TTS DEVELOPMENT ENVIRONMENT RESULTS

ARPAD ZSOLT BODO[1], GAVRIL TODEREAN[2] and OVIDIU BUZA[2]

The development of a text-to-speech synthesis system is a complex process that has to be elaborated in several fields like: speech signal processing, prosody prediction-generation, synthesis techniques/engines, database construction, etc. During research and development phase, each of these domains requires a set of tools. The need for these tools is multiple: sometimes from the complete TTS process flow, intermediate results are required to be visualised, sometimes external (manual) interaction to the system helps the researcher's work. The current paper shows a collection of these self-developed tools, which finally have formed a development environment. However this framework is built around a general voice signal editor freeware software, all the TTS specific tools and the graphical user interfaces, are the result of an own development. The main focuses of this paper is to present the development environment of PSOLA and other synthesis techniques, the diphones database processing environment, the speech signal processing toolset, prosody modification related issues and intonation curve computation algorithm.

Key words: TTS engine; Synthesis technique; Waveform; Intonation curve; Pitchmarking; Prosody manipulation; Pitch descriptor; Prosodic descriptor; Temporal structure; Speech signal processing; Intonation.

1. INTRODUCTION

The realization of a Text-To-Speech (TTS) synthesis system is a complex and voluminous development procedure. Besides its laborious characteristic, the process of development mainly means research work. In order to support this work, a dedicated toolset – a so called development environment or framework – is needed. These environments practically encapsulate the synthesis system's components and allow the developers freedom to make experiments during their work, to see and check intermediate results, to intervene and make changes, to graphically visualise parameters, etc. The development environments and the synthesis systems under construction, have interweaving architectures, coexist and cooperate very closely.

Because development environments are very system and developer specific, and are kept confidential, each research centre needs to develop its own toolset. For a smaller team, this is a significant overhead; therefore the importance of code reuse is not negligible. The current paper presents such a code reuse solution, therefore an open source program (Audacity under the GNU General Public Licence – see [8]) had been adopted, as a basis. This software is an audio recording and editing tool, compilable for several platforms like Linux, Mac or Windows.

Inheriting the basic functionalities, after some of their modification, and after adding the TTS specific functionalities, a toolset had been achieved, which can serve as a synthesis system development environment. The covered areas are: speech waveform editing, prosody manipulation, synthesis technique implementation and testing, and the most important: database creation.

In order to understand the research and development areas of such an activity, one has to be familiar with the components of a speech generation system. The main constituents are shown in the Figure 1. The dashed arrows indicate that there is a bilateral connection to the development environment. In the ideal case, there would be even more connection points.

However this framework is built around voice-signal editor freeware software, the TTS specific tools, and most of the graphical user interfaces are the result of own development. The main fields detailed below are: the PSOLA synthesis technique development environment, the diphone database processing environment, the speech signal processing toolset and the text and prosody related area.

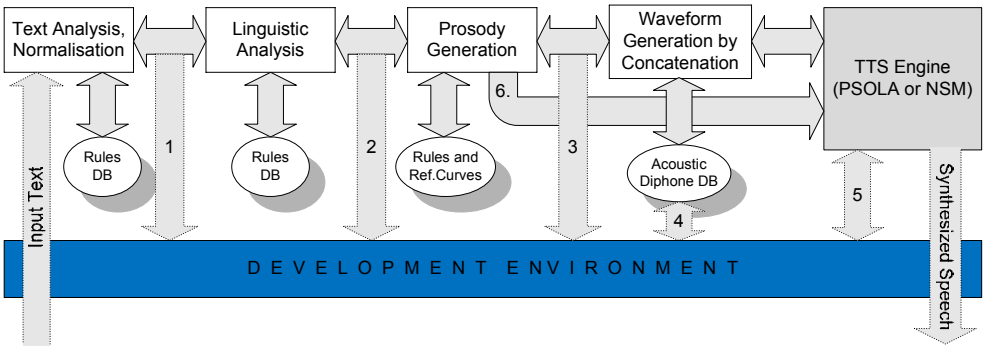

Fig. 1. TTS system integrated to its development framework.

2. DEVELOPED FRAMEWORK FUNCTIONALITIES

A TTS synthesis system development framework has to provide numerous functionalities. In the following paragraph some of the added important functionalities are presented.

