Софийски Университет "Св. Климент Охридски"
Факултет по математика и информатика
Катедра "Математическа логика и приложенията ѝ"

Дипломна работа

Синтез на реч чрез статистически модел

Иван Заманов

Научни ръководители
гл. ас. д-р Петър Митанкин
доц. др. Стоян Михов

1 октомври 2016 г.

Съдържание

1 Въведение

Речта е най-естественият начин за комуникация на човека. Използването на реч от компютърни системи за взаимодействие с потребителите е ключово за постигането на максимално естествен начин на използването им. Приложенията на система за генериране на реч са многобройни и в най-различни области.

В много ситуации речевият интерфейс е незаменим, например за автомобилни навигационни системи, където човек не може да използва зрението си, за да следи подаваната му информация. За хора с увредено зрение пък това е единственият начин да взаимодействат със съвременните компютърни системи. Синтезирането на реч се използва широко от телекомуникационни компании в гласови менюта, тъй като речта е и единственият начин за предаване на информация по телефонни линии. С все по-широкото използване на аудио/видео разговори по интернет през все по-мощни компютърни устройства се появява и възможността за превод на речта в реално време при разговори между хора, говорещи различни езици.

В образованието синтезирането на реч би могло да се използва при отсъствието на човек, говорещ даден език свободно. Съществуват и многобройни изследвания, показващи, че човек запомня по-лесно, когато биват стимулирани няколко сетива наведнъж. Синтезатор на реч би могъл да спомогне за по-ефективно учене или пък за създаването на по-интерактивни книги или защо не и играчки.

Също така, достатъчно добър синтезатор, в комбинация с достатъчно добра система за автоматично разпознаване на реч (automatic speech recognition, ASR) би бил и изключително ефективен метод за компресия и декомпресия на речеви сигнал. Никой съвременен алгоритъм не би могъл да се сравнява с представянето на даден речеви сигнал като обикновен текст, съпроводен с информация за желания начин на произнасяне (прозодия - акцент и интонация) на текста.

2 Предварителни означения

Ще отделим тази секция на някои дефиниции и означения, към които ще се придържаме в изложението, освен ако не е указано друго.

Първо, някои дефиниции на понятия от езикознанието, както са дадени в [1], глава 6.1:

- **Фонемата** (от гр. phone "глас, звук") е най-малката незначеща сегментна единица на езика, която е линейно неделима по-нататък. Тя представлява абстрактен звуков тип, обобщена слухова представа за поредица сходни физически звукове, които се чуват в речта на носителите на даден език. (Например "а", "к", "л" и т.н.)

- Фонемите са абстрактни единици, инварианти, които притежават свои разновидности, наричани **алофони** или варианти на фонемата. (Твърдо или меко "л", например, са алофони на фонемата "л")
- В речта, при актуалното си изговаряне, всеки алофон на фонемата се проявява като реален физически звук, произнесен от някого в конкретно време, място и звуково обкръжение. Този звук се определя като **фон** или **речев звук**.
- **Прозодия** (от гр. prosodia “припев, ударение”) е съвкупността от акцент и интонация на потокът на устната реч.
- Под **фонетична единица** ще разбираме фон, полуфон, дифон или трифон (два или три последователни фона в потока на естествена реч). Ще отбележим, че това включва и информация за прозодията на фонът,т.е. продължителност и височина.
- **Полуфон** ще наричаме единица, по-малка от фонът, която може да бъде **лява** или **дясна**. Ще смятаме, че един фон може да бъде разбит на не повече от два полуфона като съответните ляв и десен полуфон имат сбор от дължини равен на дължината на оригиналният фон. Един начин за извличане на полуфони е описан в Секция 5.2.

