CZECH TEXT-TO-SPEECH SYSTEM FOR A READING MACHINE

Peter Vepřek

Robotron Pty. Ltd., Australia

Abstract — In this paper, a Czech text-to-speech (TTS) system developed for a reading machine is presented. The whole TTS system was divided into rule- and lexicon-based text-to-phoneme conversion, prosody pattern calculation, rule-driven allophone selection, and linear-prediction-based speech production. Description of each of these components is given in the paper. The result is a complete yet compact TTS system that meets criteria laid out in the original project specifications.

INTRODUCTION

The purpose of this project was to design and develop a Czech text-to-speech system to provide speech output in the first Czech reading machine named *Sunrise*. The reading machine is a device resembling a flatbed scanner that can scan printed documents, perform page analysis, recognise characters, and read out the resulting text. Role of the TTS system is not only to read out the text but also to provide speech feedback to the user when he or she is operating the machine using a small keyboard.

The operating environment put some restrictions on implementation of the TTS system and also required some special functions to be provided by the TTS system. The whole system, realised fully in software, was to run as an embedded application on dedicated hardware. The amount of available memory was limited to 32 kB of RAM and 256 kB of ROM. Another limiting factor was the processor used which in this case was Intel 80188 running at 10 MHz. The synthesiser module could, however, take advantage of a DSP processor (ADSP2105) that was included on the board. The special functions required included immediate response, instantaneous speech abort, tight synchronisation for virtual cursor position backtracking used during text browsing, synchronous speech control sequences (SCSs) embedded in text, beep generation, communication over serial port, and others.

The rest of the paper is organized as follows. First, text-to-phoneme conversion is described and a set of phonemes occurring in Czech language is presented. It is followed by description of prosody pattern calculation, rule-driven allophone selection, speech signal representation, and compression techniques used. In the last section, system performance is discussed and future work is outlined.

TEXT-TO-PHONEME CONVERSION

The TTS system receives text to be spoken via a serial port where every character is represented by its 8-bit ASCII value. Czech language comprises all characters of the English alphabet plus a number of accented characters and a letter "ch". Representation of characters unique to Czech language is described in the following section.

Czech character representation

The extra Czech characters, not represented in standard ASCII, are assigned to hi-bit ASCII characters, not used by the OCR engine, according to a popular *Kamenický* code as shown in Table 1. The idea behind this mapping is that the selected hi-bit characters resemble the corresponding Czech characters.

In Czech, when letters "c" and "h" are adjacent they form a letter "ch" which is treated as a single character and not as two individual characters. This is important not only because the letter "ch" has a different pronunciation but it also affects sorting order. The letter "ch" comes after "h" in the alphabet. As implementation of this phenomenon is difficult in an otherwise English software environment, the *Kamenický* code does not handle this character. Internally, however, the speech engine can handle the "ch" character and assigns it its unique code.

Czech	á	č	ď	é	ě	í	ň	ó	ř	š	ť	ú	ů	ý	ž
ASCII	á	ç	â	é	ê	í	ñ	ó	⌐	¿	ƒ	ú	û	ÿ	æ
Code	160	135	131	130	136	161	164	162	169	168	159	163	150	152	145
Czech	Á	Č	Ď	É	Ě	Í	Ň	Ó	Ř	Š	Ť	Ú	Ů	Ý	Ž
ASCII	Â	Ç	à	É	ë	ï	Ñ	ò	₧	¢	å	ù	ù	¥	Æ
Code	143	128	133	144	137	139	165	149	158	155	134	151	151	157	146

Table 1. Mapping of Czech characters into ASCII characters according to *Kamenický* code.

