I. Nowak, M. Owsianny, J. Imiołczyk

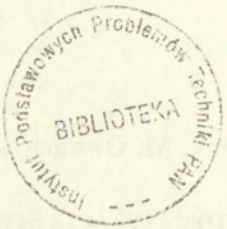
AUTOMATYZACJA STEROWANIA PARAMETRAMI AKUSTYCZNYMI W CYFROWYM SYNTEZATORZE MOWY

2/1995

WARSZAWA 1995

ISSN 0208-5658

Praca wpłynęła do Redakcji dnia 19 grudnia 1994 r.



Na prawach rękopisu

Instytut Podstawowych Problemów Techniki PAN
Nakład 100 egz. Ark. wyd. 1,50 Ark. druk. 2,0
Oddano do drukarni w styczniu 1995 r.

Wydawnictwo Spółdzielcze sp. z o.o.
Warszawa, ul. Jasna 1

Ignacy Nowak
Mariusz Owsianny
Janusz Imiołczyk
Zakład Fonetyki Akustycznej IPPT PAN
Poznań

AUTOMATYZACJA STEROWANIA PARAMETRAMI AKUSTYCZNYMI W CYFROWYM SYNTEZATORZE MOWY[1]

Streszczenie

W pracy przedstawione są rezultaty zmian dokonanych w implementacji modelu formantowego, szeregowo-równoległego syntezatora mowy, pracującego wg systemu D.H.Klatta ([5],[6]). Wersja bazowa implementacji została opracowana w ZFA IPPT PAN w Poznaniu przez M.Owsiannego ([7]). Modyfikacja przeprowadzona została w kilku kierunkach: 1.dostosowania sterowania modelem do syntezy złożonych sekwencji głosek polskich; 2.przyspieszenia działania programowego modułu syntezy; 3.poprawienia interfejsu użytkownika podczas pracy z programem. W wyniku wykonania głównej części zadania (p.1 powyżej) osiągnięte zostały: a) możliwość wielokrotnego (w ramach jednej wypowiedzi syntetycznej), bezkonfliktowego przełączania trybów syntezy; b) uproszczenie układu decyzyjnego przełączającego tryby syntezy; c) zadowalająca precyzja sterowania parametrami przy przechodzeniu procesu syntezy do kolejnych ramek czasowych (fram). Poza tym osiągnięto wydatne skrócenie czasu trwania syntezy przy zachowanej w pełni jakości wygenerowanej mowy syntetycznej, a także znaczną poprawę komfortu pracy użytkownika z programem syntezatora. Nowy program syntezatora stawia podwyższone wymagania sprzętowe: minimalną konfiguracją jest komputer IBM PC 386 z koprocesorem arytmetycznym.

1. Wstęp.

Praca niniejsza jest pośrednio rezultatem osiągnięcia przez jej autorów pewnego punktu granicznego na kierunkach prowadzonych dotąd prac eksperymentalno-badawczych. J.Imiołczyk oraz I.Nowak w latach 1990-1993 zajmowali się zagadnieniami syntezy mowy z

---

[1]) Praca wykonana w ramach zlecenia IPPT nr 372

wykorzystaniem układów scalonych, realizujących hardware'owo modele syntezatora typu źródło-filtry. Prace J.Imiołczyka w pierwszej części tego okresu doprowadziły do powstania zbioru wzorców głosek dla języka polskiego. Stosowano przy tym specjalnie w tym celu przygotowany program edytora parametrów dla sprzętowego syntezatora mowy MEA 8000, a następnie dla syntezatora PCF 8200 - obydwa układy produkcji f-my PHILIPS. Prowadzone równolegle przez I.Nowaka prace projektowo-programowe pozwoliły na względnie szybkie przygotowanie podstaw systemu syntezy mowy z tekstu dla języka polskiego i dalsze tego systemu udoskonalanie w trzyosobowym zespole (z udziałem G.Demenko) - p. [2], [3], [4]. Zebrane przy tym doświadczenia pozwoliły zrozumieć, jak istotne mogą okazać się ograniczenia, wbudowane niejako w sprzętowo realizowaną syntezę mowy. Projekty badań, projekt samego systemu syntezy, a także przygotowane oprogramowanie stają się bowiem bardzo mocno związane z protokółem wymiany informacji z konkretnym układem scalonym, a ewentualne rozpowszechnianie wyników badań zależy od dostępności hardware'u na rynku komercyjnym. Ewentualne przeniesienie wyników badań na nowy układ scalony wymaga nowych prac badawczych (nowe zbiory wzorców głosek) oraz poważnej często przebudowy algorytmów i realizujących je programów. Należy dodać do tego fakt, że konstrukcja syntezatora uniemożliwia uzykanie mowy o wysokim stopniu naturalności (chociaż można osiągnąć wysoką zrozumiałość) oraz mowy o określonym brzmieniu (np. głosy kobiece lub dziecięce). Drogą wiodącą do przezwyciężenia wymienionych ograniczeń jest implementacja modelu w pełni cyfrowego syntezatora mowy. Znaczny wzrost mocy obliczeniowej komputerów, jaki nastąpił w ostatnich latach, zapewnia perspektywę zadowalającej szybkości działania takiego syntezatora.

W wyniku prowadzonych w tym samym okresie prac, M.Owsianny zrealizował programową implementację cyfrowego, szeregowo-równoległego syntezatora mowy, działającego według modelu D.H.Klatta ([1], [5], [6]). Program ten był stale ulepszany, zostały także wprowadzone pewne udoskonalenia do samego modelu syntezy (np. precyzyjnie działający generator tonu krtaniowego). Opis ostatecznej wersji syntezatora zrealizowanej przez

M.Owsiannego (program otrzymał nazwę SMOK), wraz z odniesieniem do oryginalnego modelu D.H.Klatta, można znaleźć w [7] i [8]. Prace badawcze prowadzone przez M.Owsiannego z użyciem programu SMOK wykazały, że możliwe jest wytworzenie za jego pomocą mowy o wysokim stopniu naturalności ([7]); nie stanowi też problemu uzykanie syntetycznego głosu kobiecego ([8]). Synteza w programie SMOK działa jednakże dość wolno - nie wykorzystano w pełni możliwości sprzętowych komputera, na którym był uruchamiany, gdyż pierwotnym zamiarem autora SMOKa było zapewnienie możliwości pracy programu także na słabszym sprzęcie.

Próbę - do pewnego stopnia udaną - zmodyfikowania postaci modułu syntezy poprzez asembleryzację jego kodu źródłowego z równoczesnym przejściem na obliczenia wyłącznie stałoprzecinkowe na komputerze 32-bitowym (IBM PC 386), przeprowadził B.Szutowski ([9]). Ceną za bardzo znaczne (kilkukrotne) przyspieszenie działania syntezy była jednak utrata jakości generowanej mowy. Okazało się bowiem, że obliczenia, wykonywane pierwotnie w formacie zmiennoprzecinkowym, przy przejściu na format stałoprzecinkowy wymagają ciągłego przeskalowywania wyników pośrednich, co stanowi problem wymykający się częściowo spod kontroli programu.

Odrębne zagadnienie stanowiło udoskonalenie procesu sterowania parametrami akustycznymi syntezatora SMOK. W dotychczasowych zastosowaniach badawczych program działał bardzo precyzyjnie podczas syntezy samogłosek izolowanych (główna część badań M.Owsiannego dotyczyła właśnie samogłosek - [8]), jednak nie dawał zadowalających efektów w przypadku wypowiedzi o złożonej strukturze.

