.1 Tandem source and channel coding

As a general contract, digital transmission is expected to provide efficiency, reliability and privacy. Putting privacy aside, we can see that the goals of efficiency and reliability require contradicting approaches.

Indeed, efficiency, gauged as a ratio between amount of information transmitted and the energy required to transmit it, will require that redundancy be stripped of a given source prior to transmission. This action of optimising efficiency for data to be transmitted is called source coding. On the other hand, reliability of transmission will require some ordered redundancy to be left in the transmitted data, so the correct data can be deduced from errored data. This is called channel coding.

Let’s consider as an example the English language, and the Arab numbering system, as in: “There are 23 students present today.”[2] The natural language is

[2] This is of course a wink to Dr Wrenn’s Information theory and coding course, though I think some of his twenty-three students were sleeping.

naturally extremely redundant, and indeed, a few errors in the above sentence still let the reader get the meaning of it: “Tkere are 23 stndents pretent today.” On the other hand, our numbering system bears no redundancy at all, and even a single error totally changes the conveyed meaning: “There are 93 students present today.” Here, the small class has become a big amphitheatre!

On the other hand, all that redundancy present in the natural language terribly reduces its space-to-information ratio: we just need to compare “twenty three” to its non-redundant counterpart “23”.

However, this apparent opposition in goals is – partly – dismissed by the theorem, proved by Shannon, that source and channel coding can be separated into two disjoint operations without any loss of performance. Indeed, given a certain source of data, there is a limit on the channel that can carry this data in a reliable manner. In other words, the theorem mentioned above states that for any arbitrary channel satisfying the above condition, data could be transmitted over it, with an arbitrarily high reliability.

Now this theorem is not as miraculous as it seems, for it is hampered by several drawbacks. First of all, it gives no clue as to how to obtain an optimal solution. Second, it is not based on any assumption of practicality. Given that most closely optimal solutions obtained so far have been far from practical, either by requiring too high computation delays, or too much computation, it is assumable that an optimal solution would require arbitrarily impractical implementations. Further, the theorem does not prove the uniqueness of the solution either, which allows for other directions to be investigated.

1.2.2 Joint coding

Thus the idea of joint source and channel coding: since separate optimisation of source and channel coding, as mentioned in the above theorem has no guarantee of uniqueness, one could try and jointly optimise the two steps of coding, and see whether that would lead to a better practicality-to-performance ratio.

Some hints as to what specific direction to take are along the lines of noticing that since source coding does not practically remove all redundancy in the source, a decoder could try and take advantage of that residual redundancy for protection against channel errors. A widely investigated application of this is known as structured codeword assignment, where efforts are made so that errors introduced at channel level create the least amount possible of distortion after decoding.

At channel level, since channel coding does not protect against all errors, source coding might be designed to be robust against decoding errors. Channel optimised source encoders, aware of the properties of the channel they are coding for, explicitly include error-correcting code during source coding.

Other options include intelligent decoding, where the decoders knows the characteristics of the source they are decoding, and thus ignore unlikely decoded signals, as well as unequal error protection, where data is more or less heavily error protected depending on its sensitivity.

To illustrate the different possible approaches of joint coding, consider a typical data transmission diagram. Raw data from the source first undergoes some sort of transformation, f, to be transformed into the source sample. After transformation, the samples are quantised (operation Q), and the quantised samples are assigned a representation (operation IA: index assignment). Finally, the signal is channel-encoded (operation CC).

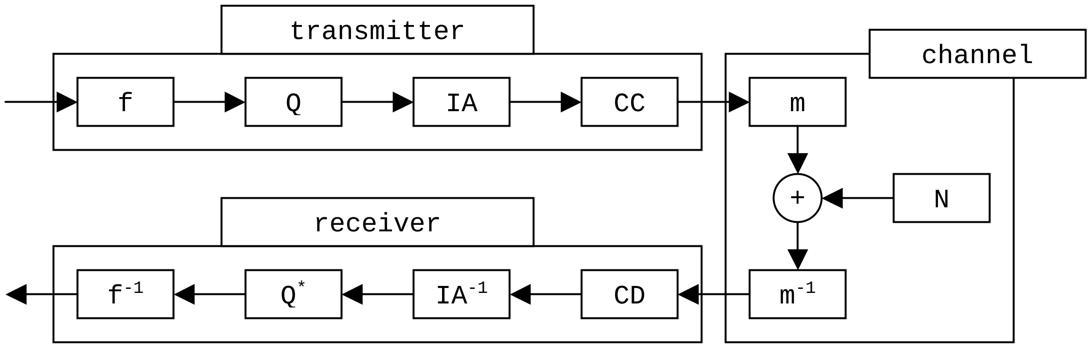

Figure 1-4 The transmission model

At the exit of the transmitter, the signal is modulated (operation m), sent over the channel, which adds transmission noise N, and demodulated (operation m^{-1}). Finally, at the receiver, the complementary operations of transmission take place: the signal is first of all channel-decoded (operation CD); the quantised samples are then recovered through the complementary of the index assignment operation. Finally, the samples are de-quantised and sent through the inverse transformation f^{-1}.

Given these operations, several levels of focus are available when attempting to joint-code data. At the narrowest level of focus, only optimisation of the index assignment is considered. These techniques are categorised as IA adjustment, which strives at optimising the index assignment operation described above in order to make it robust against channel errors. Examples of these techniques are BSA (binary switching algorithm), where permutations of a given codebook are examined to see whether they yield lower overall distortion, and SAA (simulated annealing) where controlled randomness is included in the process of refining the index assignment in order to try and reach a global distortion minimum instead of getting stuck in a local minimum.

A broader focus considers joint optimisation of IA and quantisation. This is also known as “zero-redundancy quantisation”, in the sense that data that would be allocated for error protection is rather used to refine the quantiser. Examples include COSQ (channel optimised scalar quantisation) and its vectorial counterpart COVQ (channel optimised vectorial quantisation), which are generalisations of the Lloyd-Max algorithm – respectively the Linde-Buzo-Gray algorithm in the vectorial case – to noisy channels. Another example is the self-organizing hypercube, SOHC, referred to as self-organizing map, SOM, in the scalar case.

Finally, the broadest focus aims at joint optimisation of IA, quantisation and modulation. An example of this is the MORVQ (modulation organized vector quantisation) algorithm, where the signal space is directly mapped onto a two-dimensional modulation space by using the Kohonen algorithm.

Chapter 2 A real-life application

Given the conditions of the project, I had to decide, jointly with Fernando, the exchange student, and Arturo Veloz and Jean-Marc Boucher, the two professors responsible for the project, on a “close-to-real-life” application. In particular, the thesis work of Seyed Zahir Azami [4] had already studied joint coding techniques on a binary channel.

After exchange of point of views on the subject, and taking into account the limited time resource allocated to the project, as well as the tools and support available we decided to use the vocoder developed by Fernando as our source encoder, and a Rayleigh fading channel. Furthermore, we would use the CDMA IS-95A standard for modulation purpose over the fading channel, as it was available for use in Matlab.

2.1 The TCENLP vocoder

The TCENLP vocoder is a non-linear analysis-by-synthesis vocoder. As developed by Fernando Villavicencio for his MSc thesis, it is the implementation of the architecture proposed by L. Wu, M. Niranjan and F. Fallside in a paper published in 1994 in the IEEE transactions on speech and audio processing [9].

2.1.1 Principles of the TCENLP vocoder

Consistently with common linear prediction coders (such as the DoD’s LPC-10) and the source-filter model above (see §1.1.2), the vocoder first aims at isolating the short-term modulation of the speech. This modulation is the one induced by the vocal tract, running from the vocal cords to the lips. By non-linearly filtering out this modulation, a residual is obtained. This is called the short-term prediction (STP) analysis of the signal. The second step is then to analyse the residual and find out whether it has pseudo-periodic characteristics. If it does, in case of a voiced sound, the vocoder proceeds to extract the base frequency, or pitch, of the signal; if not, in case of a non-voiced sound, the residual will already be pretty much white noisy. In any case, after analysis, and possible filtering out, of the pseudo-periodic characteristics, a noisy – i.e. carrying no information – residual is obtained. This second step is the long-term prediction (LTP) analysis of the signal.