File operations

Existing file open/create/save/close functionalities were extended to extra file formats (*e.g.* to store pitchmarks and other parameters: **.ptm*, **.tag, *.xml, *.intoncurve*). In case of operating with waveforms, the open/create/save/close operations are in the background in parallel automatically executed on the parameter files also (*e.g.* when a wav file is modified, its corresponding pitchmarking file is automatically updated).

Physically this menu contains the newly added special menu entries: *"Input sentence", "Select diphone database", "Select rule database", "Test Import Concatenated"* and one tab of the *"Preferences"*.

Waveform visualizer and editor

The reused basic functionalities are: select, copy/cut/paste/delete/trim, zoom in/out, zoom to selection, fit in, scroll, undo/redo, etc. Later, these classes had been extended according to our own needs. The visualizer operates on synchronized tracks, making possible, that in parallel to a *waveform* its *intonation curve*, *pitch structure* and its *temporal structure* can be seen as well.

Another added feature is to visualize the spectra.

Manual concatenation

A helpful feature is, to manually concatenate some acoustic files, using multiple file selections and opens (*"File/Test Import Concatenated"*). Of course the order of these files could also be controlled.

Pitchmarking tool

The diphone database elements can be processed manually using the basic functionalities, but there is a need for parameterisation as well – see Figure 2. The result of this is a parameter file next to each acoustic unit, which contains pitchmarks, phone margin markers, their type, etc.

Intonation curve calculation

Having any speech sound waveform, without e.g. pitchmark parameters, one can determine the melody of this by just using a dedicated own tool – see Figures 5, 6, 7. This tool implements a set of algorithms, and offers a GUI for configuring, combining and parameter settings.

Intonation curve manipulation

The concatenated monotone speech signal can be supplied with a different prosody. This can be done using our tool, which allows manual intonation curve editing – see Figure 4. Afterwards this curve is imposed to this speech signal, without modifying the temporal structure.

Temporal structure modification

Another tool shows the temporal structure of a concatenated speech, and allows the manual modification of this by a mouse. Having the desired temporal structure, this is imposed to the analysed speech without modifying its intonation curve.