Ще са ни необходими и следните означения:

- С главни латински букви (например X, Y) ще означаваме множества от елементи
- С малки латински букви x - елементи на множеството X
- $\vec{x} = (x_1, ..., x_n), n = |\vec{x}|$
- $\vec{X}$ - множества от вектори с елементи от X, т.ч. $\vec{x_1} = (x_1^1, ..., x_{n_1}^1), \vec{x_2} = (x_1^2, ..., x_{n_2}^2) \in \vec{X} \implies n_1 = n_2$
- X^* - множествата от всички вектори с елементи от X
- f, g - функции със стойности от $\mathbb{R}$
- $\lambda, \mu \in \mathbb{R}$
- P - вероятност
- $\exp(x) = e^x$
- С L ще означаваме множеството на **фонетичните етикети**. Това множество съответства на **алофоните** в нашата фонетична система. Ако за фонетични единици изберем фоните, то това множество ще съвпада с множеството алофони. Ако изберем дифони, то то ще бъде изоморфно на наредените двойки алофони. Конкретно използваните фонетични етикети са описани в Приложение А.

Елементите на X и Y ще наричаме описания на речеви сигнал и те ще съответстват на множества от **фонетични единици**. Често ще използваме "описание", "съответстващ речеви сигнал" и "фонетична единица" като взаимнозаменяеми понятия. За простота можем да смятаме, че съществува биекция между елементите на X и **субективно различимите** от човешкият слух единици на един фиксиран носител на езика (диктор).

3 Архитектура и съществуващи подходи

Под "синтезатор на реч" ще разбираме система, способна да произведе речеви сигнал при вход произволен текст от естествен език. В общия случай подобна система се състои от следните компоненти:

- **Анализ на текста**. В естествените езици няма еднозначно съответствие между използваните графеми (писмен знак) и фонеми. Целта на този модул е да преведе графемите на текста до фонетичното им съответствие, т.е. до списък на **алофони**. Това включва и предварителна нормализация на текста - превеждане на числа, абревиатури, съкращения и т.н. до цели думи и словосъчетания. Анализът също така трябва да вземе предвид и езикови феномени като добавяне, изпускане, замяна или разместване на звукове, озвучаване или обезвучаване. Тук обикновено широко се използват фонетични речници, ръчно създадени правила или статистически подходи.

- **Моделиране на прозодията**. Прозодията на речта е начинът на произнасяне на отделни фонеми. По-точно, различават се няколко главни прозодични аспекта, разделени на слухови (възприятийни) и акустични. Слуховите са субективните характеристики - височина на гласа, продължителност на звуковете, сила, тембър и други. Грубо, съответните им обективни акустични характеристики могат да се определят като фундаментална честота (в херци), продължителност (милисекунди), интензивност/енергия (децибели) и редица спектрални характеристики.

  Разполагайки с редица от **алофони**, този компонент има за цел да генерира описание на желаните акустични характеристики на **синтетичните фони**.

- **Генериране на звуков сигнал**. Последната стъпка е генерирането на крайния звуков сигнал на базата на описанията от предния модул.

Тази дипломна работа предлага вариант на третия компонент, отговорен за генерирането на звуков сигнал по съществуващо описание. Съществуват няколко общи подхода към този проблем.

Първият е директното синтезиране на крайния сигнал, използвайки правила за получаване формантите [1] на всеки фон. Обикновено се избират акустичен модел на речта (например source-filter моделът) и правила, по които от описанието да се генерира директно спектър на сигнала. [2] например използва набор от филтри, съответстващи на формантите на всяка една отделна фонема, конволюирани с постоянен сигнал, за да получи директно синтетичния сигнал. Постоянният сигнал съответства на трептенията на гласовите струни - синусоиден сигнал с подходящ период (според желаната височина на синтетичната реч) за звучни фонеми или бял шум за беззвучни такива. [4] използва набор от параметри, съответстващи на състоянието на гласовия апарат в даден момент, например положението на устните, стесняването, породено от разположението на езика, и др., за да определи формантите на фонемите. Стойностите на тези параметри биха могли да бъдат намерени чрез рентгенография и магнитен резонанс. Директното синтезиране чрез правила като цяло не е подходящо за генериране на различни гласове, тъй като е необходимо създаването на отделни правила за всеки глас.