W rezultacie postanowiono przeprowadzić skrupulatną analizę modelu syntezy zaimplementowanego w SMOKu, zwłaszcza tej jego części, która odpowiadała za proces sterowania syntezą. Ponieważ działanie programu SMOK w zakresie syntezy samogłosek zostało już wystarczająco sprawdzone w badaniach M.Owsiannego (p.wyżej), w pracach nad udoskonaleniem syntezatora postanowiono uznać to zagadnienie za rozwiązane i zająć się pozostałymi klasami głosek.

Z drugiej strony - dla uzyskania tempa syntezy porównywalnego z wersją stałoprzecinkową B.Szutowskiego - I.Nowak

dokonał ponownie kompleksowej asembleryzacji modułu syntezy, lecz tym razem - z użyciem koprocesora arytmetycznego, co pozwoliło utrzymać bez zmian wysoką jakość generowanej mowy. Ostatecznym efektem wykonanych prac jest program SMOKIN1.

Kolejne części pracy przedstawiają wersje modelu i programu syntezy, które były punktami startowymi dla wykonanych modyfikacji, a następnie omówione są kolejno zmiany zrealizowane w odniesieniu do różnych aspektów danego modelu syntezatora. Osiągnięte ostatecznie rezultaty przedstawiono w części 9.

2. Baza teoretyczna opracowania - krótka prezentacja dotychczasowej wersji modelu syntezatora.

Zgodnie z generalnym założeniem teretycznym dla formantowego, szeregowo - równoległego modelu syntezatora mowy, opracowanego przez D.H.Klatta ([1],[5]), syntetyczny sygnał mowy ma w dziedzinie częstotliwości widmo **P(f)** o postaci będącej iloczynem trzech funkcji: **S(f)** - widma źródła pobudzającego kanał głosowy, **T(f)** - funkcji transmitancji kanału głosowego oraz **R(f)** - charakterystyki promieniowania ust:

$$P(f) = S(f) \cdot T(f) \cdot R(f) \qquad (1)$$

W modelu, utworzonym przez D.H.Klatta, którego zmodyfikowana przez M.Owsiannego wersja ([7]) posłużyła za punkt startowy niniejszego opracowania, rolę poszczególnych funkcji pełnią wyróżnione bloki bądź układy bloków tego modelu. Schemat blokowy modelu w wersji opublikowanej w [7], przedstawia rys.1.

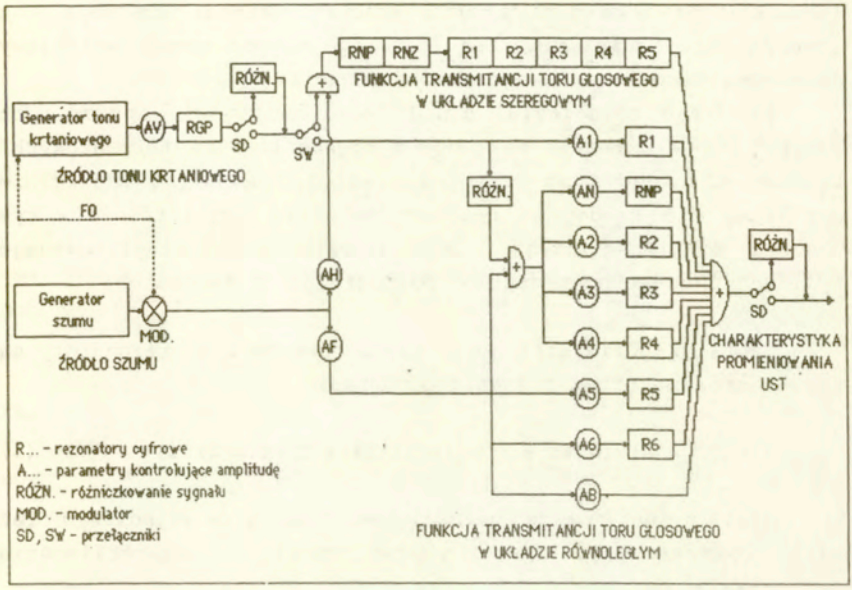
W schemacie tym źródło pobudzające reprezentowane jest przez współdziałające ze sobą układy:

1) **Generator tonu krtaniowego**, opisywanego dla pojedynczego okresu wzorem:

$$U(t) = a \cdot t^2 - b \cdot t^3 \quad \text{dla } t \in \langle 0, a/b \rangle \text{ (faza otwarta)} \qquad (2)$$
$$U(t) = 0 \quad \text{dla pozostałych } t \text{ (faza zamknięta)}$$

gdzie t reprezentuje czas, natomiast współczynniki **a** i **b** zależą od amplitudy tonu krtaniowego i względnego czasu trwania fazy otwartej pulsu krtaniowego.

Widmo tonu krtaniowego może opadać mniej lub bardziej stromo wskutek modyfikacji w rezonatorze RGP, a końcowy wynik pracy tego bloku modelu syntezy może (opcjonalnie) zostać poddany różniczkowaniu .



Rys. 1. Schemat blokowy software'owego szeregowo-równoległego formantowego syntezatora mowy, wg D.H.Klatta, z modyfikacjami M.Owsiannego.

2) **Generator szumu**, w którym wykorzystuje się fakt, iż sygnał, który w przebiegu czasowym ma rozkład zbliżony do normalnego, daje w dziedzinie częstotliwości widmo prawie płaskie. Dodatkowo, szum może być poddany modulacji prostokątnej ze współczynnikiem 50% (układ MOD) w przypadku, gdy generowany jest równolegle z tonem krtaniowym (odcinek dźwięczny); okres modulacji jest wtedy identyczny z okresem tonu krtaniowego.

Uzyskana postać pobudzenia (z modulacją lub bez) wykorzystywana jest jako podstawa do wytworzenia zarówno szumu aspiracyjnego jak i frykacyjnego.

Funkcja transmitancji kanału głosowego realizowana jest za pomocą dwóch układów rezonatorów:

1) **Układ szeregowy** (kaskadowy), realizowany dla 4-6 formantów oraz bieguna i zera (antyrezonatora) nosowego - z zadanymi szerokościami wstęg (dla tego układu sygnał wejściowy jest sumą tonu krtaniowego oraz aspiracji).

2) **Układ równoległy**, dla którego na wejściu oprócz tych samych jak w układzie szeregowym częstotliwości formantowych, bieguna nosowego oraz szerokości wstęg, zadane są oddzielnie amplitudy dla każdego z rezonatorów. Poza tym istnieje w tym układzie możliwość dodania do sygnału wyjściowego zmodyfikowanego aplitudowo szumu pochodzącego bezpośrednio z generatora.

Tak w układzie szeregowym jak i równoległym, rezonatory są reprezentowane przez równania o postaci:

$$y(n\cdot T) = A\cdot x(n\cdot T) + B\cdot y((n-1)\cdot T) + C\cdot y((n-2)\cdot T) \qquad (3)$$

gdzie **n** jest numerem próbki, **x(k)** - sygnałem wejściowym oraz **y(k)** - sygnałem wyjściowym dla k-tej próbki, **T** - częstotliwością próbkowania.

Współczynniki **A,B,C** dla rezonatorów układu szeregowego pozostają w następującej zależności od częstotliwości formantowych **F** i szerokości wstęg **BW** :

$$C = -\exp(-2\cdot\pi\cdot BW\cdot T) \qquad (4)$$
$$B = 2\cdot\exp(-\pi\cdot BW\cdot T)\cdot\cos(2\cdot\pi\cdot F\cdot T)$$
$$A = 1-B-C$$

Dla każdego rezonatora układu równoległego współczynniki **A, B** i **C** obliczane są także według (4), po czym współczynnik **A** pomnożony zostaje przez wartość odpowiadającą amplitudzie danego

rezonatora.