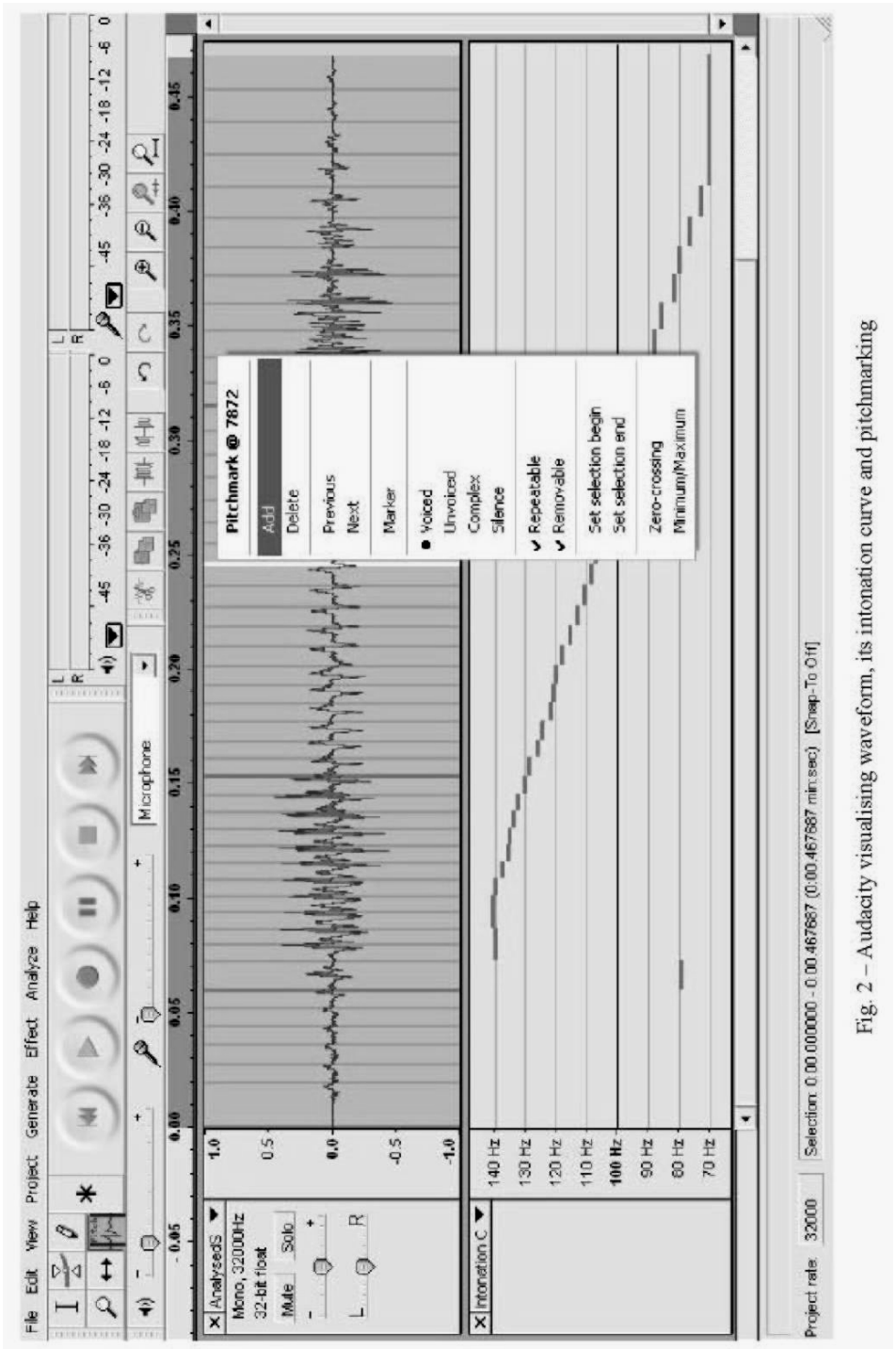

Fig. 2 – Audacity visualising waveform, its intonation curve and pitchmarking

Digital signal processing

In order to calculate the intonation curve, several digital signal processing tools had been implemented. These are: windowing techniques, band-pass filters, FFT, Autocorrelation, Cepstrum, AMDF, etc.

Menus, Context menus, Tooltips, Hotkeys

Originally, the software menus grouped the functionalities into: *File operations, Edit, View, Project, Generate, Effect, and Help*. The first three are standard menu trees with specialised menu entries. The software operates with projects, meaning, that a session itself can be saved in project files, not only the resulting sound or parameter files. The Generate and Effect menu trees offer functionalities, which are useful in general sound processing. Our functionality extensions can be accessed by menus added to the enumerated trees. Other possibilities are context menus (right click), toolbars and hotkeys, etc. According to our needs, we have extended the tooltip and fast access key sets as well.