Вторият подход е т.нар. конкатенативен синтез, при който се използват сегменти от съществуващи речеви сигнали (фонетична база), които да бъдат съединени така, че да се получи произволна редица алофони. Това е подходът, избран от [5] и възприет от настоящата работа. Съществуват различни варианти, в зависимост от използваните фонетични единици - например фони, дифони (двойки последователни фони), а дори и трифони.

Друг възможен статистически подход е използването на скрит марковски модел (HMM) [3]. Този подход се характеризира с по-голяма гъвкавост, тъй като прозодията на синтетичната реч не е необходимо да наподобява тази на даден оригинален запис. Скритият марковски модел е в състояние да генерира директно както слухови характеристики на речта (височина, продължителност, сила), така и спектрални (или кепстрални, както в [3]) характеристики на крайния сигнал.

4 Модел

4.1 Постановка

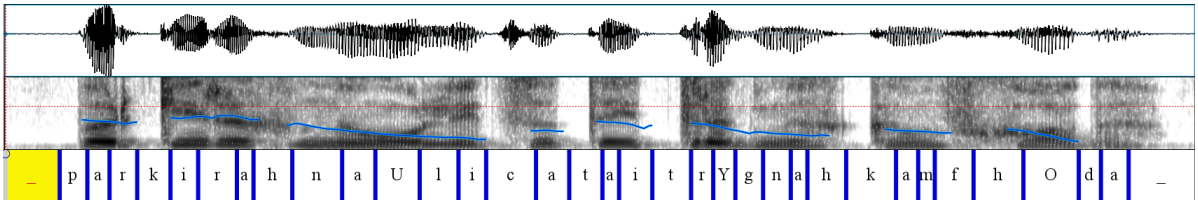
Както вече споменахме, избраният подход в тази дипломна работа е т.нар. конкатенативен синтез - получаването на речеви сигнал като конкатенация на известни фонетични единици. Задачата може да бъде разгледана като съпоставяне на поредица от етикети на някакво описание на желаната произнесена реч. С други думи, входът трябва да бъде описание на желаните синтетични фонетични единици. Етикетите съответстват на из-

[1] Области от спектъра на сигнала, представляващи локален максимум в енергийния спектър на речевия сигнал

вестните ни фонетични единици - тези, за които имаме конкретен речеви сигнал, т.е. това са всички единици от подходящо анотиран речеви корпус.

От съпоставената поредица етикети можем да получим съответния синтетичен речеви сигнал. Ще смятаме, че фонетичните единици са еднородни, т.е. или само полуфони, или само фони и т.н. Ще се стремим сигналът да звучи "естествено", а това е субективно понятие и следователно зле дефинирано. За естествено звучаща можем да приемем, например, речта от самият речеви корпус. Ще се стремим нашият модел да извежда максимално близък до оригинала изходен сигнал за някаква тренировъчна част от корпуса.

Нека например корпусът се състои от изречението "Паркирах на улицата и тръгнах към входа":



Фигура 1: *Анотация на изречението според фонетичната система в Приложение А.*

И нека искаме да получим произнесена например думата "правило". За всеки фон от думата имаме съществуващо произношение като за "р" имаме три, т.е. можем да получим думата като просто извлечем и пренаредим съответните единици. За "р" обаче ако изберем първото произношение от "паркирах" ще се получи неприятно прекъсване на речта понеже в края на произношението се наблюдава затихване в подготовка за произнасянето на "к". Произношението от "тръгнах" също не е подходящо, тъй като след него следва "ъ" - макар и близка до "а", начинът на произношение е различен, т.е. със стеснени устни. Остава второто "р" от "паркирах", което също така се намира и преди "а", т.е. е най-подходящото от трите произношения. Това е опростен пример, който не взима предвид индивидуалните продължителност и височина на фоните. В реални условия за всяка единица от думата "правило" е необходимо да имаме някаква целева продължителност и височина. Тъй като разполагаме с ограничена фонетична база (в примерът - единствено изречение), се налага прибягването до алгоритъм, който да модифицира изходният сигнал така, че той да отговаря на желаните параметри (описан в Секция 8).