Set of phonemes

A list of 45 phonemes occurring in Czech language was made and every phoneme was assigned a two-character code as shown in Table 2, where all phonemes are divided into groups according to their categories. Letters in parenthesis show the most common orthographic representation of each phoneme and words in parenthesis give simple examples.

front vowels (sort + long)	EE *(e: te**t**a)*, II *(i,y: p**i**tí)*, EX *(é: p**é**ci)*, IX *(í,ý: s**í**to)*
middle vowels (sort + long)	AA *(a: t**a**dy)*, AX *(á: t**á**ta)*
back vowels (sort + long)	OO *(o: k**o**tě)*, UU *(u: p**u**sa)*, OX *(ó: s**ó**lo)*, UX *(ú,ů: b**ů**ček)*
diphthongs	AJ *(aj: t**aj**ný)*, EJ *(ej: p**ej**sek)*, OJ *(oj: b**oj**kot)*, AU *(au: **au**to)*, OU *(ou: c**ou**ra)*
schwa	EA *(-: krk, smrk)*
semivowels (liquid + glide)	LL *(l: **l**aso)*, JJ *(j: **j**aro)*
nasals	MM *(m: **m**áma)*, NN *(n: **n**aše)*, NJ *(ň: bu**ň**ka)*, NG *(n: ba**n**ka)*
voiced stop consonants	BB *(b: **b**eton)*, DD *(d: **d**álka)*, DJ *(ď,d: **d**ítě)*, GG *(g: **g**uma)*
unvoiced stop consonants	PP *(p: **p**ytel)*, TT *(t: **t**ele)*, TJ *(ť,t: **t**ílko)*, KK *(k: **k**olo)*
voiced fricatives	VV *(v: **v**áza)*, ZZ *(z: **z**ima)*, ZH *(ž: **ž**ena)*
unvoiced fricatives	FF *(f: **f**ůra)*, SS *(s: **s**ako)*, SH *(š: **š**ála)*
affricates	CC *(c: **c**ena)*, CH *(č: **č**ára)*
whispers (voiced + unvoiced)	HH *(h: **h**udba)*, KH *(ch: **ch**ata)*
other consonants and clusters	RR *(r: **r**ozum)*, RH *(ř: **ř**eka)*, KS *(ks,x: ko**ks**)*, KV *(kv: **kv**as)*, TR *(tř: **tř**i)*

Table 2. Set of Czech phonemes.

Text-to-phoneme rules and lexicon

Czech is a very phonetic language with a fairly straightforward relationship between its orthographic and phonetic representations. Therefore, there are only about 170 text-to-phoneme rules that translate text into phonemes including phonetic spelling of single isolated characters. These rules are

complemented by a small lexicon that handles mainly abbreviations, words accepted from other languages, and foreign words.

An example of text-to-phoneme rules is shown in Table 3 where rules for letter "D" are listed. Symbols '@W' and '@@' are control phonemes representing weakening of phonemes following the symbol and resetting back to normal, respectively. Character ' ' (space) within the text context matches any separator (non-letter); '#' one or more vowels; '*' one or more consonants; and '&' one of C, Č, F, CH, K, P, S, Š, T, Ť.

" [DO] "	@W DD OO @@
" [D] "	DD EX
"[D]ĺ"	DJ
"[D]í"	DJ
"[D]Ě"	DJ
"[D]&"	TT
"#[D] "	TT
"*[D] "	TT
"[D]"	DD

Table 3. "D" rules.

Number translation

Number translation is partially-table driven. It can handle integer and real numbers, recognises dates, times, etc., and takes into account word declension.

Special translation modes

In a special, user selectable, mode punctuation marks can be translated in four different levels: none, some, most, and all, or interpreted as mathematical signs whenever appropriate. For example, a character "*/*" may be pronounced as "*lomítko*" (slash) or as "*děleno*" (divided by) or it may not be pronounced at all. Separate mode setting controls so called grouping mode. When the grouping mode is enabled, equal consecutive characters are grouped together rather than pronounced individually. For example, "*****" may be pronounced as "*pět hvězdiček*" (five stars) rather than "*hvězdička hvězdička hvězdička hvězdička hvězdička*" (star star star star star).