W tej samej konwencji zapisu, równanie antyrezonatora ma postać:

$$y(n\cdot T) = A'\cdot x(n\cdot T) + B'\cdot x((n-1)\cdot T) + C'\cdot x((n-2)\cdot T) \qquad (3')$$

gdzie A',B',C' związane są z A,B,C zależnościami:

$$A' = 1/A, \quad B' = -B/A, \quad C' = -C/A \qquad (4')$$

W wersji przedstawionej na rys.1 układy szeregowy i równoległy mogą ze sobą współdziałać dla utworzenia kolejnej próbki syntezowanego sygnału (wtedy kaskada wylicza składową dźwięczną, po czym zeruje wartość wejściową pochodzącą ze źródła pobudzenia dźwięcznego, wskutek czego układ równoległy wylicza tylko składową szumową związaną z frykacją). Odpowiada to stanowi "0" przełącznika SW. Przy stanie "1" wszystkie sygnały ze źródeł pobudzenia kierowane są do bloku syntezy równoległej.

Najmniej rozbudowany jest w omawianym modelu blok realizujący charakterystykę promieniowania ust - obróbka sumy sygnałów przychodzących z bloku szeregowego i równoległego sprowadza się do **warunkowego różniczkowania** (przełącznik SD) wartości wejściowej i skierowania jej na wyjście modelu.

3. Baza praktyczna opracowania - dotychczasowa wersja programu syntezy.

Tak ogólnie naszkicowane zasady działania syntezatora nie dają wystarczającego wyobrażenia o szczegółach zawartych w konkretnej implementacji danego modelu, zrealizowanej przez M.Owsiannego w programie komputerowym SMOK (1992). W ramach sesji pracy z tym programem możliwa jest edycja parametrów syntezy, wykonanie samej syntezy, wprowadzenie wypowiedzi poprzez przetwornik analogowo-cyfrowy (np. z mikrofonu lub magnetofonu), odtworzenie dowolnego fragmentu wytworzonej (syntetycznej) lub

wprowadzonej wypowiedzi, dokonanie porównania dwóch wypowiedzi (np. naturalnej i syntetycznej) - w tym porównanie widm chwilowych, konkatenacja i sumowanie postaci czasowych opracowywanych sygnałów oraz zapis/odczyt wyników do/z plików dyskowych. Kompletny opis programu SMOK znaleźć można w [7]. Z punktu widzenia potrzeb niniejszego opracowania, koniecznym wydaje się tutaj jedynie podanie struktury parametrów syntezy w SMOKu.

Zbiór danych do syntezy obejmuje listę **13 parametrów stałych**, których wartości pozostają niezmienne na całym odcinku czasowym danej wypowiedzi, oraz pewną liczbę ramek czasowych (fram), opisujących - każda w postaci listy **25 parametrów framowych** - chwilowe właściwości akustyczne syntezowanego sygnału mowy.

Lista parametrów stałych:

1. **DU [ms]** (duration of the utterance) - czas trwania całej wypowiedzi.
2. **SW** (switch cascade/parallel) - przełącznik trybu syntezy.
3. **SD** (glottal pulse, waveform diff. switch) - przełącznik różniczkowania tonu krtaniowego oraz postaci końcowej sygnału.
4. **SR** (sampling rate) - częstotliwość próbkowania.
5. **NWS** (number of waveform samples per chunk) - liczba próbek w jednej ramce czasowej (framie).
6. **NFC** (number of cascaded formants) - liczba formantów w szeregowym trybie syntezy.
7. **G0 [dB]** (overall gain control) - poziom ogólnego wzmocnienia sygnału.
8. **FL [%]** (random fluctuation in F0) - zmiany stochastyczne w przebiegu częstotliwości podstawowej, mają na celu unaturalnienie wypowiedzi.
9. **FNP [Hz]** (nasal pole frequency) - częstoliwość formantu nosowego.
10. **BNP [Hz]** (nasal pole bandwidth) - szerokość wstęgi formantu nosowego.

11. **BNZ [Hz]** (nasal zero bandwidth) - szerokość wstęgi antyformantu nosowego.
12. **AN [dB]** (nasal formant amplitude) - amplituda formantu nosowego.
13. **A1** (first formant amplitude) - amplituda pierwszego formantu (w równoległym trybie syntezy).

Lista parametrów framowych:

1. **F0 [Hz]** (fundamental frequency of voicing) - częstotliwość podstawowa pobudzenia dźwięcznego.
2. **AV [dB]** (amplitude of voicing) - aplituda pobudzenia dźwięcznego.
3. **OQ [%]** (open quotient) - faza otwarta okresu tonu krtaniowego.
4. **TL [dB]** (extra tilt of voicing spectrum) - dodatkowy spadek obwiedni widma tonu krtaniowego (na oktawę).
5. **AF [dB]** (amplitude of frication) - amplituda szumu frykacyjnego.
6. **AH [dB]** (amplitude of aspiration) - amplituda szumu aspiracyjnego.
7. **FNZ [Hz]** (nasal zero frequency) - częstotliwość formantu nosowego.

8-13. **F1..F6 [Hz]** (first..sixth formant amplitude) - częstotliwości formantów.

14-19. **B1..B6 [Hz]** (first..sixth formant bandwidth) - szerokości wstęg formantowych.

20-24. **A2..A6 [dB]** (second..sixth formant amplitude) - amplitudy formantów (w równoległym trybie syntey).

25. **AB [dB]** (bypass path amplitude) - amplituda szumu dodawanego bezpośrednio z generatora (w równoległym trybie syntezy).

W module syntezy programu SMOK główną pętlę stanowi cykl obliczeń dla kolejnej framy. W każdym kolejnym cyklu wywoływane są kolejno procedury AMPL i CREZ, przy czym:

1) Procedura AMPL wylicza wartości amplitudowe, współczynniki rezonatorów formantowych oraz - warunkowo (dla odcinków dźwięcznych) - współczynniki rezonatora dla tonu krtaniowego.

2) Procedura CREZ, korzystając z danych wyjściowych procedury AMPL, tworzy w swojej wewnętrznej pętli k=NWS próbek postaci czasowej sygnału mowy.

Główna pętla wywoływana jest tyle razy, ile wynosi liczba fram danej wypowiedzi (obliczana na podstawie DU, SR oraz NWS).

W dalszych częściach niniejszego opracowania wystąpią liczne odwołania zarówno do modelu teoretycznego syntezy, jak i przedstawionej powyżej programowej implementacji syntezatora. Przedstawienie wstępnej klasyfikacji dokonanych zmian oraz głównych celów realizowanych poprzez te zmiany ułatwi systematyczną prezentację wyników. Natomiast objaśnienia dotyczące szczegółów działania syntezatora podane zostaną w kolejnych punktach pracy.

4. Modyfikacja a automatyzacja w syntezatorze: relacje między obydwoma pojęciami.

Zagadnienia modyfikacji systemu syntezy i automatyzacji sterowania parametrami akustycznymi w przedstawianym tutaj cyfrowym syntezatorze mowy są ze sobą silnie powiązane. W celu uniknięcia mogących wystąpić nieporozumień ustalone zostaną poniżej - na użytek tej prepublikacji - zakresy obydwu pojęć oraz ich wzajemne zależności.