Така нашата задача е по зададена поредица фонетични единици да намерим най-подходяща поредица етикети, избрани измежду известните ни фонетични единици, такава, че да получим максимално естествено звучащ речеви сигнал. Разбира се, ще имаме за цел да синтезираме цели изречения, а не думи по отделно.

По-формално, нека имаме множество X, съдържащо всевъзможните фонетични единици от един тип (ще се спрем на полуфонът като такъв) в българския език от един носител, и множество $\hat{Y}$ от известни **изречения** от същия носител, получени от някакъв източник (речеви корпус). $\hat{Y}$ задава и съответно множество от известни фонетични единици Y, което от своя страна задава и множеството $Y^* \supset \hat{Y}$ на всички изречения, които можем да получим чрез директна конкатенация на известни единици. Тъй като $|Y| \ll |X|$ ще считаме, че елементи на Y са и всички варианти на единиците от корпуса, които могат да бъдат получени чрез алгоритъма, описан в Секция 8. Накратко, това са всички подобни единици, които се различават само по височина и/или продължителност от "същинските" елементи на Y.

Можем да приемем също така, че X е крайно, тъй като броят единици, които човек би могъл да различи, е краен, с някаква точност за продължителност, височина или изобщо спектър на сигнала. Елементите на X ще наричаме входни описания, а тези на Y - изходни. Съответстващите на $\vec{y} \in Y^*$ звукови сигнали ще наричаме изходни, тъй като такива можем да получим чрез конкатенация сигналите на $y \in Y$. В общия случай $\vec{y}$ не се съдържа в корпуса, затова и всяко съответствие $x = y \in Y$ е благоприятно, тъй като вече разполагаме с възможно най-естествен речеви сигнал за него.

При тези означения търсим функция $Synth : X^* \to Y^*$ такава, че:

$$Synth(\vec{x}) = \underset{\vec{y}' \in Y^*}{\operatorname{argmin}} Target(\vec{x}, \vec{y}') \tag{4.1}$$

където $Target(\vec{x}, \vec{y}) : X^* \times Y^* \to \mathbb{R}$. $Target$ е функция, която отразява акустичното разстояние на сигнала, получен от $\vec{y}$, до идеалния сигнал на $\vec{x}$. Достатъчно да предположим, че ако $\vec{x}$ се съдържа в нашият корпус, то идеалният сигнал за $\vec{x}$ е съответният известен от корпуса сигнал. В общия случай не разполагаме с идеален сигнал за $\vec{x}$ и затова е нужно да прибегнем до някакъв модел, който да предсказва доколко дадено $\vec{y}$ е "отдалечено" от идеалният за $\vec{x}$ сигнал.

Речта се възприема последователно, затова можем да разгледаме едно изречение като поредица от фонетични единици, в която всеки зависи от неговите съседи. Да разгледаме граф $G(\vec{x}) = (V, E)$, където:

$$\begin{aligned} &V \subset X \times Y, V = \{x_i | i = 0, ..., |\vec{x}| - 1\} \times Y \\ &E \subset V \times V, E = \{((x_i, y'), (x_{i+1}, y'')) | i = 0, ..., |\vec{x}| - 2, y', y'' \in Y\} \end{aligned} \tag{4.2}$$

На всеки път в $G(\vec{x})$ с дължина $|\vec{x}|$ съответства единствена редица $\vec{y} \in Y^*$, която представлява едно възможно синтезирано произношение на $\vec{x}$. Избирайки конкретен фон за дадена позиция, евентуални прекъсвания и артефакти биха се появили на границите с неговите съседи. Затова дефинираме ценови функции $VC : V \to R$ (по върховете) и $EC : E \to R$ по ребрата на $G(\vec{x})$.

