Automatic System of Reading Numbers

João P. Teixeira, Joaquim Silva, José Dias, Pedro Conceição and Pedro Freitas
ESTiG – Polytechnic Institute of Bragança, Bragança, Portugal
Email: joaopt@ipb.pt

Abstract— **This paper presents a brief introduction about text-to-speech (TTS) systems, its main structure and alternatives. In a more specific way, it was developed three algorithms of a system that automatic reads numbers. Each algorithm has its own functions and its own way to approach the problem. The algorithms have been programmed using Matlab software. The audio signals have been recorded and edited using Praat software. Finally a perceptual evaluation was made on each algorithm and was assigned a rating to each one. Generally, the MOS gives a very good level of classification for the three algorithms.**

Index Terms— **Automatic reading, numbers, text to speech system, TTS, numbers to speech.**

I. Introduction

Since the beginning of times that the man feels the need to express themselves and communicate. The language allows man to structure their thought, their feelings, record what he knows and communicate that to everybody. It is a skill that can be expressed in various ways, such as drawing, writing, reading, or even by emotions. But a specific case is the numbers. Numbers are used for everything and by everyone. A phone number, the age, a date, hours, a monetary quantity, a zip code, a car registration are some of the must common examples that we ordinary deal day by day. Thus it's so important to have some system or program that can read numbers automatically to be used in some applications. For example, an interface with a PC using speech interface is very helpful for blind person or a deaf mute person interaction, but also for several other applications.

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech [1-3].

Basically, synthesized speech can be created by concatenating segments of recorded/synthesized speech that are stored in a database itself of their parameters. Systems differ in the size of the stored speech segment units. A system that stores phones or di-phones provides the largest output range of sounds, but may lose some quality. For specific usage domains, the storage of entire words or sentences allows for high-quality output [4-5]. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output [6-7].

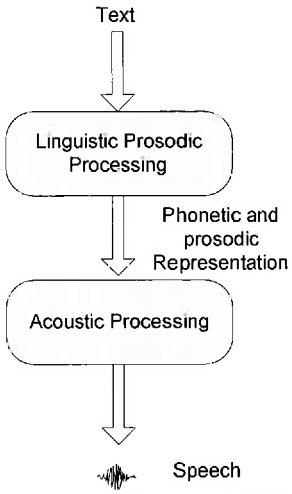

Figure 1- Generic Diagram of a TTS System.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood [5]. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer [4]. Many computer operating systems have included speech synthesizers since the early 1990s.

A. Text-To-Speech systems

Systems capable of creating voices through the text are commonly referred by the abbreviation TTS (Text-To-Speech). Currently there is a variety of this type of systems, from business applications to other developed for educational purposes. A complete TTS system is divided into two parts, the linguistic-prosodic processing and the acoustic processing [4-5], as depicted in Fig. 1.

The text is the input of the TTS system, then in the Linguistic-Prosodic module several task take place, such the text pre-processing used for converting numbers, abbreviations, acronyms and other characters into text to be processed by the linguistic sub-module. The linguistic sub-module converts text morphemes into phonemes at the segmental level, identifies several information chunks such phrases, paragraphs, words, syllables and phonemes. Some advanced linguistic modules also perform a syntactic and/or grammatical analysis if the correspondent parser is available [4-5]. The prosodic sub-module processes at the supra-segmental level and inserts prosodic information such the phrasing organization, phrase focus and word focus converting this information into prosodic acoustic parameters such Fundamental

Frequency (F0), timing, and sometimes also intensity. The identification and modulation of prosody is a hard task. For timing modulation several models have been used, such the Z-score model [8], Barbosa & Bailly model [9], or models based on Artificial Neural Networks [10]. F0 is the perceptually most important acoustic parameters to convey prosody. Several models/techniques have been studied to model this parameters such the Fujisaki model [10-11], the ToBI (Tone and Break Indices) model [12], the tilt model [13] or the INTSINT (INternational Transcription System for INTonation) [14].