Trzeba podkreślić, iż wszystkie działania (jakkolwiek rozumiane) w sferze modyfikacji oraz automatyzacji systemu syntezy mają wspólny cel: jest nim **doprowadzenie programu do stanu umożliwiającego efektywną pracę badawczą z jego zastosowaniem**. Główne zadanie badawcze, realizowane za pomocą danego programu - synteza alofonów języka polskiego - wymaga bowiem tworzenia i przechowywania znacznej liczby zbiorów danych,

szybkiego dostępu do tych danych oraz możliwości łatwej ich edycji. Sygnał mowy syntetycznej wygenerowany na podstawie danych parametrycznych powinien odznaczać się wysoką jakością, a program powinien umożliwiać tworzenie krótkich wypowiedzi (wyrazy, logatomy) z praktycznie dowolnych sekwencji głosek. Narzuca to konieczność zachowania szczególnej precyzji działania syntezatora na odcinkach przejść między kolejnymi głoskami. Na komfort pracy z programem wpływa przy tym również czas realizacji syntezy edytowanej wypowiedzi.

Termin *modyfikacja* odnosić się będzie do zmian dokonanych w syntezatorze mowy na trzech różnych płaszczyznach, oznaczonych symbolami **(M)**, **(S)**, **(I)**:

(M) Zmiany w cyfrowym modelu syntezy - np. inne zasady wyboru szeregowego / równoległego trybu syntezy, zmiana postaci wzoru obliczenia danego rezonatora, zmiana liczby i charakteru parametrów framowych (*modyfikacja modelu syntezy*).

(S) Wszystkie zmiany w programie syntezy wpływające na szybkość i precyzję wykonywania samego procesu syntezy - zwłaszcza asembleryzacja kluczowych procedur (*modyfikacja modułu syntezy*).

(I) Inne zmiany w programie syntezy, poprawiające ogólny poziom obsługi użytkownika przez program, lecz nie mające wpływu na jakość i szybkość syntezy (*modyfikacja interfejsu użytkownika*).

Modyfikacja modułu syntezy **(S)** musi być zgodna ze zmianami dokonanymi w samym modelu syntezy **(M)**. Na etapie tworzenia programu wielokrotnie dokonywano też modyfikacji samego modelu syntezy na podstawie obserwacji działania algorytmu w konkretnej wersji programu. **(M)** i **(S)** pozostają zatem względem siebie w układzie sprzężenia zwrotnego. Z tego powodu zagadnienia związane z **(M)** i **(S)** będą zazwyczaj przedstawiane łącznie.

Modyfikacja interfejsu użytkownika (I) nie ma bezpośredniego związku z modelem syntezatora ani z usprawnianiem działania algorytmu syntezy, odgrywa jednak bardzo istotną rolę w procesie przygotowania programu do roli efektywnego narzędzia badawczego.

Założonym celem jest, jak powiedziano wyżej, osiągnięcie przydatności syntezatora do przygotowania podstawowych alofonów wszystkich fonemów polskich. Ponieważ badanie postaci alofonów odbywać się może jedynie w układach z niepustym kontekstem głoskowym (w tym - w całych wyrazach), trzeba było utworzyć zbiór reguł dotyczących sterowania różnymi opcjami działania syntezatora w sytuacjach, gdy kolejne elementy w ramach jednej wypowiedzi (głoski lub różne ich elementy składowe - w przypadku glosek o złożonej strukturze) są różnego typu. Poza tym ustalone być powinny zasady aktualizacji parametrów przy przejściu do kolejnych ramek czasowych (fram). Reguły stosowane we wcześniejszych wersjach omawianego syntezatora okazały się niewystarczające, a w niektórych istotnych punktach - nie były w ogóle sformułowane.

Utworzenie zbioru reguł nie wyznacza jednakże sposobu realizacji tych reguł w konkretnej implementacji syntezatora. Jednym z możliwych rozwiązań jest dodanie nowych parametrów - w ten sposób użytkownik programu sam mógłby decydować o dalszych szczegółach procesu syntezy. Jednakże w poprzednich wersjach programu syntezy łączna liczba opcji kontrolowanych przez użytkownika była już i tak bardzo znaczna (łącznie 38 różnych parametrów). Uznano zatem dodawanie nowych opcji do listy parametrów (w dodatku dotyczących sterowania przy przechodzeniu od framy do framy - a więc o charakterze innym niż dotychczasowe parametry) za niedpowiednie rozwiązanie. Zamiast tego zawarto reguły sterowania wprost w modelu syntezy - i/lub w odpowiadającym mu czynnościowo module programu. Jest to właśnie *automatyzacja sterowania parametrami akustycznymi* w cyfrowym syntezatorze mowy.

W niniejszym opracowaniu termin *automatyzacja* (sterowania parametrami akustycznymi) (A) oznaczać zatem będzie pewien

podzbiór modyfikacji **(M)** oraz **(S)**. W sytuacjach, gdy zajdzie potrzeba odróżnienia automatyzacji na poziomie modelu syntezy i automatyzacji na poziomie programu syntezy, stosowane będą odpowiednio symbole **(AM)** oraz **(AS)**. Przy tym rozwiązanie danego problemu na poziomie **(AM)** wymaga modyfikacji **(AS)**, lecz nie na odwrót - modyfikacja **(AS)** może wystąpić samodzielnie.

5. Dyskusja podziału parametrów na stałe i framowe - wprowadzone zmiany.

Podział parametrów w edytorze programu SMOK na stałe i framowe, przedstawiony w części 3, nie jest w pełni konsekwentny: **FNP**, **BNP**, **BNZ**, **A1**, **AN** powinny być z matematycznego punktu widzenia traktowane w modelu syntezy tak samo jak parametry framowe - w każdym razie obliczenia w procedurze AMPL w odniesieniu do tych parametrów wykonywane są dla każdej kolejnej framy wypowiedzi. Stan podziału wywodzi się z pierwotnego przeznaczenia programu syntezy do generowania pojedynczych głosek. W przypadku tym, zwłaszcza podczas syntezy samogłosek, wymienione parametry istotnie mogą pozostawać stałe. Synteza sekwencji o bardziej złożonej strukturze akustycznej, przewidziana w modelu teoretycznym, wymagać może jednak modyfikowania wartości tych parametrów w toku wypowiedzi.

Jako przykład, rozpatrzmy sytuację, gdy w toku syntezy równoległej pragniemy zmienić względną amplitudę pierwszego formantu w porównaniu do formantów pozostałych. Można to osiągnąć jedynie poprzez zmianę parametrów **A2**..**A6**, co w dodatku niesie niebezpieczeństwo "niedopasowania amplitudowego" danej części wypowiedzi do jej reszty. Analogiczne problemy wystąpić mogą także przy modyfikowaniu efektu "nosowości" w ciągu głosek o zróżnicowanej strukturze.

W rezultacie, wymienione wyżej parametry zostały przeniesione na listę parametrów framowych. Operacja ta objęła jednak i kilka innych parametrów stałych: **NWS**, **NFC**, **G0** oraz **FL**. Powód tego przeniesienia jest dość oczywisty w przypadku **NWS**: plozje głosek zwartych i zwartotrących wymagają modyfikacji

parametrów w krótszych odstępach czasu niż ma to miejsce na wielu innych odcinkach wypowiedzi. Z kolei uzmiennienie **G0** oraz **FL** ułatwia między innymi sterowanie wygłosową częścią wypowiedzi, w której, jeśli jest dźwięczna, występują zwiększone wahania długości poszczególnych okresów, a opadającą amplitudę łatwo ukształtować właśnie za pomocą **G0**. Możliwość zmiany liczby formantów w syntezie kaskadowej (**NFC**) w toku jednej wypowiedzi nie jest obecnie wykorzystywana, może okazać się jednak przydatna w przyszłości, gdy np. potrzebne będzie przejście z głosu męskiego na kobiecy - i odwrotnie.