Prosody Pattern Calculation

Prosody (fundamental frequency, duration, and intensity) pattern is calculated on sentence-by-sentence basis. Every sentence is first divided into its constituent clauses and phrases according to (i) punctuation marks that are frequently and consistently used in Czech language, and according to (ii) a list of common conjunctions and prepositions. The falling baseline is then established and it is added to the default pitch. The difference between the pitch at the start and at the end of the clause is proportional to the number of phonemes in that clause. It is, however, limited by its maximum allowable value. The pitch drop of the baseline is spread over the first two thirds of the clause and the pitch remains constant in the last third. According to the following rules, changes are than superimposed onto the falling baseline. These rules are illustrated in Figure 1 and include:

1) Drop of the pitch, slower speech rate, and slight weakening at the end of a clause. The extent to which the duration, pitch, and intensity are modified depends on type of the clause.
2) Emphasizing of the first syllable of every word.
3) Emphasizing of long vowels.
4) Slight pitch drop at the end of every word.
5) Weakening of some phonemes according to the lexicon.

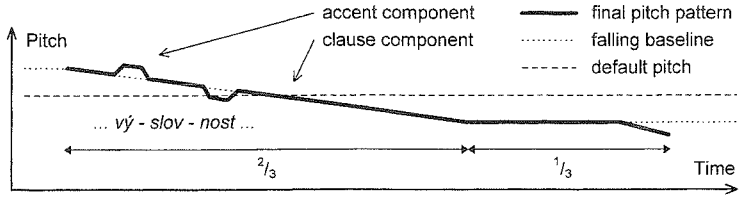

Figure 1. Pitch pattern of a sentence.

ALLOPHONE SELECTION

During speech production allophones, context-dependent variants of phonemes, are selected from a set of all available allophones using rules similar to text-to-phoneme rules. These rules, however, translate individual phonemes or groups of phonemes into allophones according to their phonemic context.

General form of a rule is "LPC [PS] RPC", where LPC is a left phonemic context, PS is a phoneme string (a single phoneme or a phoneme cluster), and RPC is a right phonemic context. Both the left and right phonemic contexts can contain class symbols, such as '*S' that represents any stop consonant, and optional parts enclosed in parenthesis. An example of such a rule is:

"(__) [DD AA] *S" -> "(DA7-1) DA7-2".

This rule will replace any initial syllable "DD AA" followed by a stop consonant with "DA7-1 DA7-2" and the same syllable in non-initial position followed by a stop consonant with "DA7-2".

The rules together with the set of all allophones were build up gradually in the following steps. (i) First, the most common allophone for each phoneme was included. (ii) Then, allophones with respect to the position of the phoneme within a word (initial, middle, or final) were added. (iii) In the next step, allophones in a particular phonemic environment were added (i.e. consonant allophones depending on whether the consonant was followed by a front, middle or back vowel; or vowel allophones depending on whether the vowel was followed by a nasal, stop consonant etc.). (iv) And last, groups of phonemes (syllables or phoneme clusters that exhibit a strong coarticulation) were assigned to the appropriate groups of allophones. Memory requirements, though, limited the total number of allophones and their clusters that could be included in the system. For this reason, the frequency of occurrence of a particular allophone or allophone cluster in spoken language and intelligibility improvement were accepted as criteria for adding new allophones. The intelligibility was evaluated subjectively using two native listeners.

SPEECH SIGNAL REPRESENTATION

Speech production was modelled using linear prediction (LP) model implemented as an all-pole lattice filter. This model was chosen because of its high compression ratio, flexibility to easily change the rate, pitch, and intensity of the produced speech signal, and readiness of analysis/resynthesis methods. Parameters of the LP model were obtained by analysing recordings of a model speaker (low pitched male). The recorded speech signal was first sampled at 11 kHz and divided into frames. Then, any signal offset was removed, the signal was pre-emphasized using the optimal pre-emphasis coefficient (Rabiner & Schafer, 1978), an LP analysis was performed, and each frame was represented by its duration, pre-emphasis coefficient, gain, voicing, and a set of partial correlation (PARCOR) coefficients. To further reduce memory requirements the following techniques were used: (i) frame lengthening, (ii) elimination of similar frames, (iii) elimination of interpolable frames, (iv) elimination of redundant PARCOR coefficients, and (v) parameter quantisation.