The final step of the encoding is to represent in some manner the last residual. As the coder's name indicates, this is done by choosing a "representative" residual from a codebook and transmitting the corresponding code. The codebook is created at the time of design, and both the encoder and the decoder share the same one. Choosing a different – but close enough – residual instead of the one obtained after filtering of course introduces some measure of distortion. The aim is to minimize the distortion while keeping the codebook size small enough that transmission of the representative code does not degrade too much the performance – in terms of bit rate – of the encoder.

Now, considering that the encoder is also designed on an analysis-by-synthesis base, the steps described above take place in a closed loop, as described in Figure 2-1: after the analysis described above, the input signal is re-synthesised from the parameters obtained and compared to the original. The general idea of the analysis-by-synthesis design is to minimize the difference between the original signal and the synthesised signal according to some metrics. In general, it is possible to know, based on the results of the comparison, how to tweak the analysis parameters to obtain better results.

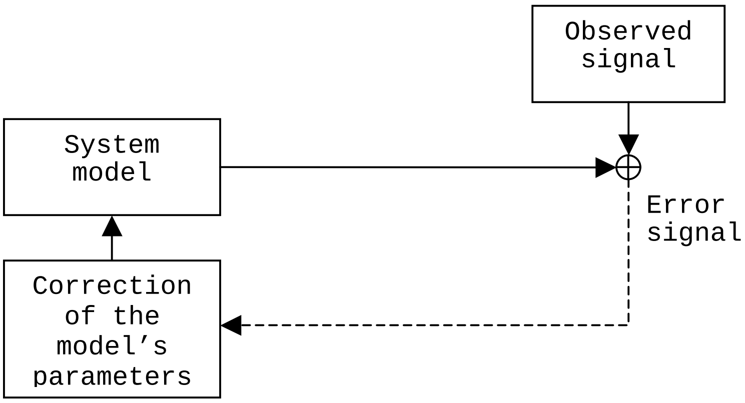

Figure 2-1 Basic concept of analysis-by-synthesis (from [8])

More precisely, the one parameter on which to play, in the case of a code-excited coding scheme is the choice of the representative residual from the codebook. In coders where linear filtering occurs, it is possible to organize the codebook, so that it is a partition of the signal space, where each entry of the codebook represents a cell. This supposes that a topology can be established on the signal space, which is usually possible. The algorithms used to establish the partition are based on the original Lloyd-Max algorithm [4].

However, in the case of the TCENLP, the non-linearity of the involved filters makes it impossible to establish a partition of the residual space. As a consequence, this specific encoder has to exhaustively consider all entries of the codebook, in order to determine the one that, at synthesis time, produces the minimal distortion with the original signal. The need for an exhaustive search makes the encoder extremely expensive in terms of required processing power.

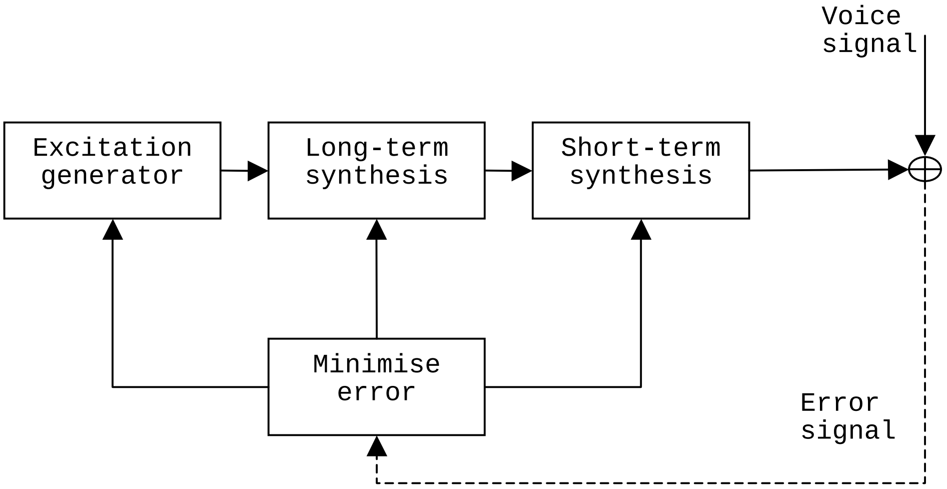

Figure 2-2 Analysis-by-synthesis in a source-filter model (from [8])

2.1.2 Quantitative data of the TCENLP vocoder

Finally, it is worth noting, as we will use it later on, the specific data output from the encoder, as developed by Fernando Villavicencio. The encoder processes frames of speech data sampled at 8kHz, each consisting in 64 samples, thus presented at one frame per 8ms. Every operation of the source-filter model analysis is operated on each frame, with the exception of the short-term analysis that needs to consider two frames at once, and thus is performed only once every two frames. Thus, the encoder will output an encoded frame every two input frames, that is every 16ms.

Short-term analysis is carried every two frames. It results in choosing an adapted non-linear (neural network) predictor from a predefined book of predictors. The codebook holding 64 entries, this data is transmitted as 6 bits values.

Long-term analysis is carried every input frame. As for the short-term analysis, this analysis yields to the choice of an adapted non-linear predictor from a fixed book of predictors[3]. The book holding 64 entries, the index is transmitted as a 6 bits value. This analysis also extracts the pitch value for the input frame, which is transmitted as an 8 bits value. Thus, in an output frame, long-term analysis requires twice the sum of the above, i.e. 28 bits.

As we said, after both short- and long-term analysis, the input data has been filtered into almost-white noise. In order to choose an adapted substitute for that residual from the excitation codebook, the residual is first normalized (e.g. to a unity power). The normalizing gain is evaluated for every input frame, and transmitted as a 5 bits value. Finally, the index representing the chosen codebook entry is transmitted as 6 bits value (indicating a codebook with a 64 entries). Overall, this last step requires 22 bits per output frame.

Thus, output frames count with 56 bits, yielding an output rate of 3.5kbps, a 94.5% improvement over the 64kpbs of the standard PCM rate[4].

2.2 The CDMA IS-95A standard

2.2.1 Rationale for the choice of the standard

The CDMA IS-95A standard is one of the North American mobile phone standards, based on a CDMA (Code Division Multiple Access) scheme. While not necessarily common at a worldly scale, and while not necessarily granted a long future, this is a sound, real-world standard.

As a mobile phone data transmission standard, it is designed to efficiently transmit voice data over multipath fading (Rice and Rayleigh) channels. Furthermore, an implementation of the standard is available in Matlab’s Simulink. Since Fernando was developing his encoder on Matlab, which I was eventually supposed to use in a simulation and since we wanted a real-world fading-channel-able modulation technology, it appeared that despite its possible lack of future and acceptance worldwide, the standard was a good choice for the project.

[3] Note that how this book is obtained is also part of the design of the TCENLP encoder, though not relevant in this dissertation. Suffice to say that a training of the encoder takes place prior to use.

[4] Please note that the values exposed in this section do differ from the values presented by F. Villavicencio in [8]. The reason for this apparent incoherence is that my work was based on Fernando’s encoder as it was before the tuning that took place for the redaction of his thesis.

2.2.2 Overview of CDMA

CDMA (Code Division Multiple Access) is a modulation technique based on spectrum spreading. By carefully operating the spreading, it also allows multiple-access to a channel, ensuring that data belonging to different users is pseudo-orthogonal to each other.

An important feature of CDMA schemes is that signals sent over the same physical channel (though in different logical channel) will appear mixed both in the time and frequency domains. Separation of the different logical channels is achieved at the receiver by correlating the received signal with the spreading pseudo-random sequence that the logical channel of interest was encoded with. The property of pseudo-orthogonality ensures that only the signal of interest will be restored, while all other signals will only contribute to noise, as they are not de-spread in bandwidth. Filtering with a narrow band-pass filter, centred on the decoded signal allows further reduction of the noise contributed by all the other signals.