The acoustic processing module produces the acoustic speech signal corresponding to the sequence of phonemes and with the prosody modelled in previous processing blocks. Several methods were been used since the Klatt formant model [15], the LPC family models been the Residual Excitation Linear Prediction RELP the most acceptable in quality but no longer in use, the sinusoidal model [16] also obsolete, the PSOLA models been the Time Domain PSOSA the most common [17], the concatenation model still with good quality actually, the articulatory models [18] very heavy and not in use, the selection of units [19] model with very good quality and the HMM [20] model that were very promising.

The most important qualities of a speech synthesis system are naturalness and intelligibility. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics.

B. Objetives of the work

Usually the number appears inside a text as a sequence of numerals that should be converted to a text for the TTS read this text. Anyhow the way this conversion should be made depended on what the number is. The conversion of a telephone number must be different than the conversion of an amount of money, for instance. In the case of the telephone number it is more appropriate to convert digit by digit but the amount must be converted as a whole number. Also the prosody must be appropriated for each case. The telephone number typically can be read as a sequence of groups of 3 digits.

Therefore the type of number must be identified to be correctly converted and read. Actually there are the following important tasks concerning the numbers conversion that must be taken into consideration:

1. Identification of the type of number;
2. Conversion of the numeric characters to text or some symbolic representation of the corresponding text;
3. Synthesis of the corresponding text.

The objective of this work consists in make the conversion and synthesis of universal numbers from 0 to 1 billion (10^9).

In this paper we are presenting three different algorithms, each one with its own structure, functions and characteristics, but with the same purpose, the automatic identification of numbers and the respective reproduction, from 0 to 10^9. The synthesis used for this specific purpose consists in the concatenation of recorded segments.

In this particular case the system was developed for the European Portuguese Language. Basically, the present automatic system of reading numbers, starts by read the numbers introduced by the user and proceeds to its identification or in which group the number belong, units, dozens, hundreds, thousands, dozens of thousands, hundreds of thousands, millions, dozens of millions, hundreds of millions. After that, it is needed to make the concatenation of the speech sound corresponding to the number. If it is necessary, the particle "e = and" are added.

II. DATABASE OF SOUNDS

The synthesis of the identified number consists in a simple concatenation of the corresponding digits and particles.

The database of sounds consists in the speech of each digit in different positions and the particles.

To build the database of sounds the cutting of each segment is crucial to obtain a better quality both at segmental and supra-segmental levels because no prosody processing will be used.

The Praat software [21] was used to record and editing of the speech sounds.

The prosody of a digit is significantly different in length and in intonation depending on the position of the digit in the number, therefore different intonations, depending of the positions of the digit, are needed. Consequently, two different sounds for the same digit were recorded. One will be used at middle positions of entire number and the other only at the final position.

The sound used at middle position have a lower duration than the other and it must be cut so that the intensity and the pitch don't decrease because it will be proceeded by another sound that could be another number or units like million or a particle "e"= "and". Fig. 2 shows the cut of the word "cem" correspondent to the "100". In the figure the top plot presents the acoustic wave form and the lower picture shows the spectrogram of the waveform in a black and white color pallet, with a red dots the estimated frequency formants and with yellow line the fundamental frequency (F0).

The same sound segment (100) in a final position is presented in Fig. 3. This segment is used also for the number alone that actually is also a segment in a final position. The segment in a final position is naturally a longer sound (for the example of Fig. 3 it has 421 ms long and the one of Fig. 2 it has only 163 ms long). Also the F0 denotes a decrease at the end of the segment that gives the information to the listener that this is the last digit of the number.

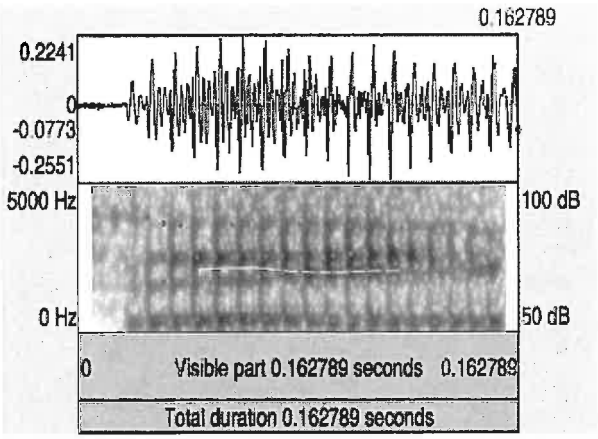

Figure 2. Cut of the number 100 "cem" recorded sound used at a middle position.