Без ограничение на общността можем да предположим, че:

$$\begin{aligned} VC(x_i, y_i) &= \sum_{j \in I_g} \mu_j g_j(x_i, y_i) \\ EC(x_i, y_i, x_i', y_i') &= \sum_{j \in I_f} \lambda_j f_j(x_i, y_i, x_i', y_i') \end{aligned} \tag{4.3}$$

където I_f и I_g са някакви индексни множества, а f_j, g_j - функции със стойности от $\mathbb{R}$. Така при фиксирани λ_i, μ_i задачата се свежда до намиране на минимален път в графа $G(\vec{x})$ за дадено $\vec{x}$, като

$$C_{G(\vec{x})}(\vec{y}) = VC(x_0, y_0) + \sum_{i=1}^{|\vec{x}|} (VC(x_i, y_i) + EC(x_{i-1}, y_{i-1}, x_i, y_i)) \tag{4.4}$$

Параметрите λ_i, μ_i на модела ще изберем така, че да минимизират функцията $Target$ върху корпуса, т.е. върху известни идеални сигнали. Ще означим тренировъчна извадка от корпуса с $C = \{(\vec{x_1}, \vec{y_1}), ..., (\vec{x_N}, \vec{y_N})\}$. Тогава параметрите ще изберем така, че да минимизират:

$$\sum_{\vec{x}, \vec{y} \in C} Target(\vec{x}, \vec{y}) \tag{4.5}$$

В Секция 7.4 ще опишем някои мерки за близост между речеви сигнали, които ще използваме като функции $Target$.

Подобен модел е предложен още от [5] и е широко използван при конкатенативния подход.

4.2 Условно случайно поле

Задачата можем да разгледаме и от вероятностна гледна точка. Нека имаме елементарни събития от $X \times Y$. Нека χ е случайна величина. Ще отбелязваме събитията $(\chi = x) = \{(a, b) | a = x, b \in Y\}$ и аналогично $(\chi = y) = \{(a, b) | a \in X, b = y\}$. Нека $n \in \mathbb{N}$ е фиксирано и имаме множества от случайни величини $\vec{X} = X_1, X_2, ..., X_n$ и $Y = Y_1, Y_2, ..., Y_n$. $\vec{X}$ ще наричаме входни, а $\vec{Y}$ - изходни. Ще отбелязваме събитието $(X_1 = x_1, X_2 = x_2, ..., X_n, = x_n)$ като $(\vec{X} = \vec{x})$ или просто $\vec{x}$, съответно $(\vec{Y} = \vec{y})$ като $(Y_1 = y_1, Y_2 = y_2, ..., Y_n = y_n)$ или $\vec{y}$.

Ако разгледаме X като входна азбука, а Y - изходна, интересно за нас е съвместното разпределение на $\vec{X}$ и $\vec{Y}$, т.е. $P(\vec{X} = \vec{x}, \vec{Y} = \vec{y})$. Нека имаме фиксирано разпределение $P(\vec{x}, \vec{y})$, т.е.

$$\forall \vec{x}, \vec{y} : 0 \leq P(\vec{x}, \vec{y}) \leq 1$$

$$\sum_{\substack{\vec{x} \in X^n \\ \vec{y} \in Y^n}} P(\vec{x}, \vec{y}) = 1$$

тогава:

$$P(\vec{x}) = \sum_{\vec{y} \in Y^n} P(\vec{x}, \vec{y})$$

$$P(x) = \sum_{i=1}^{n} \sum_{\substack{\vec{x} \in X^n \\ X_i = x}} P(\vec{x})$$

$$P(\vec{y}) = \sum_{\vec{x} \in X^n} P(\vec{x}, \vec{y})$$

$$P(y) = \sum_{i=1}^{n} \sum_{\substack{\vec{y} \in Y^n \\ Y_i = y}} P(\vec{y})$$