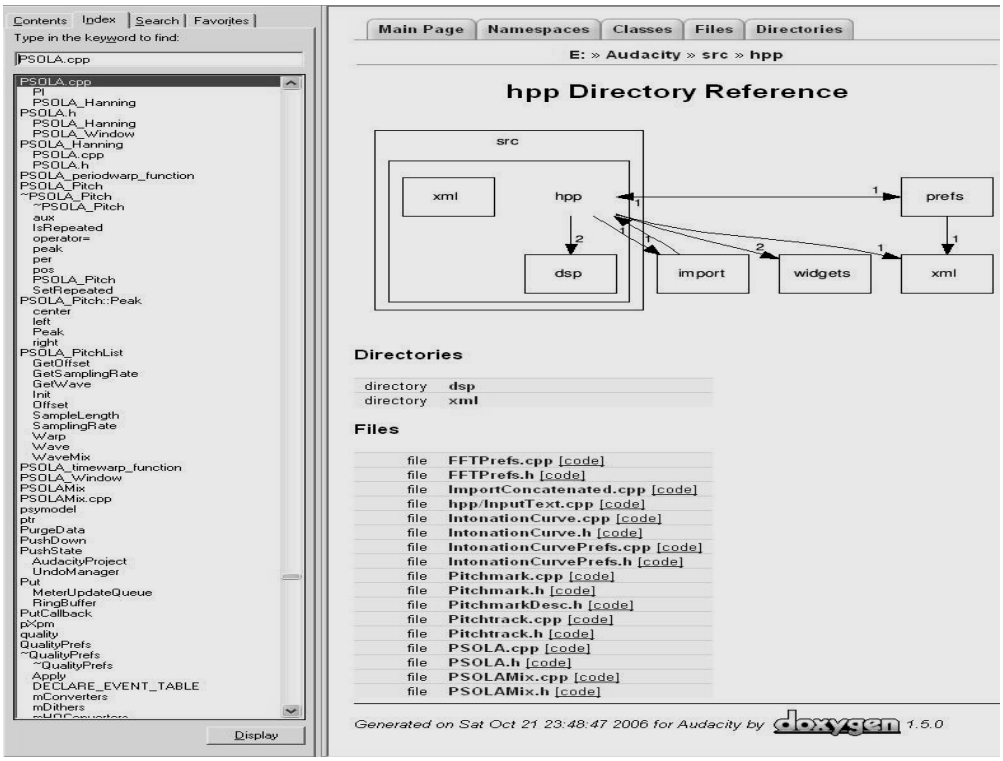

Fig. 3. Doxygen generated help.

Toolbars, Tool switch

There exist Control, Editor, Mixer and Meter toolbars. The first two are really relevant for us.

Besides editing (so called Selection Tool), other new tools had been implemented, like: *Pitchmark tool*, *Extended Envelope tool*, etc. These tools are derived from the *Selection tool* and have inherited many of its functionalities. Toolbar buttons make possible to switch between our tools.

Play, Recording

Playing and recording are other inherited functionalities, which had been adopted.

Signal generator, Effects, Help

An existing functionality is the signal generator, which can provide: tone (sin, square, and saw tooth), white noise and silence. There are many built-in effects that are not used by us. There exists a help menu too, but the own implementation is commented Doxygen-like and a compiled html (**.chm*) file is generated (see Figure 3).

3. SYNTHESIS TECHNIQUE DEVELOPMENT WITHIN THE FRAMEWORK

After making some experiments with the synthesis technique NSM (Nonlinear Springing Method) from [1], the next step was to implement the PSOLA (Pitch Synchronous Overlap and Add) synthesis technique – [6]. To detail the theoretical background of these techniques is out of scope of the current paper.

PSOLA is a technique capable of independently modifying prosodic parameters of a speech signal (intonation curve, temporal structure and intensity) according to the prediction of a prosody generator. In order to test this, the results had to be visualised, making it possible to see each characteristic: amplitude envelope, intonation curve and temporal structure. Moreover, for a very efficient testing, the input parameters should be editable as well. During the implementation of this Time Domain PSOLA method, the development environment had to be equipped with certain features, which could be used by the database development as well.

Figure 4 shows a concatenated monotone speech signal before and after the modification of its prosodic characteristics. The first waveform (Sgl_orig) has a quite monotone intonation, just as the diphones were stored in the acoustic database.

In the current stage, the prosody generator was also under development, so the test values prosody input for the synthesis technique had to be acquired from some other source. The solution was to develop such prosodic parameter visualized, which allow editing as well. This can be done with a mouse by dragging the lines, which