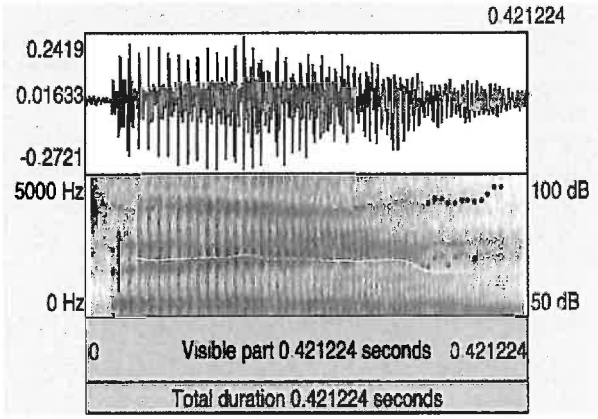

Figure 3. Cut of the number 100 "cem" recorded sound used at a final position.

Concatenative synthesis is based on the concatenation of segments of recorded speech. Generally, concatenative synthesis of long segments produces the most natural-sounding synthesized speech because no changes in the sounds are needed. This is the principles beyond the unit selection synthesis method [19]. However, differences in the phase or in the magnitude of the acoustic wave of the concatenated sounds due to the segmentation of the sounds sometimes result in audible glitches on the output. There are three main sub-types of concatenative synthesis.

In this work the Time-Domain specific synthesis is used. It concatenates pre-recorded words and numbers to create complete utterances. It is used in applications where the variety of texts to be produced is limited to a particular domain, like transit schedule announcements or weather reports. The technology is very simple to implement, and has been in commercial use for a long time, in devices like talking clocks and calculators. The level of naturalness of these systems can be very high because the variety of sentence types is limited, and they closely match the prosody and intonation of the original recordings.

Because these systems are limited by the words and phrases in their databases, they are not general-purpose and can only synthesize the combinations of words and numbers with which they have been pre-programmed. The blending of words within naturally spoken language however can still cause problems unless the many variations are taken into account. This alternation cannot be reproduced by a simple word-concatenation system, which would require additional complexity to be context-sensitive.

III. Presentation of the Algorithms

In this section the three developed algorithms are presented.

A. Algorithm 1

This algorithm was created and developed at "Matlab" software to reproduce numbers up to nine digits (inclusive) and mobile phone numbers of Portugal (with nine digits). It is based on a sequence of "if" conditions in various functions for different size numbers.

For start, the database of speech digit sounds was recorded containing the essential sounds like base numbers, units and particles. This speech files only will be charged to the memory of the program when they are needed.

The main function starts by creating an empty vector that will receive the segments to be concatenated and finally send to loudspeaker as the output. Then converts the number introduced by the user to a string and determines the size and the position of each digit to be later associated to the respective file of the database. If the number has nine digits and starts with 91, 92, 93 or 96 it will ask the user if the number is a mobile phone number.

For the other numbers, a function dependent of the size and position of the digit is called. This function will insert the speech segment in the vector. Each position has a Matlab function that will read the respective wave file for each base number. Since each number has two records for last position and middle position, the position will define the file to be read. This allows giving different intonations depending of the position of each number.

The particle "e" = "and" and the words "mil" = "thousands" or "milhões" = "millions" are inserted according to the respective position. In Portuguese the word "hundreds" are not added because it is pronounced in a single word. For instance, 500 is only one word "quinhentos" and not "five"+"hundreds".