Ponadto z przyczyn, które zostaną omówione w jednej z dalszych części pracy, zlikwidowano parametr **SW**, zastępując go automatycznym wyborem syntezy: równoległej w obrębie danej framy dla **AF**>0, szeregowej dla **AF**=0 (poziom modyfikacji (**AM**)).

Należy zwrócić także uwagę na fakt, że parametr **DU** nie powinien faktycznie pełnić funkcji sterującej, a jedynie informacyjną, prezentując po prostu sumę iloczasów wszystkich fram. Natomiast dla precyzyjnego modyfikowania długości wypowiedzi w nowym układzie parametrów stałych i framowych wystarczają: uzmiennione **NWS** oraz funkcje edytora - wstawianie i usuwanie fram.

W efekcie, w zmodyfikowanym programie (z wyjątkiem likwidacji **SW** - poziom modyfikacji (**S**)), **lista parametrów stałych** obejmuje jedynie **trzy** wielkości, z których dwie poddają się bezpośredniej edycji:

1. **DU** (parametr informacyjny)
2. **SD** (możliwa edycja)
3. **SR** (możliwa edycja)

Lista parametrów framowych po modyfikacji zawiera zatem następujące **34** pozycje:

1. **NWS**
2. **G0 [dB]**
3. **F0 [Hz]**
4. **AV [dB]**

5. **NFC**
6. **OQ [%]**
7. **TL [dB]**
8. **FL [%]**
9. **AF [dB]**
10. **AH [dB]**
11. **FNP [Hz]**
12. **BNP [Hz]**
13. **FNZ [Hz]**
14. **BNZ [Hz]**
15. **AN [dB]**
16-21. **F1..F6 [Hz]**
22-27. **B1..B6 [Hz]**
28-33. **A1..A6 [dB]**
34. **AB [dB]**

(Znaczenie poszczególnych parametrów jest oczywiście zgodne z opisem umieszczonym w części 3.)

Dla zapewnienia możliwości użytkowania powstałej już sporej bazy danych eksperymentalnych - zbiorów parametrów o układzie zgodnym z algorytmem programu SMOK - przygotowany został niewielki dodatkowy program konwersji tych zbiorów na nową, przedstawioną wyżej postać.

6. Problem szybkości działania syntezy i modyfikacje z nim związane.

Moduł syntezy programu SMOK napisany został niemal w całości w języku Turbo Pascal. Jak w każdym języku programowania o przeznaczeniu ogólnym, kod utworzony przez kompilator musi spełniać wymogi bezpieczeństwa (lista rozkazów procesora 16-bitowego, wywoływanie procedur, indeksowanie tablic, wykonywanie pętli, praca z danymi w formatach zmiennoprzecinkowych). Na ogół (właśnie ze względu na bezpieczeństwo wykonywanych operacji) jest to jednak kod nieoptymalny. Przygotowanie procedur, których

szybkość działania ma kluczowy wpływ na sprawność modułu syntezy, w wersji asemblerowej, było zabiegiem ściśle technicznym (poziom modyfikacji (S)), mającym jednak istotne znaczenie dla całego programu syntezy. Jak już wspomniano we wstępie, asembleryzację wykonano dla 32-bitowego komputera typu IBM PC, wyposażonego w koprocesor (warunki te spełnia każda konfiguracja zgodna "w dół" z PC 386DX z zainstalowanym koprocesorem arytmetycznym). Program SMOKIN1 bada konfigurację sprzętową i daje się uruchomić tylko na komputerze spełniającym wymienione wymagania sprzętowe (dodatkowym wymogiem, nie związanym z działaniem modułu syntezy, jest karta graficzna VGA lub SVGA).

"Przy okazji" asembleryzacji w całym module syntezy przyjęto metodę odwołań wskaźnikowych podczas zapisywania wyliczonych próbek do pamięci - zamiast wywoływania znacznie wolniej działającej procedury paskalowej New(.), oraz zrezygnawano z normalizacji sygnału. Po utworzeniu ciągu próbek dla danej wypowiedzi w programie SMOK następuje bowiem korekta amplitudy, tak by osiągane było dopuszczalne maksimum i minimum sygnału. Ponieważ parametry **G0**, **AV**, **AF**, **AH**, **A1..A6** pozwalają i tak dowolnie kształtować amplitudę wypowiedzi z framy na framę, ta dodatkowa normalizacja nie wydaje się być potrzebna. Co więcej - utrudniałaby ona kontrolę wzajemnych relacji amplitudowych dla zbioru wzorców głosek. W programie SMOKIN1 zachowano jedynie kontrolę przesterowania sygnału, wykonywaną na bieżąco przed zapisaniem kolejnej próbki. Skutkiem powyższego (także poziom modyfikacji (S)) jest oczywiście redukcja ilości operacji maszynowych przypadających na 1 próbkę.

. Dalsze znaczne przyspieszenie działania modułu syntezy zrealizowano modyfikując działanie generatora szumu. W części 2 na str.5 wspomniano, że zadaniem takiego generatora jest dostarczenie sygnału, który w dziedzinie czasowej ma w przybliżeniu rozkład normalny (a - co za tym idzie - w dziedzinie częstotliwości rozkład równomierny). Programowa realizacja tego procesu w SMOKu wykorzystuje w tym celu funkcję (pseudo)losową modułu bibliotecznego kompilatora Turbo Pascala. Jednakże ciąg

pojedynczych wywołań tej funkcji daje rozkład zbliżony do równomiernego. Aby uzyskać ciąg o rozkładzie normalnym, potrzebny w modelu syntezy, dla każdej próbki generowanego sygnału mowy dokonuje się (w programie SMOK) sumowania wartości 16-krotnie wywołanej funkcji losowej i standaryzacji otrzymanej sumy. Jest to bezpośrednia implementacja jednego z podstawowych twierdzeń rachunku prawdopodobieństwa; zgodnie z tym twierdzeniem, granicą standaryzowanej sumy **n** rozkładów równomiernych przy **n** dążącym do nieskończoności jest rozkład normalny. W zastosowaniach praktycznych wystarczające przybliżenie uzyskuje się dla **n≥16**, stąd 16-krotne wywoływanie funkcji losowej w pętli obliczeń dla każdej próbki. W rezultacie powstaje bardzo duże obciążenie modułu syntezy obliczeniami, które z punktu widzenia modelu syntezatora stanowią jedynie niewielki fragment jego działania. Na problem powyższy w programie SMOKIN1 spojrzano nieco inaczej. Na starcie całej sesji z programem przygotowywana jest jednorazowo tablica zawierająca 8000 liczb losowych, wygenerowanych zgodnie z twierdzeniem o sumie rozkładów równomiernych. Natomiast podczas syntezy generator szumu wywołuje w każdym kroku pętli programowej **jednokrotnie** paskalową funkcję losową z parametrem ograniczającym zakres wartości do liczb całkowitych od 0 do 7999, a uzyskana dana jest wskaźnikiem do elementu w przygotowanej uprzednio tablicy rozkładu "prawie normalnego". Metoda ta daje efekt o tyle mniej dokładnie naśladujący rozkład normalny (w porównaniu do rozwiązania przyjętego w programie SMOK), o ile rozkład wartości funkcji losowej w Pascalu różni się od równomiernego. Praktyczna weryfikacja uzyskanych wyników poprzez porównanie wypowiedzi syntetycznych otrzymanych na podstawie analogicznych zbiorów parametrów w programach SMOK i SMOKIN1 nie wykazuje przy tym żadnych istotnych różnic w jakości obydwu wypowiedzi. W ten sposób, kosztem zwiększenia obszaru danych programu (tablica losowa zajmuje 4*8000=32000 bajtów pamięci), zredukowano liczbę wywołań funkcji losowej w samym procesie syntezy 16-krotnie.