Thus channel assignment is determined by creating adequate sets of pseudo-orthogonal, pseudo-random sequences. The sampling frequency of the sequences, also known as the chip rate, is chosen so that the bandwidth of the spread signal is several times that of the original signal. Assuming that synchronization is established between an emitter (of several channels, each spread by its own set of pseudo-random sequences) and a receiver, the receiver will be tuned on one logical channel by using the corresponding spreading sequence, in sync with the emitter, to de-spread the received signal.

One of the positive points of spread-spectrum modulation is that it is naturally resistant to multipath fading, since each path can be demodulated independently from the other. However, this is not perfect and there are still cases when fading affects the demodulated signal.

In CDMA systems, the bandwidth of the spreading sequence is chosen to be much greater than that of the transmitted signal, so that the bandwidth of the spread signal is roughly that of the spreading sequence. Thus the cross-correlation properties of the modulated signals are rather equivalent to those of the spreading sequences. As such, the condition of pseudo-orthogonality, which is equivalent to good cross-correlation properties, i.e. cross-correlation of two signals yields a large bandwidth, low energy, preferably noise-like signal, is implemented on the spreading sequences, independently of the transmitted signals, which greatly simplifies the design of such systems.

2.2.3 Fundamentals of IS-95A

The IS-95A standard, as a CDMA system, naturally implements the properties above. Being a standard for the industry, is also implements more features, in order to ensure data integrity, and adapts the CDMA principles to the particular case of mobile phone communication, where there is a pronounced dissymmetry between the two sides of the communication link. While the base station suffer virtually no energy restriction, and thus can emit several channels at high powers, the mobile phones live on a small battery and have to restrict themselves to managing only the channels they are concerned with, at the lowest manageable output power.

The two following diagrams show respectively the forward and reverse channel diagrams (respectively from the base station to the mobile phone and vice versa).

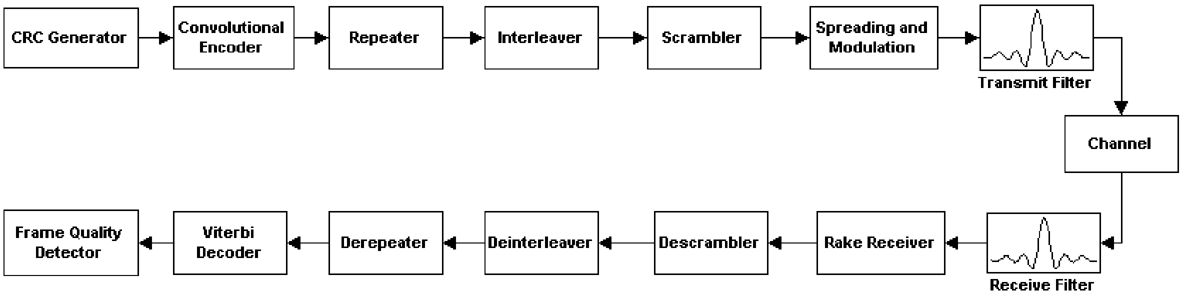

Figure 2-3 IS-95A forward channel diagram

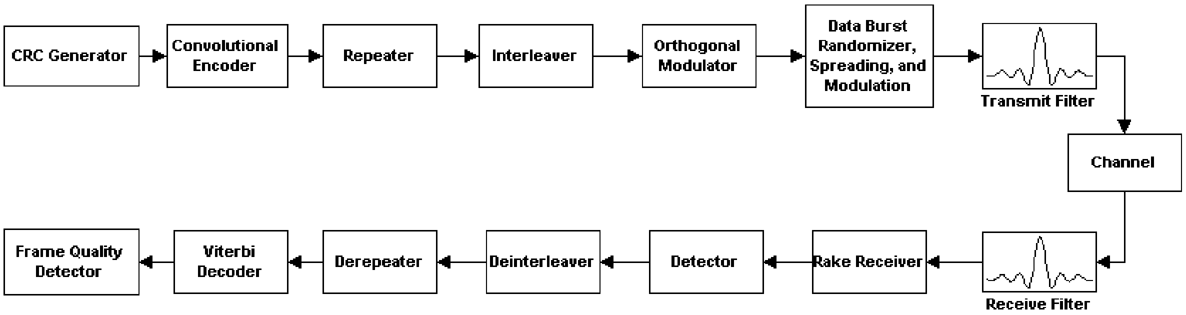

Figure 2-4 IS-95A reverse channel diagram

We can see that besides the spreading and modulation steps belonging to the CDMA scheme as described above, there is also several steps of channel coding (that is error protection) taking place. However, as we will see later, in our case, we are interested in providing our own channel coding scheme, so that we do not use the channel coding facilities provided by the standard. On the diagrams above, we drop the data path running from the

CRC generator to the scrambler (in the forward channel) or the interleaver (in the reverse channel), and the corresponding path on the receiver side.

The IS-95A standard specifically implements the CDMA scheme using the following characteristics. First of all, the pseudo-random sequences bandwidth is chosen so that after modulation, the signal has a bandwidth of 1.25 MHz, which corresponds to one tenth of the total band of frequencies allocated to a mobile operator.

Furthermore, the standard uses two sets of pseudo-random codes. The so-called short PN codes are a pair of sequences, of length 2^{15} symbols, used to modulate the signal into in-phase and quadrature components. Note that different base stations operating at the same frequency can use the same sequences by simply using different offsets into the sequences. The long PN code, of length 2^{42}-1 symbols, is used for spreading, scrambling and power control on the reverse link.

In addition to these codes, the standard also uses a set of mutually orthogonal codes, called the Walsh codes, which will ensure the mutual-orthogonality property central to the CDMA scheme.

In addition to the traffic channels, which carry the data, the standard defines a few other (logical) channels for the proper operation of the system: on the forward link, the pilot and sync channels provide channel estimation and synchronization, thus allowing correct demodulation of the traffic channels. On the forward and reverse link, a channel is dedicated to control information, spontaneous requests from the mobile station and paging.

The base station simultaneously transmits all the traffic channels, plus the three control channels (pilot, sync and paging). To allow individual decoding by the receiving mobile station, each traffic channels is spread by its own Walsh code sequence. Furthermore, as in all mobile communication system, each traffic channel is granted its own power, in order to optimise the overall transmitted power. In the CDMA system, this is ensured by adding the spread traffic channels together after correction by a factor corresponding to the intended transmitted power. Note that the factor is determined with respect to the intended transmitted power per bit. In case a traffic channel operates at a lower bit rate, bits are repeated in order to obtain the same transmitted bit rate, but the correcting factor applied to the channel is modified accordingly.

Finally, both links use a rake receiver to acquire the transmitted data. The aim of the rake is to efficiently manage the effects of multipath fading during transmission. While standard receivers would greatly suffer from the fading incurred, rake receivers are able to individually isolate each path components of a signal and recombine them. On the forward link, the pilot channel allows for coherent detection in the (mobile) receiver, while on the reverse link, the absence of such channel requires the (base) receiver to implement a non-coherent detection scheme.

2.3 *Our home-made channel coding attempt*

Given our source coder, the TCENLP vocoder, and our channel, beginning at the spreading and modulation stages of the IS-95A CDMA standard, what was left to do was implement our attempt at a channel encoder. Based on the information, mainly gathered from the Zahir Azami thesis [4], we decided we could have a hopefully successful try at applying a simple index assignment optimisation on the codebooks prepared in the source encoder.

On the other hand, the vocoder was developed by my colleague Fernando Villavicencio in the framework of his own master’s thesis, so that he was not keen on including external work into his implementation. As a consequence, we decided to develop the “joint” channel encoder as a separate component, which admittedly seems to contradict the joint approach.