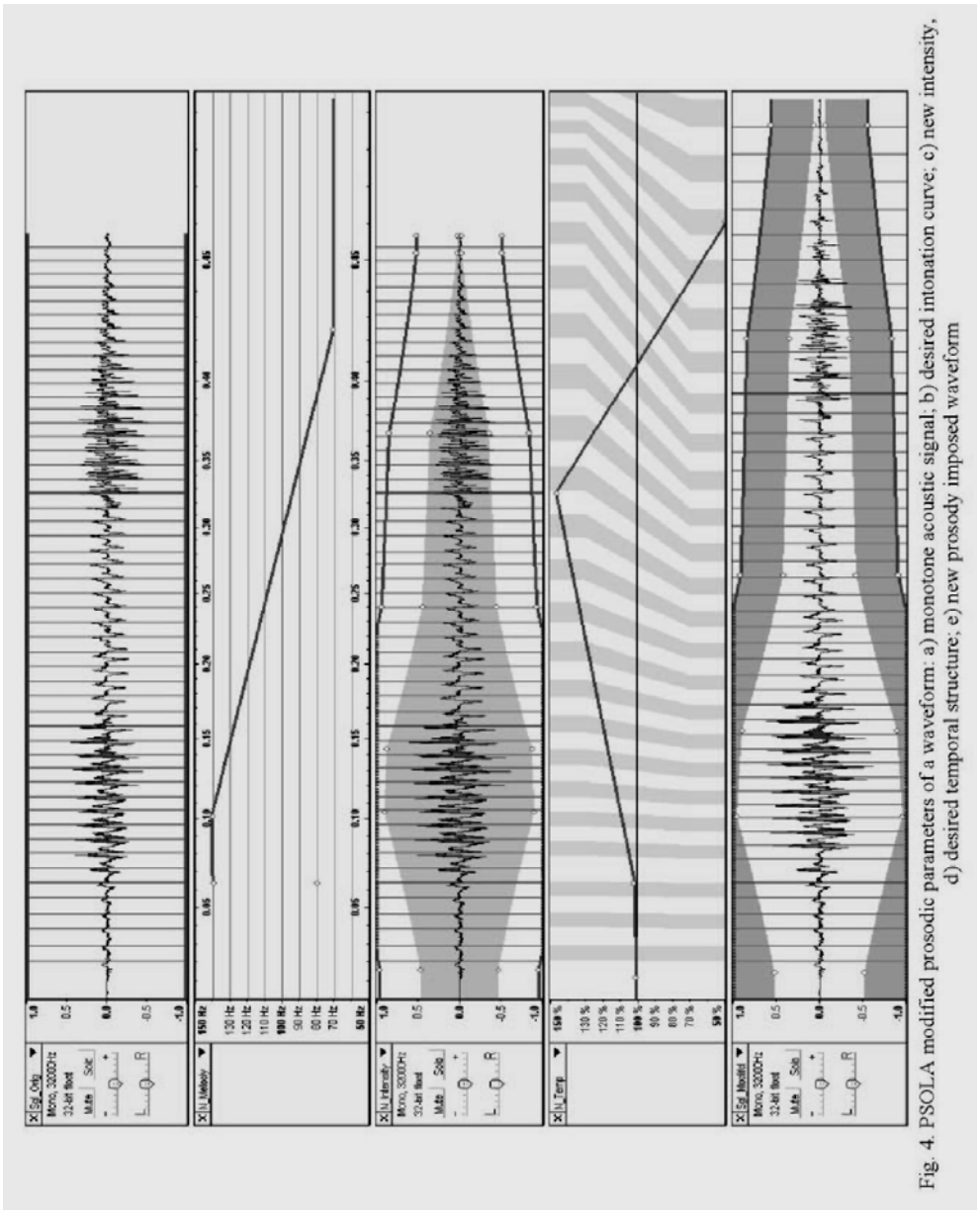

Fig. 4. PSOLA modified prosodic parameters of a waveform: a) monotone acoustic signal; b) desired intonation curve; c) new intensity, d) desired temporal structure; e) new prosody imposed waveform

represent the value curve. The desired intonation curve (N_Melody) is shown by b). The desired intensity envelope (N_Intensity) is shown by c), while the new temporal structure (N_Temp) is constructed in d).

After manually creating the input prosody, the so called PSOLA Mix command is executed, in order to impose these characteristics onto the original monotone speech signal. The result (Sgl_Modifd) can be seen on the Figure 4 – e).

4. SOUND DATABASE DEVELOPMENT WITH THE FRAMEWORK

In order to serve the acoustic database realisation, the development environment had to be prepared for this as well. Our TTS system is a diphone based concatenative one, which uses PSOLA synthesis technique. Consequently, highly pre-processed database elements were needed. The first database realisation phases did not need this environment. Once that target text was recorded, the raw waveform was completely processed by this self developed tool. Its main advantage is, that on the market there exist no tool, which contains in one all of the needed features. The waveform editor, the filtering, the amplifier, the file operations help to obtain the final diphone wav file.