Така за всяко $n \in \mathbb{N}$ получаваме различно вероятностно пространство. $\vec{x}$ са входните описания за нашия компонент, а $\vec{y}$ - известните единици, от които можем да конструираме сигнал за произнасянето на $\vec{x}$. Целта ни е да намерим такова разпределение, че $P(\vec{x}, \vec{y}') > P(\vec{x}, \vec{y}'') \implies \vec{y}'$ е "по-естествен" от $\vec{y}''$. При тази формулировка задачата за намиране на най-подходяща поредица известни единици $\vec{y}$ по зададени описания $\vec{x}$ се свежда до намирането на:

$$\hat{\vec{y}} = \underset{\vec{y} \in \vec{Y}}{\operatorname{argmax}} P(\vec{y}|\vec{x})$$

За да уточним вида на разпределението $P(\vec{X} = \vec{x}, \vec{Y} = \vec{y})$, една подходяща рамка е т. нар. условно случайно поле (УСП, Conditional Random Field) [6].

УСП предполага, че $P(\vec{x}, \vec{y})$ може да се разложи на множители по начин, съответстващ на топологията на неориентиран граф. По-точно:

Def. Нека G е неориентиран граф, $G = (V, E), V = \vec{Y}$, т.е. върховете са случайните величини от $\vec{Y}$ и:

$$P(Y_i = y_i|\vec{x}, Y_j = y_j, Y_i \neq Y_j) = P(Y_i = y_i|\vec{x}, Y_j = y_j, Y_i \sim Y_j) \tag{4.6}$$

където $Y_i \sim Y_j \Leftrightarrow (Y_i, Y_j) \in E$. Разпределението, зададено от G, ще наричаме условно случайно поле.

При предположението (4.6) разпределението има вида:

$$\begin{aligned} P(\vec{x}, \vec{y}) &= \frac{1}{Z} \prod_k \Phi_k(\vec{x}, \vec{y}), \\ Z &= \sum_{\vec{x}, \vec{y}} \prod_k \Phi_k(\vec{x}, \vec{y}) \end{aligned} \tag{4.7}$$

където $\forall k, (\vec{x}, \vec{y}) \in (X \times Y)^n : \Phi_k(\vec{x}, \vec{y}) > 0$. Φ_k съответстват на максималните клики на графа G. По-точно, $\Phi_k(\vec{x}, \vec{y}) = \Phi_k(C_k(G))$, където $C_k(G)$ е k-тата

максимална клика [2] на G (според някакво тяхно изброяване). Според [7] условието (4.6) е еквивалентно на това Φ_k да имат вида:

$$\Phi_k(\vec{x}, \vec{y}) = exp(\phi_k(\vec{x}, \vec{y})) \tag{4.8}$$

откъдето

$$P(\vec{x}, \vec{y}) = \frac{exp(\sum\limits_k \phi_k(\vec{x}, \vec{y}))}{\sum\limits_{\vec{x'}, \vec{y'}} exp(\sum\limits_k \phi_k(\vec{x'}, \vec{y'}))} \tag{4.9}$$

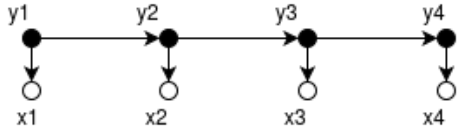
За условната вероятност $P(\vec{y}|\vec{x})$ имаме:

$$\begin{aligned} P(\vec{y}|\vec{x}) &= \frac{P(\vec{x}, \vec{y})}{P(\vec{x})} = \frac{\frac{1}{Z} exp(\sum\limits_k \phi_k(\vec{x}, \vec{y}))}{\sum\limits_{\vec{y'}} (\frac{1}{Z} exp(\sum\limits_k \phi_k(\vec{x}, \vec{y'})))} \\ P(\vec{y}|\vec{x}) &= \frac{exp(\sum\limits_k \phi_k(\vec{x}, \vec{y}))}{\sum\limits_{\vec{y'}} exp(\sum\limits_k \phi_k(\vec{x}, \vec{y'}))} \end{aligned} \tag{4.10}$$