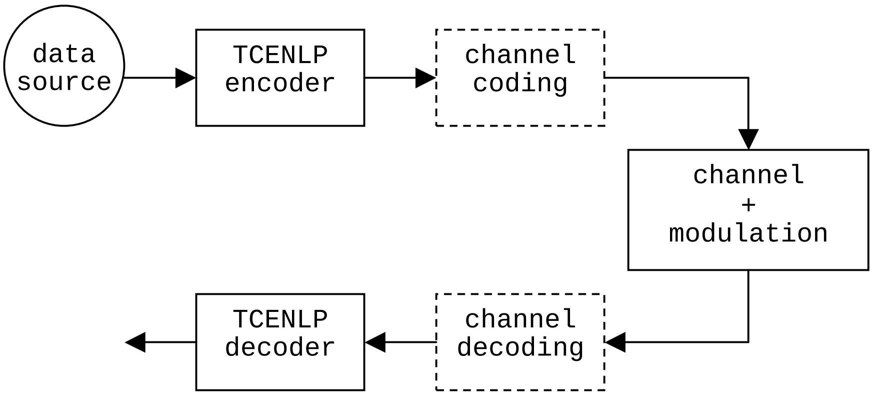

Figure 2-5 Joint “disjoint” channel coding

The vocoder, as implemented by Fernando, outputs the following data every two input frames:

- two excitation indexes, indicating the best choice excitation vector for each input frame,

- two LTP indexes, indicating the best choice set of parameters for the non-linear long term prediction filter for each frame,
- one STP index, indicating the best choice set of parameters for the short term prediction filter for the double frame made of both frames,
- two pitch values, as extracted by the short term predictor, one for each frame, and
- two gain values, as used in order to normalize each frame.

We decided to treat each of these parameters separately, according to their nature. The values (gain and pitch) could be translated into a representation that would be more robust than the binary representation against channel errors.

On the other hand, the indexes (excitation, LTP and STP) could be reassigned in order to minimize data distortion in case of index error. In other words, given a measure on the data space (e.g. in the case of the excitation codebook, an Euclidian measure on the vectors of samples), and a measure on the index space, ideally adapted to take into account the errors introduced during transmission over the channel, we want to try and give both spaces the same topology, so that two data vectors that are close in the data space, according to the measure defined there, would be represented by indexes close in the index space, according to the measure defined there.

Once the reassignment is computed, we can apply it out of the vocoder itself, in a disjoint component. Thus, the apparently disjoint channel encoder does perform a joint coding operation.

Given the choice of this approach, we have several problems to solve in order to apply it. We need of course to determine an adapted measure on the different data spaces. While it might seem trivial that the Euclidian measure is plainly adapted to scalar or vectorial data such as the pitch and gains values, or the excitation vectors, we might have more trouble defining a measure on the STP or LTP filter parameters, which are no more than an arbitrarily ordered set of parameters for the non-linear short- and long-term prediction filters.

On the other side of the channel encoder, we also need to determine an adequate choice of a measure on the index spaces. As we said, this measure should be determined in order to take into account the distortion introduced by the channel, so that the index reassignment is channel-optimised. This determination should at the very least require a statistical analysis of the channel, in order to try and determine statistical properties of the errors introduced upon transmission.

Chapter 3 Implementation and results

As I stated earlier, both professors in charge of the project, Jean-Marc Boucher at ENST de Bretagne and Arturo Veloz at Cinvestav, as well as both students, Fernando Villavicencio at Cinvestav and myself decided on the components that would surround my work on joint coding, namely Fernando's implementation of the TCENLP vocoder described in [9] as a source coder, and the IS-95A CDMA standard as binary-channel-over-fading channel. The reason for the choice of the source coder was that, given its design, the TCENLP already had interesting joint-coding features. On the other hand, the channel standard was chosen, because it came from the "real-world", it corresponded to the requirements stated in the project outline [A.2] and we had an available implementation on Matlab.

Unfortunately implementation of a complete, running system took most of the time allocated to the project, so that while implementation details are numerous, results are few. This is most likely due to the following conjunction of parameters: first of all, as time passed, it appeared that the project definition might have been too ambitious. Furthermore, while the project started rather late, towards the end of April, I had to get to know the required points of theory. Finally, the broad openness of the project allowed for much options to be explored, while it might have proven wiser to restrict the focus of the project to some particular aspect.

As we said, implementation of a complete system, as shown in Figure 2-5, took place on Matlab, and more precisely on the Simulink tool shipped with the Release 12 of Matlab. As it is described in its own documentation, Simulink is "a software package for modelling, simulating, and analysing dynamical systems". In particular, it allows for model description via a very intuitive graphical user interface, thus making it a very practical tool for developing, and later simulating, models.

We shall examine in turn implementation choices and details for the channel, our source of data, our attempt at joint (channel) coding and our data analysis tools.

3.1 IS-95A and fading channels

The project outline states that we want to conduct our study of joint channel coding for speech signals on fading channels. While Simulink provides such channels, and it is easy to set up a system integrating such channel, standard modulation techniques do not allow reasonable error rates upon transmission over such a channel. As an example, let us consider a system where a source of modulated signals emits a constant value of imaginary 1. After transmission through a fading channel including both a line-of-sight transmission – with Rician characteristics – and a multipath transmission – with Rayleigh characteristics – the signal is plotted on the complex plane.

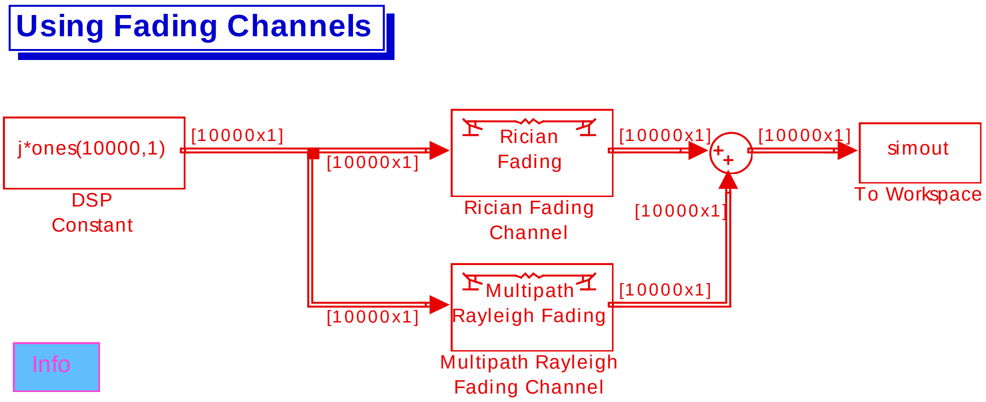

Figure 3-1 A fading channel example from Simulink

The system is represented in Figure 3-1 above, as it would appear in Simulink. While most details are not relevant to this dissertation, we can appreciate the clarity of the diagram. Of course, this is only a simplistic example, but the user interface to Simulink is a great help when dealing with more complex models. The "Rician Fading" and "Multipath Rayleigh Fading" blocks appearing on the diagram are also built-in in Simulink.

On Figure 3-2 below, the output of the combination of the fading channels appears. The red dot is the input signal, while the blue curve is the output signal. An immediate remark is that, obviously, a great deal of distortion occurs. Now, to grasp how much of a hindrance this is, imagine a binary phase-shift keying modulation (BPSK) scheme. The signal emitted by a BPSK modulator, would be represented on the graph below as the red dot and its symmetrical with respect to the abscissa line (i.e. values ±j). A simple decoding scheme is to observe the received signal's imaginary part sign, deciding according to its sign what was the value of the signal emitted. This is basically what the BPSK demodulator does. However, with the distortion caused by the fading channels, as represented by the blue curve below, this analysis is not possible anymore. Such a simplistic scheme would statistically always fail to detect the right input signal, thus leading to a 50% bit error rate (BER) between the (binary) input and output of the system.