After having these diphones, each file needs to be parameterised. This was being done using our Pitchmark tool. This is shown by Figure 2, where one can see the pitchmarking context menu and its elements. The tool in this example adds/deletes a pitchmarker (vertical line) at the sample 7872. The type of this is not a phone margin; therefore, the "Marker" is not set. But this pitch period is voiced, repeatable and removable. The tool offers some helper functions as well, like: jump to previous or next, place at a zero crossing, etc.

When this is manually performed on the complete diphone, the export/save functionality automatically creates a file with the same name as the wave, but having an extension of *.ptm. For this, a special file format had been worked out. Whenever next time this wave is opened, the parameters from the *.ptm file are automatically loaded.

5. INTONATION CURVE CALCULATION BY THE ENVIRONMENT

In the process of creation of the reference intonation patterns (used during prosody generation) a very important procedure is the determination of the intonation curve of any speech signal. In order to perform this, a tool had been developed, which allows runtime configuration of parameters and the algorithm tailoring too. Figure 5 shows the two ways of intonation curve computation: Figure 5a is trivial, based on the pitchmarking information (possible only when these parameters are available); Figure 5b presents the very high level schema of a self-developed algorithm.

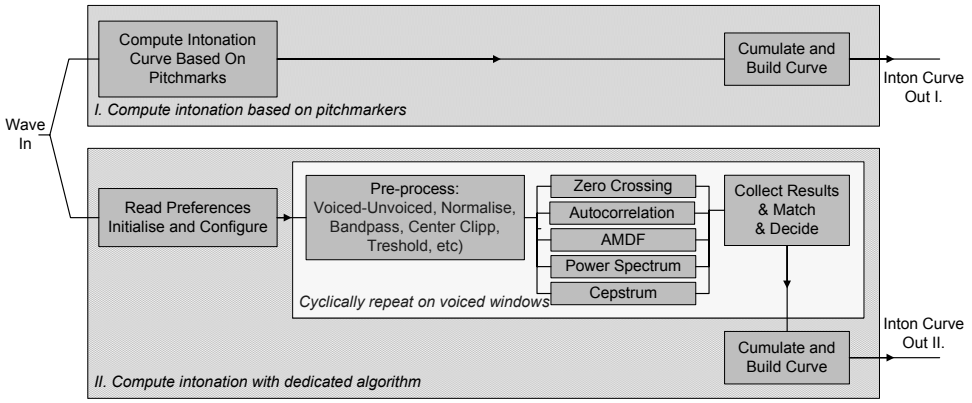

Fig. 5. Intonation curve determination: a) I. Using pitchmarkers; b) II. With an own algorithm.

After reading the preferences (see Figure 7), namely the initial parameters and the composition of the algorithm, follows a phase, that is executed on each consecutive window. The first step is the pre-processing. This is also configurable from its elements and parameters point of view (e.g. whether is there a band-pass filtering and which is the band, etc). The next step is to apply a variety of different frequency determination methods (Zero crossing, Autocorrelation, AMDF, Power Spectrum, Cepstrum). Finally the results are compared and matched, the very divergent results are omitted. At the end the cycle results are cumulated and the intonation curve is built.

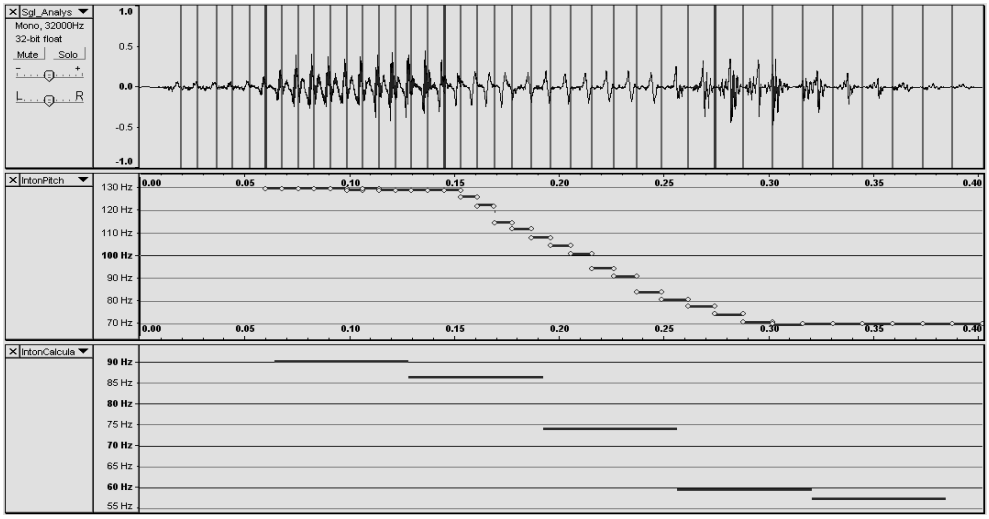

Fig. 6. Practical result of intonation curve determinations: a) analysed speech signal; b) intonation curve based on pitchmarkers; c) own algorithm computed intonation curve.