For phone numbers the process is similar, but each digit will be reproduced as a single units. Analyzing each position it is added to the vector the correspondent digit. The way is reproduced is simple, starting from left to right, the first two digits are read in one group, after the following three digits are together in another group, then the next two digits, and finally the last two digits. The last digit of each group is reproduced using the final position version of the digit. Like the other type of numbers the sounds vector is reproduced at once.

Fig. 4 presents the general flowchart of this algorithm.

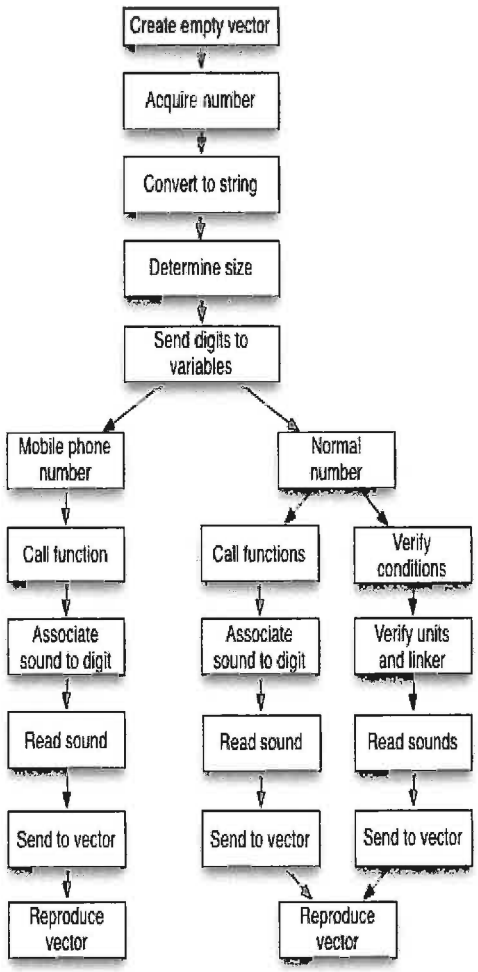

Figure 4. Flowchart of the Algorithm 1.

B. *Algorithm 2*

This algorithm was implemented at the Matlab software to reproduce numbers up to 999999999.

The algorithm was constituted by two functions. The first function was required to place conditions needed to reproduce numbers and the connections needed to reproduce correctly all digits. The second function is basically to select and load the acoustic file required. In this function it is verified if the digit is terminal or not, if the number is terminal it loads the terminal digit recorded initially, if not it loads the middle position digit.

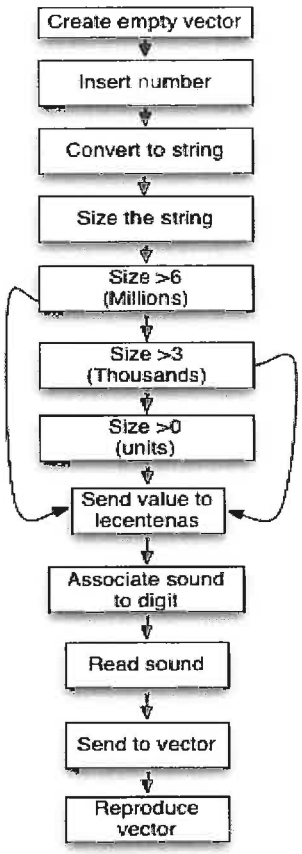

Figure 5. Flowchart of the Algorithm 2.

After insert a number, this number will be converted to string. With the string, it is easier to determine the length of this number. From here this algorithm will associate the digits to the variables ($D_9D_8D_7$ $D_6D_5D_4$ $D_3D_2D_1$) according to their position and to the length of the number. Each group ($D_9D_8D_7$), ($D_6D_5D_4$) and ($D_3D_2D_1$) is defined as belonging to the millions, thousands or units respectively. The group of digits is sent to the function that will 'read' the 'hundreds' (lecentenas). In the condition of the millions, there is other condition to indicate when the algorithm will insert the word "milhão" (million) or "milhões" millions. In the second function, he can also, verify if the number to load is terminal or not, in other words, it 'sees' when and what number are in the last position.