Opisaną powyżej modyfikację zaliczyć trzeba do poziomu **(S)**, gdyż model syntezatora (p.część 2) nie narzuca wyboru konkretnej

realizacji generatora szumu, a jedynie określa ogólne cechy wygenerowanego ciągu danych. Natomiast następna modyfikacja jest ingerencją w strukturę modelu przedstawioną na rys.1 (cz.2, str.7). W modelu tym funkcjonuje przełącznik **SW**, pozwalający obliczać składową dźwięczną alternatywnie w układzie syntezy szeregowej (**SW**=0), bądź równoległej (**SW**=1). Składowa pochodząca od szumu frykacyjnego (**AF**>0) realizowana jest i tak w trybie równoległym. Oznacza to, że np. głoski dźwięczne trące przy **SW**=0 będą obliczane w ten sposób, iż dla kolejnej próbki wykonane zostaną po 2 serie obliczeń (dla składowej "szeregowej" i "równoległej" oddzielnie). Jednakże dla głosek takich blok syntezy równoległej może wykonać w jednej serii wszystkie potrzebne obliczenia. Aby zmniejszyć liczbę rozkazów maszynowych potrzebnych dla wytworzenia tego typu odcinków wypowiedzi przyjęto, iż synteza będzie wykonywana w całości w trybie równoległym przy **AF**>0, a na pozostałych odcinkach - w całości w trybie szeregowym. Sam przełącznik **SW** został wyeliminowany (p.także cz.5, str.16) - jego rolę przejęło badanie wartości **AF** w bieżącej framie. Ceną za takie rozwiązanie jest konieczność "sztucznego" sterowania trybem syntezy, gdy z jakiegoś powodu (może być nim np. zamiar ograniczenia liczby formantów do dwóch lub trzech - synteza szeregowa generuje sygnał z co najmniej czterema formantami) pragniemy wygenerować głoskę nie-trącą dźwięczną w trybie równoległym. W sytuacji takiej trzeba mimo wszystko ustawić "techniczną" wartość **AF**=1[dB]: wyzwala to tryb równoległy syntezy, a nie wywiera żadnego wpływu na postać czasową sygnału. Dopiero od około 5[dB], przy przeciętnym poziomie **G0** oraz **A1**..**A6**, wpływ ten daje się zauważyć, a typowym przedziałem wartości amplitudy frykacji **AF** dla głosek trących jest 20-65[dB]. Oczywiście, należy zdawać sobie sprawę z "teoretycznej nieczystości" przyjętego w programie SMOKIN1 rozwiązania, lecz za jego zastosowaniem przemawiało istotne skrócenie obliczeń.

Opisana powyżej modyfikacja z oczywistych powodów należy do poziomu **(A)** - automatyzacja (sterowania parametrami akustycznymi).

Po wszystkich modyfikacjach (w tym również opisanych w następnej części niniejszej pracy), porównanie czasu realizacji syntezy kontrolnej wypowiedzi "i y e a o u" o długości 2.3 sek za pomocą programów SMOK i SMOKIN1 dało następujące wyniki:

a) Dla komputera 386DX / 33 MHz wyposażonego w koprocesor arytmetyczny: SMOK - ok. 13.90 sek, SMOKIN1 - ok. 2.75 sek (czas syntezy przekracza długość wygenerowanej wypowiedzi o około 20%).

b) Dla komputera 486DX / 50 MHz (w układzie tym jednostka arytmetyki zmiennoprzecinkowej jest zintegrowana z innymi układami procesora): SMOK - ok. 3.35 sek, SMOKIN1 - około 0.75 sek (czas syntezy trzykrotnie krótszy od utworzonej wypowiedzi).

7. Automatyczna kontrola przełączania parametrów na granicy fram.

Prace wykonane w odniesieniu do tego punktu mają decydujący wpływ na zdolność syntezatora do generowania poprawnych połączeń między głoskami (lub ich elementami) rozmaitego typu. Metody automatycznego sterowania parametrami akustycznymi syntezy w podobnych sytuacjach nie zostały zaimplementowane w SMOKu. W programie SMOK cykl syntezy dla kolejnej framy zaczyna się wywołaniem procedury AMPL, w związku z czym wartości amplitudowe oraz współczynniki rezonatorów aktualizowane są od początku framy, bez względu na jej strukturę i lokalizację w generowanej wypowiedzi (por. cz.3). W związku z powyższym przyjęto w tej części opracowania konwencję odmienną niż w cz.5 i cz.6. Wszystkie przedstawione dalej mechanizmy sterowania oraz struktura danych dotyczyć będą wyłącznie zmodyfikowanej wersji syntezatora i programu SMOKIN1. Przy tym, wprowadzone modyfikacje należą do poziomu automatyzacji sterowania **(AS)**, jako że model przedstawiony na rys.1 (str.7) jest modelem ogólnym i statycznym, nie precyzuje zatem technik kontroli parametrów ani konkretnych momentów aktualizowania współczynników amplitudowych oraz postaci rezonatorów. Dynamiczne aspekty sterowania pozostają związane z

projektem programu i jego konkretną realizacją.

Przede wszystkim przypomnieć trzeba, że zmodyfikowany zbiór danych, na podstawie którego wykonywana jest synteza, ma postać ciągu ramek czasowych (fram), z których każda opisuje zbiór 34 parametrów; dodatkowo 3 parametry stałe opisują globalne cechy tworzonej wypowiedzi. Ze względu na taką strukturę zbioru parametrów, zamiast sekwencji głosek należy więc brać pod uwagę sekwencje fram o rozmaitej wartości konkretnych parametrów. Cechami istotnymi są przy tym:

1) dźwięczność / bezdźwięczność

Cechę tę definiują wspólnie częstotliwość podstawowa **F0** oraz amplituda pobudzenia dźwięcznego **AV**. Frama jest dźwięczna tylko wtedy, gdy **F0**>0 oraz **AV**>0. W pozostałych przypadkach frama jest bezdźwięczna.

2) tryb sytezy (szeregowa / równoległa)

Frama kwalifikuje się do syntezy szeregowej, jeśli **AF**=0. W przeciwnym razie frama kwalifikuje się do syntezy równoległej.

3) liczba formantów w kaskadzie

Liczba ta, formalnie rzecz biorąc, może zmieniać się w zakresie 4-6.

Zakłócenia w postaci czasowej generowanego sygnału mogą w związku z tym powstawać w następujących sytuacjach.

a) Poprzednia frama była dźwięczna. W ramach obliczeń dla tej framy rozpoczął się cykl kolejnego okresu **F0**, ale w trakcie jego realizacji program dotarł do granicy framy. W związku z tym w przypadkowym punkcie okresu **F0** rozpoczyna się liczenie próbek według nowych wartości amplitudowych. Jeśli wartości te różnią się w niewielkim stopniu od poprzednich (tak jest np. podczas syntezy izolowanych samogłosek), to problemu nie ma. Dużo gorzej ma się sprawa przy skokowym wzroście amplitud (np. dla plozji dźwięcznej) lub zmianie trybu syntezy. Łatwo wtedy o wyprodukowanie przypadkowego trzasku lub nawet przesterowanie sygnału.