Figure 6 presents the practical results of this algorithm. Figure 6a shows the analysed speech signal that due to presentation reasons has an extreme melody. Figure 6b displays the trivial but very precise intonation curve, which is obtained according to Figure 5a from the pitchmarking information. Of course when there is no pitch information, only the second, algorithm can be used. The Figure 6c shows the result of the Figure 5b presented algorithm. In order to see the particular details of this computation, one can check the Figure 7.

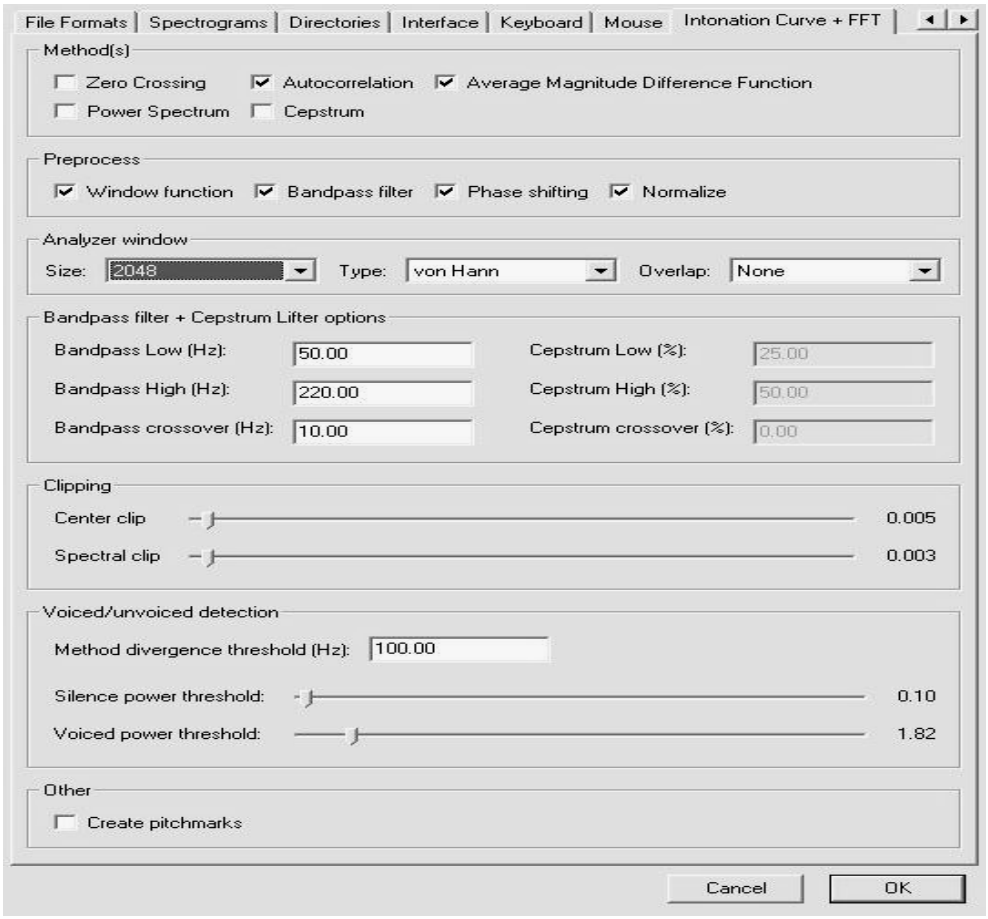

Fig. 7. Configuration tab for the intonation curve computation algorithm and parameters.