After that, the values will be passed to the vector, this means that vector receives the values given by the function 'lecentenas' and the elements/particles needed to connect the numbers ('e', 'milhões' and others). Only required sounds will be loaded to the computer memory.

Fig. 5 presents the general flowchart of this algorithm.

C. *Algorithm 3*

This algorithm was been developed and implemented in Matlab software to read numbers from one digit to nine digits (0 – 999 999 999). It is based on a set of conditions.

There is a main function that calls others functions as they are needed.

At the beginning, the database was recorded according to previous section II. After that, the sounds will be associated to the respective variable, and charged only when they are called.

The main function gets the number from the user and creates an empty vector that will be filled with the sequence of numbers that will be reproduced.

To determine the size of the number and the position of each digit, the number is converted to a string and all positions are associated into the variables.

Depending on the size of the number, the respective function is called. The function will compare the digit in each position with the respective variable, and read the corresponding sound, verify all the required conditions to add the particle "e" when it is needed and the units "thousand", "million" or "millions" are also added. The vector is filled with the sequence of sounds to be reproduced at once avoiding pauses between digits.

To improve the prosody, all the digits has two types of records, differentiated by their position, if they are in the middle of the number or the final position, according to description made in section II.

Fig. 6 presents the general flowchart of this algorithm.

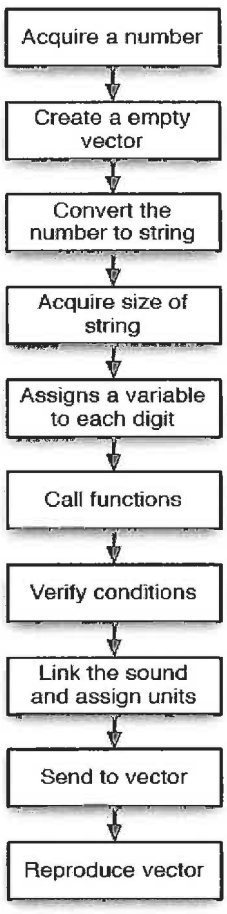

Figure 6. Flowchart of the Algorithm 3.

IV. RESULTS

A perceptual test was performed in order to measure the accepted quality of each algorithm. A mean opinion score (MOS) test was used with a scale from 1 (very bad) to 5 (excellent). A total number of seventeen (17) subjects were asked to evaluate each one of the fifteen (15) reproduction of a random numbers between 0 and 999999999.

The score were given by the quality of sound, audio perception and correct reproduction.

After statistical analysis of the MOS the results are showed at Table I.

All tested numbers had been reproduced correctly without processing errors.

It is possible to verify from table I that the results are similar to the three algorithms and with general level of very good quality. Anyhow the prosody was not perfect.

Fig. 7 shows the average ratings of algorithm 1. It can be noticed a small variations between 3.8 and 4,6 points.

Fig. 8 shows the same plot for algorithm 2. At this case the average ratings is between 3,75 and 4,30 points.

Finally, Fig. 9 displays the MOS for the 15 subject for algorithm 3. Now the MOS varies between 3,80 and 4,50 points.

TABLE I
Results of all ratings

Average of points to algorithm 1	4,20
Median of points to algorithm 1	4,00
Variance of points to algorithm 1	0,64
Standard deviation of points to algorithm 1	0,80
Average of points to algorithm 2	4,07
Median of points to algorithm 2	4,00
Variance of points to algorithm 2	0,74
Standard deviation of points to algorithm 2	0,86
Average of points to algorithm 3	4,16
Median of points to algorithm 3	4,00
Variance of points to algorithm 3	0,64
Standard deviation of points to algorithm 3	0,80
Average of points to all algorithms	4,15
Median of points to all algorithms	4,00
Variance of points to all algorithms	0,67
Standard deviation of points to all algorithms	0,82