Już na podstawie tej jednej przesłanki zdecydowano, że w "pętli framowej" procedury AMPL i CREZ nie będą wywoływane kolejno po sobie. Odnowienie wszelkich parametrów amplitudowych i współczynników rezonatorów będzie uzależnione od zakończenia aktualnego okresu **F0** rozpoczętego w poprzedniej framie, jeśli była ona dźwięczna. Jedynie w przypadku poprzedniej framy bezdźwięcznej aktualizacja następuje od początku framy następnej. Tak więc, procedura AMPL jest wywoływana z wnętrza procedury CREZ w pierwszym momencie, który dopuszcza bezpieczne jej wywołanie.

Skutkiem ubocznym takiej organizacji syntezy jest możliwość pominięcia parametrów bardzo krótkiej framy, jeśli znalazła się ona we wnętrzu aktualnie realizowanego okresu **F0**.

Przykład: Sygnał generowany jest przy częstotliwości próbkowania **SR**=10000[Hz]. W pierwszej framie wypowiedzi mamy **F0**=125[Hz], co oznacza, że w ramach jednego okresu wyliczanych będzie 10000/125=80 próbek. Jeśli w pierwszej framie **NWS**=100 próbek, to w momencie dotarcia do granicy z drugą framą wyliczonych zostało tylko 20 próbek dla drugiego okresu; pozostałych 60 (do końca drugiego okresu) będzie liczonych nadal na podstawie danych z pierwszej framy, mimo że formalnie należą one do obszaru framy drugiej. Jeśli w tej drugiej framie jest przy tym **NWS**≤60, to parametry drugiej framy nie zostaną w ogóle uwzględnione, a następna ich aktualizacja dotyczyć będzie framy trzeciej. **NWS**>60 w drugiej framie oznacza natomiast, że aktualizacja współczynników według parametrów tej framy nastąpi dopiero przed wyliczeniem wewnątrz niej sześćdziesiątej pierwszej próbki.

b) W ramach jednej syntetycznej wypowiedzi następuje dwukrotne przełączenie trybu syntezy (szeregowa -> równoległa -> szeregowa, lub równoległa -> szeregowa -> równoległa). Zgodnie ze wzorami (3) na str.8 oraz (3') na str.9, określić trzeba przy tym startowe dla danej syntezy wartości wyjściowe **y((n-1)·T)** oraz **y((n-2)·T)** dla każdego rezonatora (wartości wejściowe **x((n-1)·T)** oraz **x((n-1)·T)** dla antyrezonatora). Tymczasem nie ma ich skąd wziąć (bo właśnie rozpoczyna się synteza według zmienionego

trybu). W takiej samej sytuacji znajduje się układ syntezy na początku wypowiedzi (gdy liczona jest pierwsza próbka) - wtedy ustawia się poprzednie wartości wyjściowe i wejściowe na "0". Taką samą metodę - zerowania "historii" rezonatora - realizuje się przy przełączaniu trybów syntezy, ale z ważnym wyjątkiem. Okazuje się bowiem, że wskutek właściwości rachunkowych modelu ciąg próbek rozpoczynający syntezę równoległą uzyskuje dość wolno zadaną dynamikę zmian. Jest to zjawisko normalne w nagłosie wypowiedzi, lecz w jej wnętrzu powstaje w analogicznej sytuacji efekt skandowania głosek, gdy przełączenie dokonuje się w celu syntezy spółgłoski dźwięcznej trącej po samogłosce. Natomiast w przypadku sekwencji samogłoska - spółgłoska trąca bezdźwięczna objaw powyższy nie występuje. Jest tak zapewne dlatego, iż w wypowiedziach naturalnych także ma miejsce wyraźne obniżenie amplitudy między samogłoską i następującą po niej spółgłoską trącą bezdźwięczną, więc przebieg syntezy w tym przypadku "mimochodem" naśladuje mechanizmy naturalne. Rozwiązaniem problemu sekwencji samogłoska-głoska trąca dźwięczna jest wprowadzenie podwójnego przeliczenia próbek pierwszego okresu na granicy krytycznych fram. Wyniki obliczeń pierwszego przebiegu nie są nigdzie zapisywane, natomiast uzyskuje się na ich podstawie "wiarygodne" wartości wyjściowe określające historię każdego z rezonatorów. W rezultacie sygnał z toru równoległego uzyskuje od razu założoną dynamikę i "efekt skandowania" znika.

Oczywiście, wszystkie powyższe operacje wykonywane są w momencie syntezy wybranym zgodnie z puntem a).

c) Dodatkową operacją, przewidzianą w systemie automatycznego sterowania, jest zerowanie historii rezonatora 6, albo rezonatorów 5 i 6 podczas zwiększenia z framy na framę liczby formantów (**NFC**) w syntezie kaskadowej. Operacja ta zapobiega użyciu przypadkowych wartości wyjściowych **y((n-1)·T)** oraz **y((n-2)·T)** w równaniach tych rezonatorów, zatem służy celom analogicznym jak opisane w punkcie b) - przy przełączaniu trybów syntezy.

Automatyzacja sterowania objęła jeszcze jedną ważną zmianę, sięgającą do samego modelu syntezy (poziomy **(AM)** i **(AS)**). Zmiana

ta skorelowana jest z wprowadzeniem alternatywy rozłącznej dla wyboru syntezy szeregowej / równoległej (p. część 6, str. 20-21) oraz ze specyfiką struktury akustycznej głosek języka polskiego. Otóż wśród par spółgłosek posiadających frykację - np. "s" i "z", "f" i "v", element bezdźwięczny w przypadku niektórych par głosek powinien mieć wyraźny pierwszy formant, generowany oczywiście w oparciu o szumowe źródło pobudzenia (jest tak dla "f"). Natomiast widmo głosek dźwięcznych trących ujawnia okresowość w niskim pasmie częstotliwości (można to symulować z pomocą pierwszego rezonatora toru równoległego), podczas gdy w przedziale wyższych częstotliwości dominuje składowa szumowa (typowe jest "z"). Poprawne generowanie głosek trących wymagało utworzenia automatycznego przełącznika, który podaje pobudzenie dla pierwszego rezonatora w torze równoległym alternatywnie: z generatora tonu krtaniowego dla głosek dźwięcznych, z generatora szumu - dla bezdźwięcznych.

W rezultacie wprowadzonych mechanizmów, program SMOKIN1 pozwala bezpiecznie przełączać zarówno tryby syntezy, jak i przeplatać dźwięczne oraz bezdźwięczne odcinki wypowiedzi - jest to spełnienie minimum wymagań dla skutecznej generacji wypowiedzi o złożonej strukturze.

8. Dodatkowe usprawnienia w programie syntezy.

Modyfikacje usprawniające pracę programu przeprowadzono tak, by pozostawić bez zmiany listę funkcji programu zawartą w jego menu - jest ono bowiem w programie SMOK dobrze zaprojektowane. Modyfikacje te, należące oczywiście do poziomu (I), obejmują:

a) Prezentację parametrów stałych i framowych zgodną ze zmianami struktury tych parametrów podanymi w cz.5.

b) Generalną zmianę metod zarządzania pamięcią użytkowaną przez program - przejście od stosowania funkcji bibliotecznych Turbo Pascala do ściśle kontrolowanych operacji wskaźnikowych - przyspiesza to wszelkie operacje usługowe programu.

c) Zmianę postaci i zasad funkcjonowania graficznej edycji parametrów framowych - z zastosowaniem typowej w takich sytuacjach możliwości wycofania się z tej edycji w dowolnym jej momencie i przywrócenia pierwotnych wartości danego parametru. Edycja w trybie graficznym może dotyczyć dowolnego parametru framowego oprócz **NWS**.