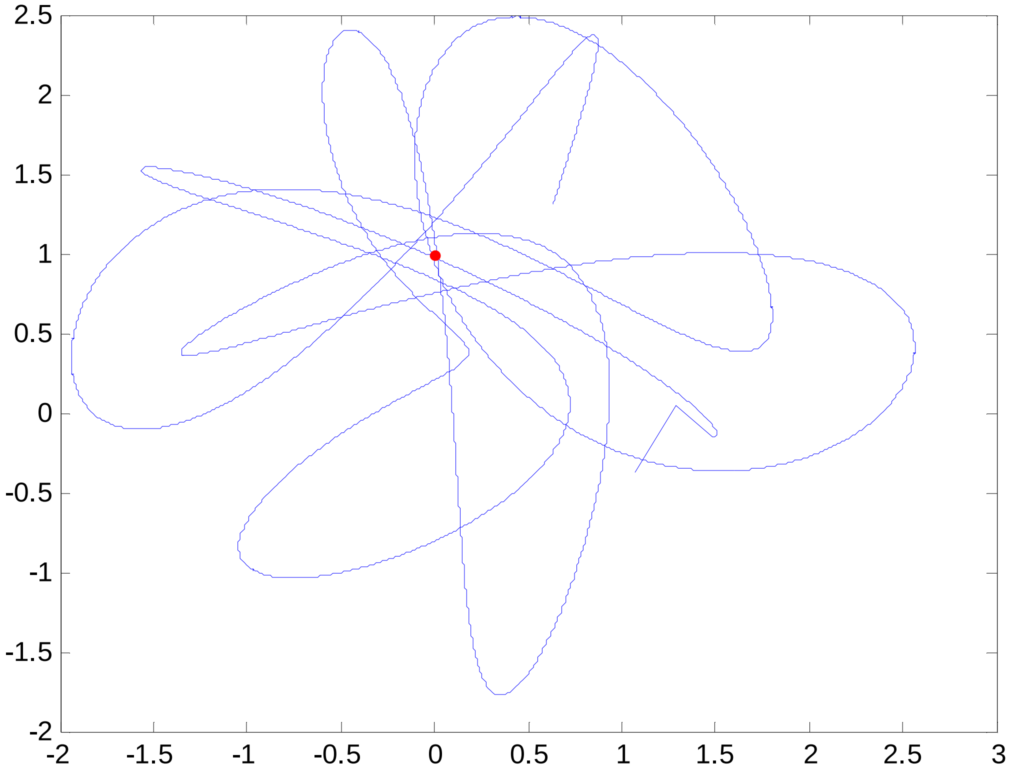

Figure 3-2 Effect of a fading channel over a constant signal

For this very reason, we decided that we needed a modulation scheme for our fading channels that would resist their degrading effects. Since we did not consider it worth to develop such a model during the project (even if develop was only taken as meaning implement an existing specified system), we decided to use the available "real-world" system that came with Simulink: an implementation of the North-American CDMA mobile telephony standard IS-95A, along with examples of full communication channels. Although the standard specifies fully geared channel coding system, along with modulation, we decided that, since our project was about joint source-channel coding, it would not make much sense to use the provided channel-coding scheme. In the end, we only kept the modulation and spreading components of the IS-95A standard, operating over an analogue channel with Rician and

Rayleigh Fading properties, as well as additive white gaussian noise (AWGN). This set of components shall now be referred to as simply “our channel”[5].

Our channel is thus a binary channel, for it admits binary data as an input, and outputs binary data. The efficiency of the spreading modulation in IS-95A, as mentioned in 2.2.2 allows for rather satisfying results for the properties of our channel. Indeed, analysis shows that the BER generated by the channel is slightly less than 2%. Of course, one might argue that this is still a rather large value, considering the field of application, but compared to the results given by simple – though robust – modulation schemes, such as BPSK, it is a quite satisfying result.

As stated in 2.3, we also would need an analysis of the statistical properties of the errors introduced by our channel, in order to design adapted measures on the data transmitted over it. While I was able to gather data about the channel, in particular about the repartition of errors in time, I did not have time to either analyse it, or put it to use into determining an *ad hoc* measure.

3.2 The TCENLP vocoder

Unfortunately, during the development of our project, the TCENLP vocoder was not completed yet. As such, it was not possible, though it was developed under Matlab as well, to integrate it in out project. Since we decided anyways to use it for the project, if only to let Fernando re-use the project as a base for an eventual PhD, we had no choice but develop an emulator, a stub in place of the original vocoder.

For that purpose, Fernando provided me with a model of the output data, as we should expect it. As we said earlier, the model specifies output frames of 56 bits, every 16ms. In order to correctly manage the data along the communication line, I had to arbitrarily choose a layout for the data within a frame, which came as described in Table 1 below. In the specification came also statistical information about each field in the output frame, in order to simulate more closely the true encoder. Note however, that this information is at best

[5] Note by the way that S. Zahir Azami, in [4] refers to the ambiguity of what we call a channel, which, depending on the authors, varies from just an analogue channel, to an analogue channel plus a modulation step, and even sometimes a channel coding step.

guesswork, and does not come from an analysis of the output of the true encoder, which, again, was not ready for that.

Bits	Field	Description
0-5	STP	The index of the chosen non-linear short-term predictor for both input frames corresponding to this output frame
6-13	Pitch1	The pitch value found for the first input frame corresponding to this output frame.
14-19	Exc1	The index of the excitation chosen from the excitation codebook for the first input frame corresponding to this output frame.
20-25	LTP1	The index of the chosen non-linear long-term predictor for the first input frame corresponding to this output frame.
26-30	Gain1	The gain computed for the residual after STP and LTP filtering for the first input frame corresponding to this output frame.
31-38	Pitch2	Idem Pitch1 for the second input frame that produced this output frame.
39-44	Exc2	Idem Exc1 for the second input frame that produced this output frame.
45-50	LTP2	Idem LTP1 for the second input frame that produced this output frame.
51-56	Gain2	Idem Gain1 for the second input frame that produced this output frame.

Table 1 Description of the TCENLP emulator output frame.

While the determination of scalar values, such as the gain or pitch values, was relatively easy, it proved more difficult to set up a satisfying scheme concerning the codebook index values, namely the STP, LTP and Exc fields. We decided that the gain would cover its full range uniformly. On the other hand, a normal distribution would be used to determine if a pitch field would be 0 or not. A null value would indicate a non-voiced segment of speech, while a non-null value would indicate a voiced segment. In case a voiced segment was determined, its pitch value would be computed with a normal distribution of mean 50 and variance 10, which corresponds more or less to the expected variation of the pitch in speech data produced by an adult male. In the statistically rare case a pitch value would be out of range (since normal distributions are not bounded), clipping would occur. Note that we decided to represent the pitch values as an 8-bit unsigned integer, thus giving us a range of 0-128. On the other hand, the parameters of the distribution ensure that most of the chosen values will be in the range 40-60, so that we are admittedly "wasting" some output bandwidth. We want to remember though that this is only a crude first-try approach, and was most likely refined in the final version of the true vocoder.

For the index fields, STP, LTP and excitation, the idea was that the distribution of each would likely be normal: while a reduced set of indexes would represent most of the input data presented, the rest would represent less likely, and more specific cases. This observation is rather consistent with several systems we trained ourselves on, in particular the linear prediction coder variant of the US Department of Defense's LPC-10.

However, we also wanted to take into account that the indexing was arbitrary for every step. In other words, Fernando's implementation of the TCENLP did not guarantee that the indexes most likely to be picked up would be clustered together. The idea for the emulator was thus to choose the indexes according to a normal distribution, but only after having operated a permutation on them.

Finally, in order to take full advantage of the features offered by Simulink, we decided to implement the emulator in C, in the framework of the development of Simulink components. However, C is not well furnished in mathematical functions. According to [5], the only statistical function available in C is the uniform distribution, and even its quality is not always quite satisfying. For that reason, we decided that the statistical part of our implementation, in other words, the determination of the different required distributions and permutations, should be obtained via interfacing with Matlab through its C API.

3.3 Interfacing the components

Once the channel, as understood in its largest form, and the source of data were ready, the only thing left to implement was logically enough the disjoint attempt at joint coding (as explained in 2.3). However, there was first another issue to solve. Indeed, our decisions and choices on the different components of the project led us to have to try and interface two components with seemingly incompatible data flows.