Frame lengthening

Initially, frame length was set to 10 ms (not including 7 ms overlaps into to the two adjacent frames) and an LP analysis was performed. The frame was then *lengthened*, the frame length was incremented by 1 ms, and the second LP analysis was performed. The two obtained sets of LP coefficients were then compared by calculating the differences between values of their corresponding LP spectra sampled every 50 Hz. The differences were perceptually weighted so that higher frequencies had lower weights and the differences were all added up. When the total difference was below a threshold, the two sets of LP coefficients were considered very similar and the speech signal was considered stationary within the lengthened frame. The process of frame lengthening was repeated until the spectrum of the lengthened frame changed significantly or until a change of voicing or a significant change of gain or pre-emphasis coefficient were detected. At this point the obtained coefficients, frame duration, and other parameters were stored. The point of analysis was moved by the frame length, the frame length was reset back to 10 ms and the whole process was repeated. Using this algorithm, the stationary portions of the signal were represented by longer frames resulting in fewer parameters.

Elimination of similar frames

Once the speech signal was completely transformed into a sequence of variable length frames, spectra of every two adjacent frames were compared using the method described above. When a pair of *similar* frames was detected, the shorter of the two frames was removed and the duration of the longer frame was incremented by the duration of the removed frame as illustrated in Figure 2. This situation, however, was not common due to the already variable frame length.

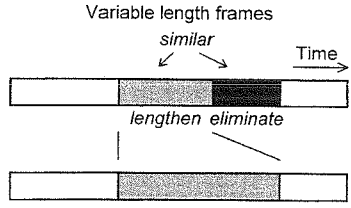

Figure 2. Elimination of *similar* frames.

Elimination of interpolable frames

During this step, frames were assessed three at a time. If the spectrum of the second (middle) frame was sufficiently similar to the spectrum of a frame that was obtained as a half-way interpolation between the first and the third frames, the middle frame was considered *interpolable* from the two adjacent frames and therefore redundant. This frame was then removed and the duration of both adjacent frames was incremented by half of the duration of the removed frame as illustrated in Figure 3.

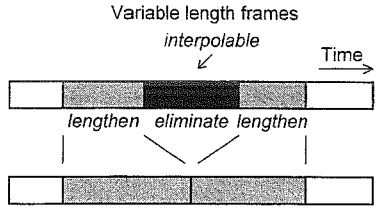

Figure 3. Elimination of *interpolable* frames.

Elimination of redundant PARCOR coefficients

Up to this point, every frame had a constant number of PARCOR coefficients given by the sampling frequency (i.e. 15 coefficients for F_s = 11 kHz). It is well-known that the fixed number of coefficients is unnecessarily high for some configurations of the vocal tract especially for unvoiced sounds. The unnecessary coefficients were removed using the following algorithm. First, a target LP spectrum given by all (15) coefficients was calculated. Then a spectrum given by one, two, three ... coefficients was calculated (see Figure 4). Once a spectrum sufficiently similar to the target one was encountered, the process stopped giving the minimum number of coefficients needed to represent the frame accurately.

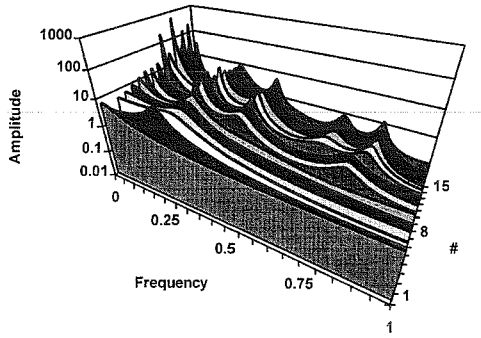

Figure 4. Frame spectrum as a function of the number of PARCOR coefficients.