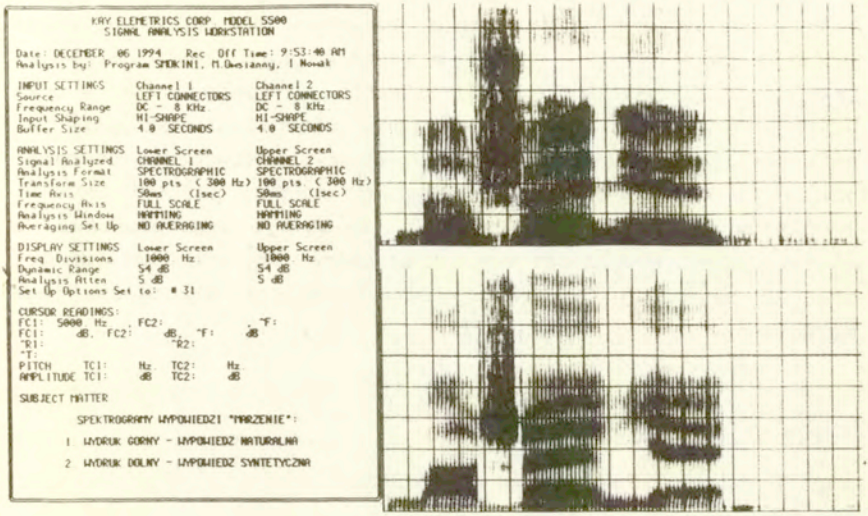
d) Umieszczenie w programie SMOKIN1 szeregu zabezpieczeń, zapewniających poprawne jego funkcjonowanie nawet w przypadku nieprawidłowego działania użytkownika lub braku miejsca w pamięci na zapisanie kolejnych danych.

Charakter niniejszego opracowania nie pozwala na bardziej szczegółowe omówienie wewnętrznych struktur i zabezpieczeń programu SMOKIN1.

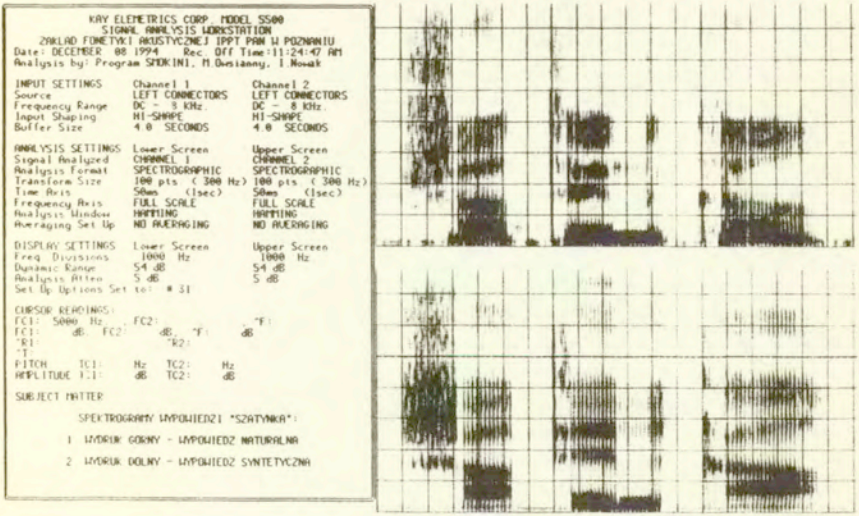
9. Podsumowanie osiągniętych rezultatów.

W wyniku wprowadzenia do software'owego syntezatora SMOK opisanych powyżej szczegółowo zmian powstał program, stanowiący precyzyjne i elastyczne narzędzie, umożliwiające generowanie nie tylko izolowanych głosek, ale także ich dłuższych sekwencji. Ostateczna wersja tego programu (SMOKIN1) została stworzona na drodze wieloetapowych działań, w których istotną rolę odegrały przeprowadzane na bieżąco testy z wykorzystaniem syntezy mowy. Pozwalały one wykryć i wyeliminować różnego typu mankamenty oprogramowania i wytyczały kierunki dalszych modyfikacji. Przyjęcie takiego trybu postępowania zwiększa niejako gwarancję niezawodności programu SMOKIN1.

Eksperymenty z zakresu syntezy mowy przeprowadzone przy użyciu ostatecznej wersji programu wykazują, że umożliwia on dość wierne naśladowanie zjawisk fonetyczno-akustycznych występujących w mowie naturalnej. Świadczy o tym na przykład znaczne podobieństwo zamieszczonych poniżej wydruków spektrogramów wypowiedzi naturalnych i syntetycznych.



Rys. 2. Spektrogramy naturalnej (wydruk górny) oraz syntetycznej (wydruk dolny) wypowiedzi "marzenie".



Rys. 3. Spektrogramy naturalnej (wydruk górny) oraz syntetycznej (wydruk dolny) wypowiedzi "szatynka".

To, co napisano powyżej nie oznacza, że program SMOKIN1 nie będzie podlegał dalszym zmianom. Prawdopodobne są na przykład pewne modyfikacje w zakresie kształtowania charakterystyki pobudzenia dźwięcznego w części równoległej syntezatora. W toku użytkowania programu mogą się ponadto ujawnić niedociągnięcia, na które nie natrafiono w przeprowadzonych dotychczas testach. Jak się jednak wydaje, już w swojej obecnej postaci SMOKIN1 umożliwia syntezę wypowiedzi o wysokiej zrozumiałości i naturalności. Zasadniczy warunek uzyskania tej wysokiej jakości stanowi odpowiednio rozległa wiedza fonetyczno-akustyczna jego użytkownika.

BIBLIOGRAFIA.

[1] ALLEN J., HUNNICUTT M.S., KLATT D.H., *From text to speech: The MITalk System*, Cambridge University Press 1987.

[2] IMIOŁCZYK J., NOWAK I., DEMENKO G., *A Text-to-Speech System for Polish*, Proceedings of the 3rd European Conference on Speech Communication and Technology EUROSPEECH'93, Berlin, 1993, vol. 2, pp. 889-892.

[3] IMIOŁCZYK J., NOWAK I., DEMENKO G., *Implementacja systemu syntezy ciągłej mowy polskiej z tekstu ortograficznego wprowadzonego z klawiatury komputera typu PC*, Prace IPPT, **11**, 1993.

[4] IMIOŁCZYK J., NOWAK I., DEMENKO G., *High-Intelligibility Text-to-Speech Synthesis for Polish*, 1994, Arch. of Acoustics, vol. 19, No 2, pp. 161-172.

[5] KLATT D.H., *Software for a cascade/parallel formant synthesizer*, J. Acoust. Soc. Am.,**67**, 971-995, 1980.

[6] KLATT D.H., KLATT L.C., *Analysis, synthesis, and perception of voice quality variations among female and male talkers*, J. Acoust. Soc. Am., **87**, 2, 820-857, 1990.

[7] OWSIANNY M., *Software'owa realizacja formantowego syntezatora mowy*, w: "Księdze pamiątkowej ofiarowanej Pani Profesor Marii Steffen-Batogowej oraz Panu Profesorowi Tadeuszowi Batogowi, red. J. Pogonowski, Wydawnictwo Naukowe UAM, Poznań (w druku).

[8] OWSIANNY M., *The synthesis of female voices using a software synthesizer*, Archives of Acoustics, **19**, 2, 185-199, 1994.

[9] SZUTOWSKI B., *Assemblerowa implementacja modelu software'owej syntezy formantowej mowy*, Praca magisterska obroniona na UAM w Poznaniu (promotor J.Pogonowski), 1994.