The TCENLP vocoder, as we pointed out earlier, outputs 56 bits frames every 16ms, amounting to a bit rate of 3.5kbps. On the other hand, the IS-95A standard specifications require 576 bits frames every 20ms, amounting to a total bit rate of 28.8kbps. While we could certainly have a 28.8kbps flow carry the data of a 3.5 kbps flow (not considering for this project the obvious waste of bandwidth incurred), we had to devise a way to practically achieve it. In particular, we had to satisfy constraints of regularity of flows: more precisely, the flow of data out of the channel, to the TCENLP decoder had to have the same properties of 56 bits frames every 16 ms as the flow out of the encoder. Reasons include that the decoder, if and when ready, would require to work on the fly on a steady flow of data, as well as out ability to compare input and output data given the flow architecture in Simulink.

Thus, the requirements for a "rate adaptation" component are the following, using the symbols on Figure 3-3:

1) The same component must be able to adapt flow 1 to flow 2 **and** flow 2 to flow 1.

2) The component accepts and outputs data in frames (even of one bit length only).

3) The component must be able to admit any frame size at any rate for both flows.

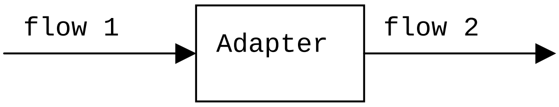

Figure 3-3 Bit flow adapter

Furthermore, the component shall comply to the following requirements, ensuring, in the case of our project, the integrity of the data coming to the last decoding stages. Referring to Figure 3-4 and assuming that the bit rate of flow 2 is greater than that of flow 1[6], the requirements come as follow:

4) The data in flow 1' must be, not withstanding a possible delay, the same as the data in flow 1. In other words, the black box must act as nothing more than a delay component.

[6] This assumption just reflects the fact that if the bit rate of flow 2 in Figure 3-4 was greater than that of flow 1, passage through the adapter would drop some of the data of flow 1, and neither of the following requirements could be respected.

5) The frames of flow 1’, though maybe delayed with respect to those of flow 1, must contain individually the same data. In other words, the flow of bits must not lose its synchronisation with the flow of frames.

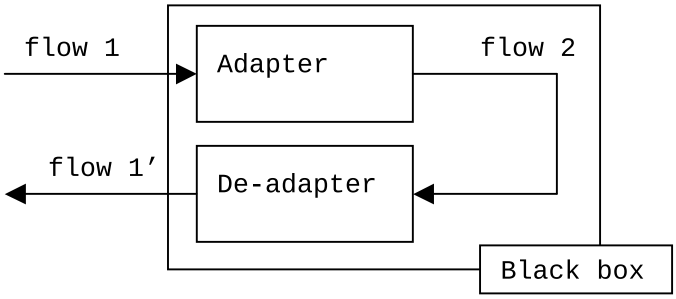

Figure 3-4 Bit flow adapter and “de-adapter”

As a consequence of the first requirement, the component will have two working modes: if the bit rate of the input flow is less than that of the output flow, it will manage to transfer all the data in the input flow to the output flow, adding padding as necessary. On the other hand, in the opposite situation, the component will select some of the data from the input flow, and transfer it in the output flow, without adding any padding. Because of Requirement 4), of course, the data dropped in the second case shall exactly correspond to the padding added in the first case, so that the only data going untouched through the system described in Figure 3-4 is the original data of flow 1.

Thus, the component mixes abilities of both a delay line and a bit selector – be the selection performed to choose where to include padding, or on the contrary to know what data to drop. To satisfy requirement 5) and ensure synchronisation of data and frames is only a matter of carefully choosing the size of the delay lines at de-adaptation. The component himself has no way to determine this factor by itself, but it is a constant of a system – in other words, in our project at least, flow 2 incurs a known delay – and can thus be passed to the component as a parameter.

As a consequence of the above requirements, the bit selection algorithm has to ensure that it will work symmetrically: the bits selected to receive padding in the output frame at adaptation stage must be the exact same bits that are dropped from the input frame at de-adaptation stage. The algorithm I came up with goes along the following lines: along the life of the component, the input and output bit rates are given and constant. Thus, there is a constant factor of proportionality `rate` between them. Considering the adaptation stage, for example, the idea is then simply to put each of the input bits every `rate` output bits. Of course, `rate` will generally not be an integer constant. Thus, also maintaining a real offset

`offset` into the output frame, we will place the input bits at positions floor of `offset + n * rate` for all positive integer `n` such that `offset + n * rate` is within the output frame. The offset is then re-updated to the value of the first position out of the output frame, modulo the output frame size.

In C, this algorithm translates into a rather simple `for` loop. Given a bit mask `mask`, whose size `sz` is the size of the frame in which the selection will take place – the output frame at adaptation stage, the input frame at de-adaptation stage –, given the offset `offset` mentioned above, and given the coefficient `rate` defined as the ratio of the smaller bit rate over the larger bit rate, the loop that will select the adequate bits within the mask marking them as 1 is the following[7]:

```
double ptr;

for (ptr=offset; ptr<sz; ptr+=1/rate)
{
    mask[(int)floor(ptr)] = 1;
}
offset=ptr-sz;
```

Figure 3-5 the core of the frame adaptation algorithm

Once implemented in C, within the Simulink framework (for further detail, refer to [10]), the component was integrated in our model, as being part of the channel. Thus, we could consider our model as integrating a source of data (the TCENLP vocoder emulator) and a channel with compatible data formats.

3.4 The channel part of the joint coding

Mainly based on ideas gathered in S. Zahir Azami's thesis [4], I decided to try and implement an added protection on the data output by the TCENLP vocoder. As we pointed out earlier, the vocoder contains three codebooks. Two of those assign an index to a set of parameters for non-linear filters, the short- and long- term predictors. The third codebook assigns an index to the residual of the signal after filtering through both predicting filters. According to [4], we can consider both codebook values and keys as elements of vectorial

[7] Note that the `floor` function used in the algorithm is part of the ISO C standard, is also described for reference in [5] and that it does what it is expected to do, i.e. return the floor value of the passed parameter, that is the largest integer value less than the parameter.

spaces. In particular, in the case of the excitation codebook, the residuals are vectors of audio data, naturally part of vectorial spaces with as many dimensions as they contain samples.

By choosing appropriate measures – or distances – on these vectorial spaces, we can give them a topology. The "joint coding idea" is thus to manage to give, after re-organisation, the same topology to both spaces of data. In other words, we are trying to achieve that two close members of the data space are represented by two close members of the index space. Thus, error introduced during transmission over a noisy channel on the indexes will not get even more amplified in the data space.

However, at that point, much of the project's allocated time had been spent setting up other parts of our system, and I did not have much time to spend on this matter. To successfully apply this algorithm to the different codebooks of the TCENLP required that I define some sort of measure on the data. For the short- and long- term predictors, the members of the codebooks were sets of non-linear filter – neural network – parameters, so that defining a measure on it did not seem any easy task.

Thus, to try and obtain results before the end of the project, I decided, after consultation with Fernando and Arturo, to try and implement a similar idea on the scalar data we had: the gain and pitch fields of each frame (see Table 1). In Fernando's implementation, those fields were simply encoded in binary form. However, when transmitted over the channel we set up, which introduced binary error, this data would be mangled. We thus thought that because the error was introduced in binary form, maybe the binary representation was not the most adapted to the channel. Given the property of the Gray codes that two adjacent codes are Hamming-distant by only one unit, we thought that maybe Gray code would prove more adapted to resist the introduction of errors by the channel, as far as the decoded data was concerned, than the original encoding.

The other problem we had to solve was designing adequate tools to evaluate the impact of our attempt. The ideal solution would have been to compare original and restored (after vocoding, transmission over the noisy channel and decoding) speech samples. This was unfortunately not a possibility, since we did not have a functional vocoder, but rather an emulator of it. Thus, given our system, we could only directly compare the data of interest, the gain and pitch values. Assuming that these would have a rather direct influence on the decoded speech, an adapted measure would simply be the Euclidian measure. By comparing the distance of the data of interest after transmission through the fading channel to the same data before transmission, both with and without our encoding scheme, we could get a rough idea of whether this was a viable idea or not.