According to the Figure 7 shown settings, the intonation determination algorithm uses a band-pass filter with Low_Frq = 50Hz, High_Frq = 220Hz and a Slope = 10Hz. The Hanning window has a 2048 sample width and is applied without overlapping. The used methods are the Autocorrelation and AMDF. Their

result should not diverge with more then 100Hz. Furthermore, some thresholds can be set for V-UV detection, etc.

Finally, there is a checkbox “Create pitchmarks”. This is an unfinished feature, which is a goal of the acoustic database automatic parameterisation.

6. TEXT ANALYSIS AND PROSODY

For the input text processing an xml database collects the needed rules. In order to test this, some parser tester functionalities had been added to our development environment.

The developed linguistic analyser and prosody generator is modular. Each module has a corresponding database. These easy extendable DBs contain: sentence type decision needed rules, linguistic rules, reference intonation curves, rules for labelling, tags, etc. The advantage of this architecture lays in the simplicity of its extension with extra rules.

7. CONCLUSIONS

There are several tools for various functionalities enumerated above, having an environment, which contains all of them in one utility, or having the standard functionalities together with the own TTS system specific needs in the same framework is the most important achievement.

Because the details of existing systems is not public, since scientists tend to create something unique and because these synthesis systems are strongly language specific, the phenomenon of “reinventing the wheel” is meat. This has the definite advantage of getting deep into topic, but for a good result a huge amount of effort is needed.

A development environment and the target synthesis system are evolving together. Of course the final goal is to have a standalone target synthesis system, therefore it’s carve out should be possible to be done easily. The costs and efforts needed for the realisation of the two systems is approximately the same. The investment seems to return, because the flexibility of research work is a great benefit. Besides in the future, this valuable toolset will be available any time.

The following issues have been solved:

- TTS specific basic file functionalities have been implemented (new file formats, database accesses).
- TTS specific User Interfaces have been developed (tool UI, visualizing of TTS specific parameters, intonation curves, temporal structure, graphical manipulating tools).
- Acoustic database processing tool has been developed.

- Our framework is applicable for different TTS Engine development.
- Numerous signal processing functionalities have been incorporated.
- Intonation curve computation, artificial prediction and generation are also part of our framework.

Open issues:

- A "nice to have" category is to build in a prosodic and linguistic descriptors or a prosody matrix visualizer, like in [4].
- Multiple selections and editing of pitchmarks are already under development.

REFERENCES

1. BODÓ Á.Zs., *Experiments for prosody modification using the Nonlinear Springing Method*, *Trends in Speech Technology*, 3rd Conference on Speech Technology and Human – Computer Dialogue (SpeD05), Cluj-Napoca, 2005, 177–181.
2. DASCĂLU-JINGA L., *Melodia vorbirii în limba română*, Academia Română, Univers Enciclopedic, Bucureşti, 2001, 27–31.
3. HOLMES J., HOLMES W., *Speech Synthesis and Recognition*, 2nd Edition, Taylor & Francis, London, 2001, 47–66, 93–108.
4. OLASZY G., NÉMETH G., OLASZI P. *et al*., *Profivox – A Hungarian Text-to-Speech System for Telecommunications Applications*, International Journal of Speech Technology (IJST), Kluver Academic Publishers, 2000, **3**, *3/4*, 201–215.
5. SPROAT R., *Multilingual Text-To-Speech Synthesis*, Kluwer Academic Publishers, 1998, 141–150.
6. VERHELST W.D.E., *An implementation of the PSOLA/KDG Waveform Synthesis Technique*, Eindhoven, Netherlands, 1990, 1–9.
7. WERNER S., KELLER E., *Prosodic aspects of speech*, in: *Fundamentals of Speech Synthesis and Speech Recognition: Basic Concepts, State of the Art, and Future Challenges* (E. Keller, editor), WileyBlackwell, Chichester, UK, 1994, 23–40.
8. *** Audacity, http://audacity.sourceforge.net/.

Received November 24, 2009

[1]*Sprint Consulting Ltd.,*
Technical University of Cluj-Napoca,
Faculty of Electronics, Telecommunications and Information Technology
[2]*Technical University of Cluj-Napoca,*
Faculty of Electronics, Telecommunications and Information Technology
Corresponding author: zsolt.bodo@sprintconsulting.com