В тази рамка, единствено е необходимо да дефинираме дъгите на графа - съседствата (зависимостите) между величините $\vec{Y}$. Ще се спрем на най-простия вариант - $V = \{(Y_i, Y_{i+1})\}_{i=1}^{n-1}$. Така графът ще е обикновена верига (Фигура 2), а за ϕ_k, ще използваме следния вид:

$$\phi_i(\vec{y}, \vec{x}) = \phi(y_i, y_{i+1}) = \sum_m \lambda_m f'_m(y_i, y_{i+1}), i = 1, ..., n-1$$
$$\phi_j(\vec{y}, \vec{x}) = \phi(y_i, x_i) = \sum_m \mu_m g'_m(x_i, y_i), j = 1, ..., n$$

където $\lambda_m, \mu_m \in \mathbb{R}$ са параметри, които ще нагласяме така, че да намерим оптималното разпределение.



Фигура 2: *Линеен CRF, при $V = \{(Y_i, Y_{i+1})\}_i^{n-1}$. Празните точки отразяват факта, че такава зависимост съществува, но не са елементи графа.*

Ще отбележим, че условието $|\vec{x}| = |\vec{y}|$ съответства на ограничението на всяко $x_i \in \vec{x}$ да бъде съпоставен единствен етикет $y_i \in Y$. На практика

[2] "Клика" като пълен подграф на G, а не като подмножество на върховете на G.

това означава да нямаме възможността да получим съответстващ сигнал за дадено x_i като комбинация на два или повече y'_i, y''_i. С други думи, при този вариант на графа "детайлността" на $\vec{y}$ е поначало фиксирана, което съответства на хомогенността на фонетичните единици в X и Y. Също така, локализирайки по този начин въздействията на етикетите един върху друг, не бихме могли да вземем предвид някои езикови явления като например озвучаване и обеззвучаване под въздействието на единици, които не се намират непосредствено до дадена позиция в редицата. Една по-съвършена система би могла да използва взаимнозаменяемо двойки звучни/беззвучни фонеми в някои ситуации, ако това ще доведе до по-добро качество на изходния сигнал. Ще смятаме, че този тип проблеми се решават при генерирането на входните описания и първоначалната анотация на речевия корпус.

В крайна сметка условното разпределение има вида:

$$P(\vec{y}|\vec{x}) = \frac{exp(\sum_{i=1}^{n-1}\sum_{k}\lambda_k f'_k(y_i, y_{i+1}) + \sum_{i=1}^{n}\sum_{k}\mu_k g'_k(y_i, x_i))}{\underbrace{\sum_{\vec{y'}\in Y^*} exp(\sum_{i=1}^{n-1}\sum_{k}\lambda_k f'_k(y'_i, y'_{i+1}) + \sum_{i=1}^{n}\sum_{k}\mu_k g'_k(y'_i, x_i))}_{Z(\vec{x})}} \quad (4.11)$$

$Z(\vec{x})$ на практика е константа, зависеща от корпуса, която не е необходимо да бъде изчислявана експлицитно. Ако за характеристични функции f'_k, g'_k използваме описаните при (4.4) функции $f(x_i, y_i)$ и $g(y_{i-1}, y_i)$ за всички клики, то получаваме еквивалентен на модела (4.4):

$$P(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} exp(\sum_{i=1}^{n-1}\sum_{k\in I_f}\lambda_k \frac{1}{f_k(y_i, y_{i+1})} + \sum_{i=1}^{n}\sum_{k\in I_g}\mu_k \frac{1}{g_k(y_i, x_i)}) \quad (4.12)$$

Тъй като двата модела са еквивалетнтни, оттук нататък ще използваме понятията "най-вероятен път" и "минимален път" като взаимнозаменяеми.