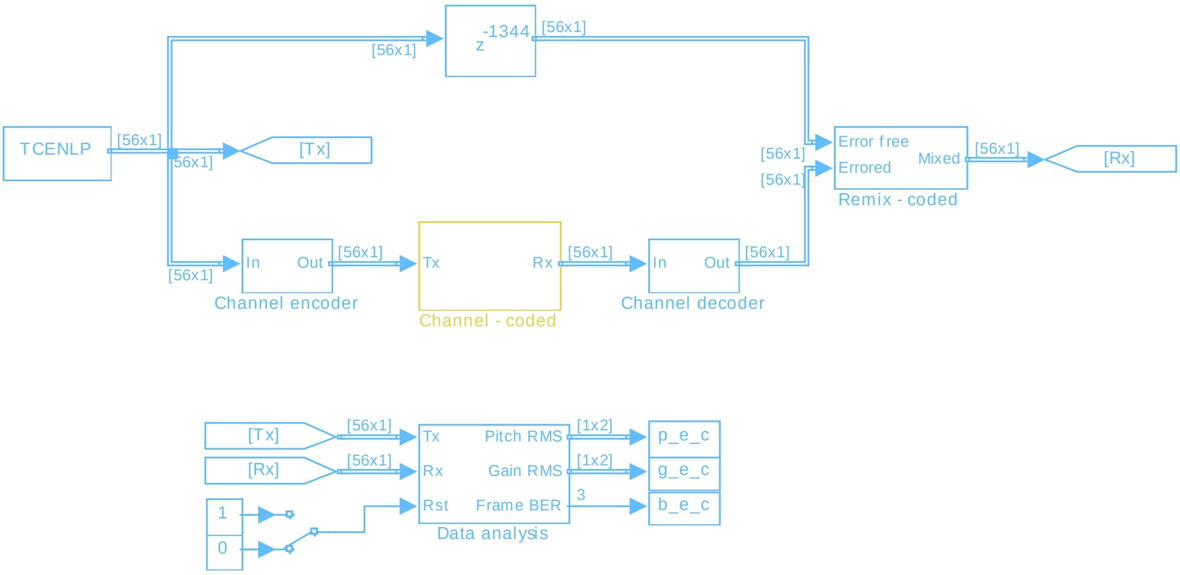

Figure 3-6 The channel coding testing model

For that purpose, we created the system in Figure 3-6 above. The TCENLP block holds the emulator for the vocoder, as described in 3.2. The "Channel – coded" block holds the fading channel along with the modulating part of the IS-95A standard, as described in 3.1 and the frame adaptation component described in 3.3. As we can see, the data out of the vocoder travels along two different paths. The top one, is merely a delay path, and leaves the data untouched, save for the delay, equal to the one introduced by the bottom path. The bottom path, on the other hand performs the data encoding and decoding (for the gain and pitch fields only), and processes the encoded data through the channel. Both paths finally join in a mixing component that regroups data from both paths, keeping from the bottom path only the gain and pitch fields, and all the other fields from the top path. Thus, the data reaching the Rx label sees only the gain and pitch fields altered, while all the others remain untouched. While this does not bring much in our configuration, it would allow isolating the effects of the changed fields in the TCENLP decoder, when we would have it running.

The channel coding blocks perform the conversion from binary representation to Gray representation for the pitch and gain fields of the frame, and back. These operations did not require development of Simulink components in C, but rather could be achieved using the available base blocks: selector blocks isolate the fields to be converted, which are then processed through an adequately configured data-mapping component. The isolated fields are finally re-concatenated together.

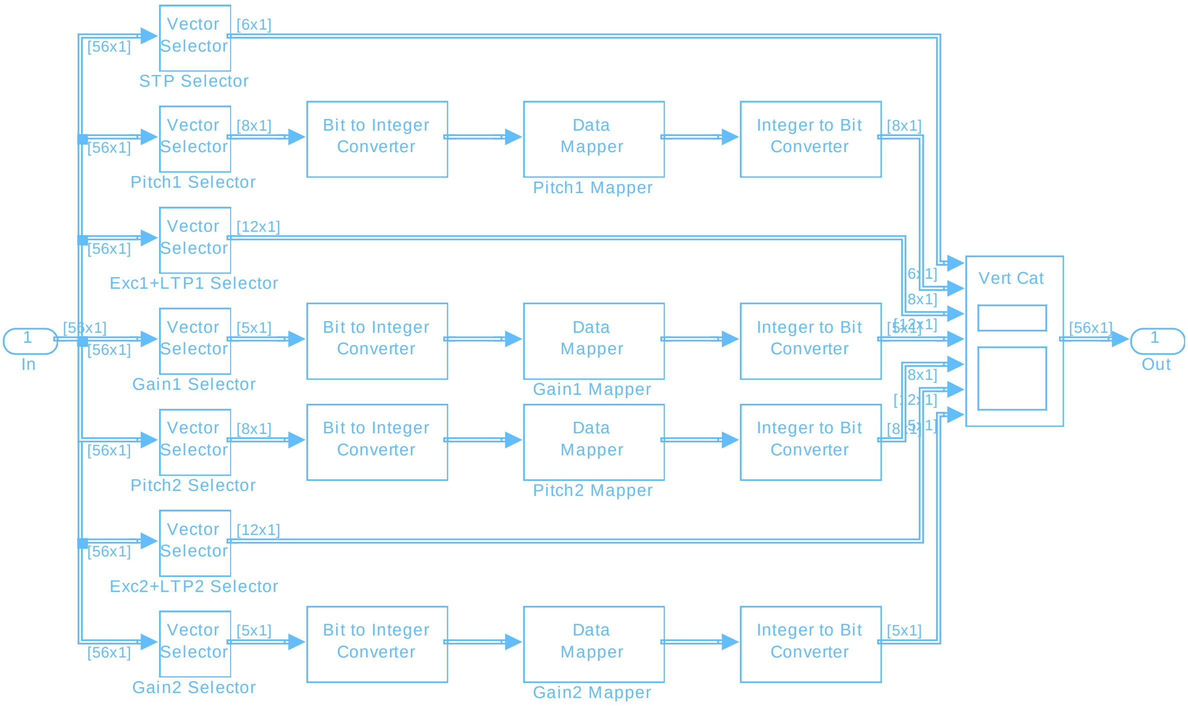

Figure 3-7 The channel coding block

Finally, the data analysis block compares the transmitted and received data, in terms of bit error rate for the whole frame – which is in turn representative of the bit error rate for the fields of interest, since the others are transmitted error-free – and mean Euclidian distance between the transmitted and received fields.

For the data to be meaningful, we developed three versions of the system. The first version has the physical channels (within our big channel block) configured to introduce no error. This is to test that the system is correct and does not introduce errors by design. The second and third version both carry an operational channel component that introduces error as shown in 3.1. However, while the third version performs the channel coding scheme that we came up with, the second version performs no channel coding. This is to try and determine the effects of our scheme.