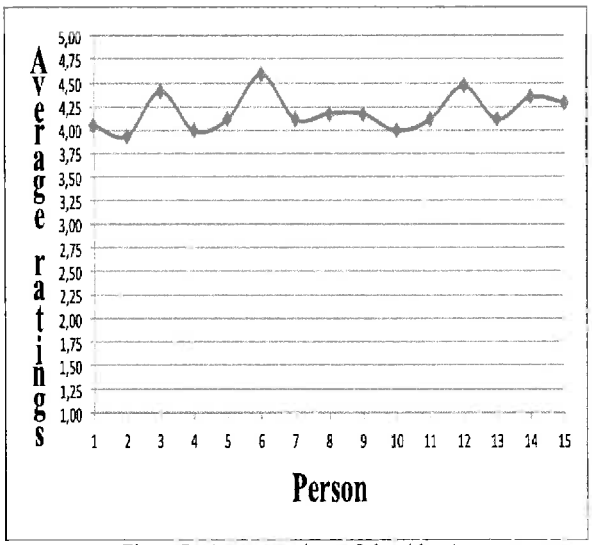

Figure 7. Average ratings of algorithm 1.

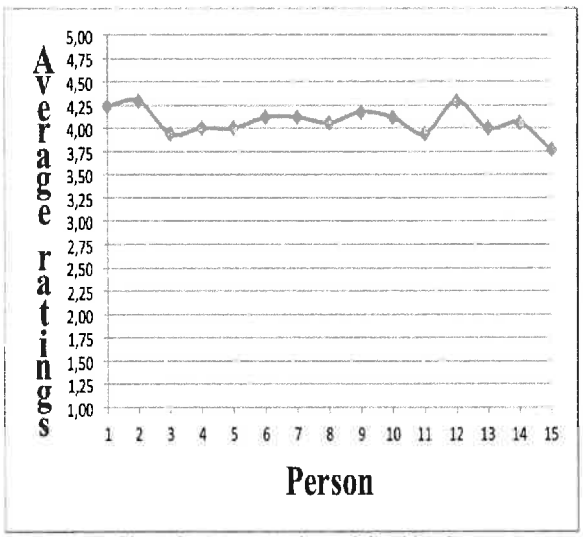

Figure 8. Average ratings of algorithm 2.

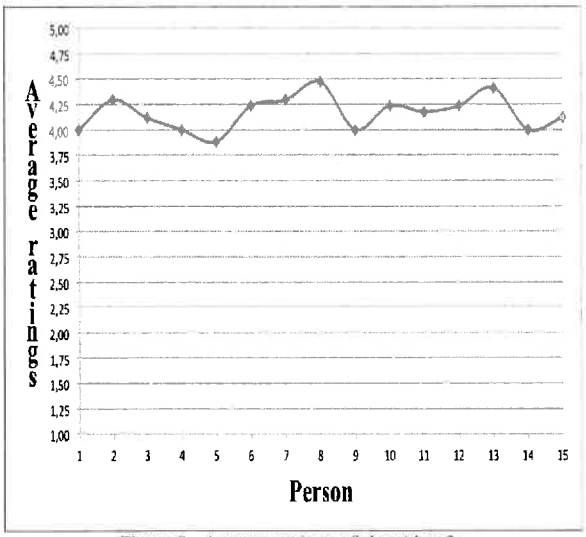

Figure 9. Average ratings of algorithm 3.

CONCLUSIONS

The paper presents three algorithms developed and implemented in Matlab software to reproduce numbers until 10^9 and mobile phone numbers of Portugal. Despite the small differences between them the results of a perceptual test gives a MOS at very good level. The algorithm number 1 received a relatively higher MOS (4.2).

For the synthesis systems it's important begin with good quality of the recorded sounds and with precise cuts, otherwise it will no longer provide good results. The speaker must have a clear voice and nice diction.

The algorithms need to take into consideration various specific numbers to insert all necessary conditions for a perfect reproduction. Not less important is that the reproduction occurs only once to avoid pauses between the digits, units and particles.

Improvement of the sensitized speech system can be achieved introducing some prosodic modifications. This allows the listeners, for example, recognize what is the last digit of the number.