Parameter quantization

All of the frame parameters were quantised based on work done by Itakura & Saito (1972), Viswanathau & Markhoul (1975), Gray & Markel (1976), and on GSM 06.10 recommendations. The final quantisation, however, was a compromise between a high compression ratio and an easy parameter accessibility. The duration, pre-emphasis coefficient, gain, voicing, and number of PARCOR coefficients were all packed into 5 bytes (37 bits). The first four PARCOR coefficients were allocated 12 bits each and the remaining coefficients 8 bits each. If, due to the quantisation, the last one or more coefficients became zero, the coefficients were not stored and the number of coefficients was adjusted accordingly.

SPEECH SYNTHESIS

The speech signal was generated by the DSP processor mentioned earlier using an all-pole lattice filter. Frame parameters were interpolated between adjacent frames. To improve intelligibility by removing undesired pops and clicks, impulses of the voiced excitation were not generated immediately before or after the unvoiced excitation (i.e. when excitation changed from voiced to unvoiced and vice versa). In order to do this, there was a one-frame delay during the synthesis so that the voicing of the following frame would be known. Speech samples were generated at a rate corresponding to the sampling frequency used (11 kHz). In order to improve the acoustic quality of the signal, in particular to reduce the perceived metallic characteristic of the signal, speech samples were output at double the rate. The oversampling effect was achieved by inserting linearly interpolated samples in between the generated samples.

DISCUSSION AND FUTURE WORK

The whole Czech text-to-speech system designed and developed for a particular application has been presented. The main emphasis during the development was put on intelligibility and practical realisation of the system. The result is a complete yet compact text-to-speech system with minimal hardware requirements that meets criteria laid out in the original project specifications.

Implementation of the TTS system was optimised in order to reduce computational and memory requirements. This included efficient search techniques, pointer and array index optimisation, locating and coding of bottleneck procedures in assembler and other techniques.

An LP speech production model was chosen because of the high compression ratio, the flexibility to easily change speech rate, pitch, and intensity, and the availability of analysis/resynthesis algorithms. Future work will concentrate on improving the naturalness of the speech output and overcoming some of the limitations of the linear prediction model i.e. the difficulty of capturing short term events such as bursts, and modelling of nasals and some fricatives that require not only poles but also zeroes in the transfer function of the model to authentically capture the spectrum.

On part on the text-to-phoneme conversion, number translation should take into account noun gender to correctly translate numbers e.g. "***1** stůl*" (1 table, masculine noun) should be translated as "***jeden** stůl*" but "***1** židle*" (1 chair, feminine noun) as "***jedna** židle*" etc. Also, more sophisticated sentence parsing technique needs to be developed to assist in prosody pattern calculation.

ACKNOWLEDGMENT

My thanks go to Mr. M. Hudeček, the managing director of Robotron Pty. Ltd., for his help and support during this project.

REFERENCES

Gray, A. H. & Markel, J. D. (1976) *Quantization and bit allocation in speech processing*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, pp. 459-473.

Itakura, F. & Saito, S. (1972) *On the optimum quantization of feature parameters in the PARCOR speech synthesizer*, IEEE Conference on Speech Communication and Processing, pp. 434-437.

Rabiner, L. R. & Schafer, R. W. (1978) *Digital processing of speech signals*, (Englewood Cliffs, N.J.: Prentice Hall).

Viswanathau, R. & Markhoul, J. (1975) *Quantization properties of transmission parameters in linear predictive systems*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 23, pp. 434-446.

Interim European Telecommunication Standard I-ETS 300 036 (GSM 06.10), *European digital cellular telecommunications system (phase 1); full-rate speech transcoding*, ETSI, March 1992.