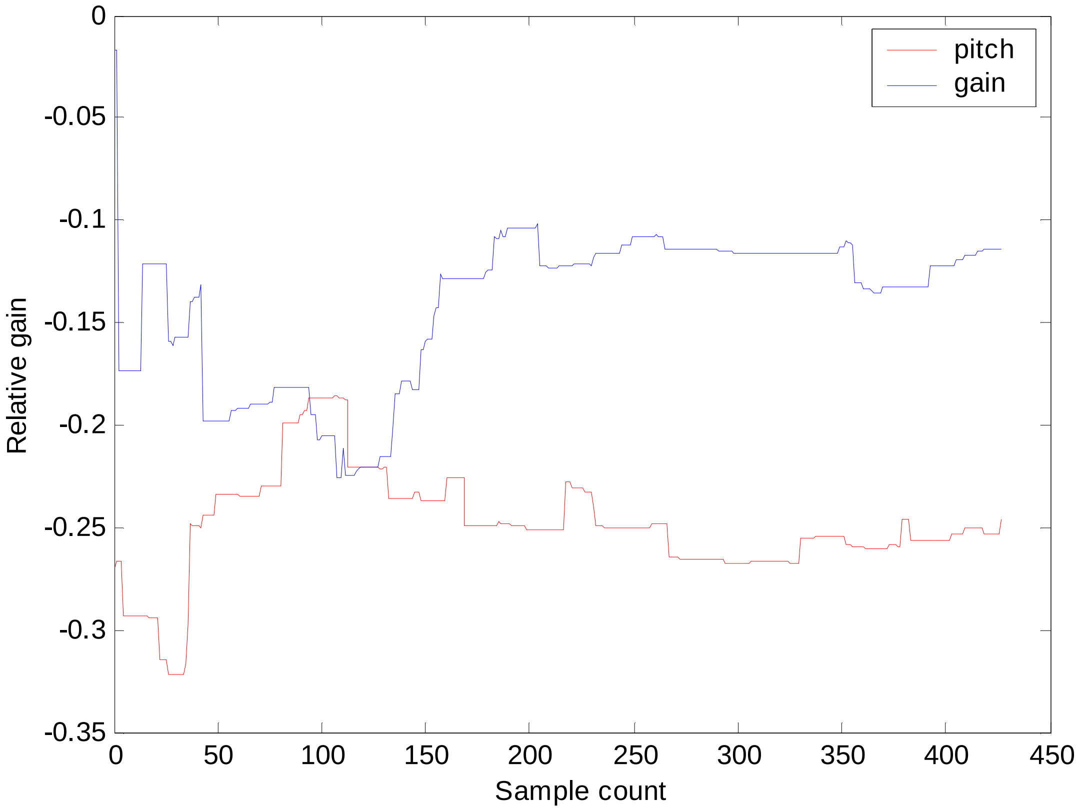

Figure 3-8 Relative “gain” in Euclidian error for the pitch and gain fields

However, the results obtained proved quite surprising. While we were expecting the Euclidian error between original and transmitted fields to drop when enabling our coding scheme, we could only witness that it rose in the simulation. Figure 3-8 shows, for the pitch and gain fields, the “gain” in error as computed by the simulation plotted against time. In both cases we actually observe a drop, by up to 25% for the pitch field, of the error.

Further study, over simpler channels, all showed similar results, i.e. that introduction of Gray coding in place of binary coding for the scalar data of our system did not reduce the measured Euclidian error, and even in most cases worsened it, thus shattering the preconceived idea that led us to try and decide to the experiment described. Furthermore, a more careful theoretical analysis indeed showed that what we had done was to map Gray coding (which indeed has nice properties when considered in a vectorial space of as many dimensions as the code has bits) to a one-dimension space, the real line, and that by doing so we could not take advantage of its properties of Hamming-proximity. This did not guarantee that we would necessarily get worse results, but it did guarantee that we would not get better results.

It is interesting to mention that the basically wrong idea on which this experiment was based was supported by the professor in charge in Mexico. It was thus his and my surprise to find out, upon closer examination, that it was indeed invalid and that there was indeed no reason to expect any improvement of the Euclidian error in the coded scalar values.

Conclusions

This dissertation presented a first attempt at studying joint source and channel coding over fading channels. Besides refreshing the reader's mind about well-documented theories and systems, it shows that it was possible to implement a communication system including a rather unique vocoder, a multipath, fading channel and part of a real-world mobile communication system.

While the aim of the project was apparently somewhat ambitious in comparison with its means, thus restricting the reach of its results, the author managed to put to shreds a misconception he was carrying, train his C and Matlab skills, and work on a rather interesting rate adaptation problem.

Nonetheless, the project could be pursued, maybe in the framework of a PhD thesis, in order to experiment about joint coding over a fading channel. The first step to consider, would be to integrate the finalized version of the TCENLP into the application, thus allowing to get more significant results, directly at the speech data level – such as, subjective analysis of the decoded speech, etc.

Furthermore, how to take advantage of known joint coding techniques, such as IA assignment, within the TCENLP vocoder, in order to replace the arbitrary index assignment currently taking place, could also be further studied, thus opening one axis of research. Another axis worth exploring would be to simplify the source of data, dropping the vocoder and replacing it by an abstract binary source, to concentrate only on how to adapt the generalisation of the Lloyd-Max algorithm presented in [4] from a binary symmetric channel to a fading channel.

Appendices

Appendix A. Original project description

A.1. Context

This is a joint research program led by the Cinvestav in Mexico and the ENST de Bretagne in France.

It is effectively led as a scientific cooperation exchange between France and Mexico, managed by ECOS-Nord in France and ANUIES/CONACYT in Mexico: while the responsible teachers in the respective universities manage the project, a French student goes to Mexico for four months from April 20th to August 20th, and a corresponding Mexican student will go to the French university from September on.

A.2. Outline

A digital communication system is typically made of two fundamental elements, with opposing functions: source encoder does compress information by suppressing redundancy in the signal, allowing, while keeping a low deterioration of the signal quality, to reduce bit rate, and thus to increase the capacity of the transmission medium. However, this natural redundancy is naturally robust w/r to noise. To compensate for the reduction of this robustness, the channel encoder, whose role is to fight against transmission errors, artificially introduces redundancy, to allow for error correction.

Until now, the two processes have been optimised separately. We are now looking to optimise globally the source and channel encoders, in order to better quality for a given bit rate.

The current speech joint source-channel encoding techniques assume errors caused by a symmetrical binary channel, as well as isolated, and as such easy to correct. We know how to compute the optimised structure of the vectorial quantifier for a CELP-type speech encoder, which is designed for the symmetrical binary channel. This is only a first step in joint optimisation for in reality, channels are less ideal: they could show gaussian perturbations, as observed with satellite communications, or perturbation following a Rayleigh law, as observed with multipath communications, such as mobile communications. In those cases, errors are grouped, and as such more difficult to correct.

The objective of the project is thus to progress in the elaboration of algorithms by choosing more realistic cases. For image encoders, solutions were designed that associate a DCT and vectorial quantification source encoding with a QAM modulation. Though the transmitted signals have different qualities, the student might want to get inspiration from these ideas for speech encoding. The student will as well consider extending the optimisation of vectorial quantisation with gaussian or Rayleigh noise.

More specifically, the placement shall include the following steps:

- to study past experiments (within the same project) and get general knowledge about the subject,
- to choose a specific application (e.g. DPCM with a Rayleigh channel and convolutive encoding) for a realisation,
- to realize a simulation (conditions of realisation open) of the chosen application and to derive optimised parameters thereof.

References

1. L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall signal processing series, Upper Saddle River NJ, 1978.

2. R. Goldberg and L. Riek, *A Practical Handbook of Speech Coders*, CRC Press LLC, Boca Raton, 2000.

3. A.M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems*, J. Wiley & Sons Ltd, Chichester, 1994.

4. S.B. Zahir Azami, *Joint Source/Channel Coding Hierarchical Protection*, PhD. Dissertation, École Nationale Supérieure des Télécommunications, Paris, 1999.

5. B.W. Kernighan and D.M. Ritchie, *The C Programming Language*, Prentice-Hall software series, Englewood Cliffs NJ, 1988.

6. Unspecified author, *CDMA Reference Blockset*, The Matlab R12 help, also available at http://www.mathworks.com/access/helpdesk/help/toolbox/cdma/cdma.shtml, as consulted on 19 September 2002.

7. S. McCloud, *Understanding Comics, the invisible art*, HarperCollins Publishers, Inc., 10 East 53rd Street, New York, NY 10022, 1994.

8. F. Villavicencio, *Compresión de voz con cuantificación vectorial y predicción no lineal basada en redes neuronales*, MSc. Dissertation, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Unidad de Guadalajara, Guadalajara, México, 2002.

9. L. Wu, M. Niranjan and F. Fallside, *Fully vector quantised, neural network based, code excited, non linear predictive speech coding*, IEEE Transactions on speech and audio processing, vol. 2 no 4, pp 482-489, 1994

10. Unspecified author, *Simulink*, The Matlab R12 help, also available online at http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/simulink.shtml, as consulted on 19 September 2002.