The system will continue with development in order to recognize different types of numbers like dates, hours, personal identification numbers, bank balances along others. Also the reproduction of math equations or math operations is a task that already received some attention by other researchers [22-23].

REFERENCES

[1] Klatt, D. H., *Review of text-to-speech conversion for English* - Journal of the Acoustical Society of America, 82 (3) - 1987. Pages 737-793.

[2] R.A. Sharman, *Text To Speech System*, United States Patent number 5774854, 1998.

[3] Teixeira, J. P., *Análise e Síntese de Fala – Modelização Paramétrica de Sinais Para Sistemas TTS*. Editorial Académica Española ISBN: 978-3-659-06206-3, 2013.

[4] Teixeira, J. P.,Barros, M. J. and Freitas, D., *"Sistemas de Conversão Texto-Fala."* Procedings of CLME, Maputo, 2003.

[5] Barros, M. J., *Estudo Comparativo e Técnicas de Geração de sinal para Síntese de Fala.* Master dissertation, Faculdade de Engenharia da Universidade do Porto, 2002.

[6] Saraswathi, S., *Design of Multilingual Speech Synthesis System.* Academic journal article from Intelligent Information Management, Vol. 2, No. 1, 2010.

[7] Sproat, Richard W., *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach.* Springer, 1997.

[8] Campbell, W. N., *Timing in Speech: A Multi-Level Process.* In "Prosody: Theory and Experiment". Edited by Merle Horne, Kluwer Academic Publishers, pages 281-334, 2000.

[9] Barbosa P., Bailly G., *Characterisation of rhythmic patterns for text-to-speech synthesis*, in Speech Communication, 15: 127-137, 1994.

[10] Teixeira, J. P., *Prosody Generation Model for TTS Systems - Segmental Durations and F0 Contours with Fujisaki Model*. LAP LAMBERT Academic Publishing ISBN-13: 978-3-659-16277-0, 2012.

[11] Fujisaki, H., *Dynamic characteristics of voice fundamental frequency in speech and singing. In MacNeilage.* In P. F., Editor. The Production of Speech, pages 39-55. Springer-Verlag, 1983.

[12] Pierrehumbert, J. B.. *The Phonology and Phonetics of English Intonation*. PhD thesis, Massachusetts Institute of Technology, 1980.

[13] Taylor, P., "Analysis and Synthesis of Intonation using the Tilt Model". *Journal of the Acoustical Society of America*. vol 1073, pp. 1697-1714, 2000.

[14] Hirst, D. and Di Cristo, A., *Intonation Systems – A Survey of Twenty Languages*. Cambridge University Press, 1998.

[15] Klatt, D. H., "Software for a cascade/parallel formant synthetizer". Journal of the Acoustical Society of America, 67:971-995, 1980.

[16] Marques, J. S. S., *Modelamento Sinusoidal da Fala – aplicação à codificação a ritmos médios e baixos*. PhD thesis – Instituto Superior Técnico, Lisbon, 1990.

[17] Charpentier, F e Moulines, E., "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones". Speech Communication, 9 (5/6):452-467. 1990.

[18] Silva, C. A., *Automatic Extraction of the Parameters of an Articulatory Model for Speech Synthesis*, PhD Thesis, DEI- University of Minho, Portugal, 2001.

[19] A. Conkie, "A robust unit selection system for speech synthesis." In: Proc. 137th meet. ASA/Forum Acusticum, Berlin, March 1999.

[20] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in Proc. EuroSpeech1999, vol. 5, 1999, pp. 2347–2350.

[21] Boersma, Paul and Weenink, David. "Praat: doing phonetics by computer". Phonetic Sciences, University of Amsterdam. http://www.fon.hum.uva.nl/praat/

[22] Ferreira, H. and Freitas, D., "AudioMath: Towards Automatic Readings of Mathematical Expressions", Human-Computer Interaction International (HCII) Las Vegas/USA, 2005.

[23] Braga, D.; Freitas, D. and Ferreira, H., "Processamento Linguístico Aplicado à Síntese da Fala". Procedings of CLME, Maputo, 2003