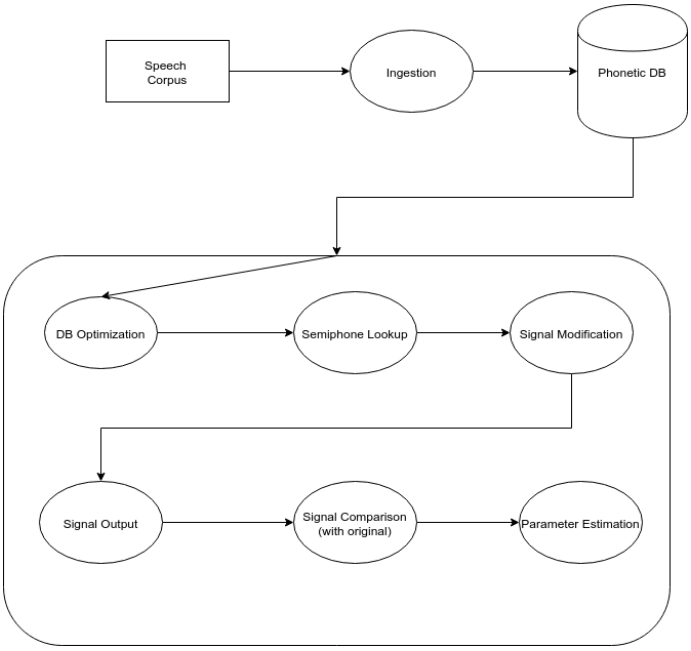
Едно от най-интересните свойства на CRF е, че функцията на правдоподобие има единствен екстремум. За съжаление, както ще видим по-късно, в нашия случай обичайното трениране чрез оптимизиране на тази функция не би могло да доведе до оптимални параметри.

За намиране на минималния/най-вероятен път използваме добре известна процедура с динамично програмиране - алгоритъм на Витерби [9].

5 Архитектура

5.1 Речеви корпус

За разработка на модулът е използвана част от корпуса BulPhonC [10]. Използваният речеви корпус се състои от 320 изречения с обща дължина



Фигура 3: *Архитектура на модула и на процеса на синтез*

приблизително 40 минути, прочетени от женски глас. От тях, 300 се използват като фонетична база, 10 за оптимизиране на коефициентите λ_i и μ_i и 10 за оценка на модела. Всяко изречение има прилежаща анотация с граници на съдържащите се в него фони - примерът в Секция 4.1 е едно от тях. Също така корпусът съдържа и прилежащи сигнали от ларингограф, използван при записването. Приложение А съдържа описание на възможните етикети на фони.

5.2 Извличане на характеристични стойности

Тъй като не разполагаме с особено голям корпус, за фонетична единица ще използваме полуфонът. Интуитивно, това би трябвало да ни даде по-голяма възможност за "изглаждане" на сигнала и донякъде да компенсира малкият избор на единици със съвпадащ контекст. За извличане на свойства на единиците от корпуса използваме програмата Praat [8], която разполага с голям набор от функции за анализ на реч. По-точно, необходимите ни процедури са за намиране фундаменталната честота на сигнал (pitch detection), pitch marking - процес, който маркира границите на периодите на фундаменталната честота, когато такава има (т.е. при гласни и звучни съгласни), и изчисление на Mel Frequency Cepstrum коефициенти (MFCC, описани в Секция 6.3).

За всеки звуков файл и прилежащата му анотация процедурата по извличане е следната.

- Всички поредици от фони, състоящи се само от шум или тишина, биват сляти в едни фон, маркиран като "шум". Единичните такива не биват променяни.
- Границите на фони с продължителност под 30 милисекунди биват изместени така, че да се осигури тази минимална дължина - всъщност, необходимият резултат е всеки полуфон да има дължината на поне един фундаментален период - долната граница в този случай е 66 Hz. Когато това не е възможно, например понеже изместване границите на някой фон би направил съседен фон прекалено малък, даваме етикет на дясна полуфонема на целия интервал.
- Извличаме списък с маркери от сигнала като прилагаме процедурата за pitch marking върху съответният сигнал от ларингографа за изречението. Маркерите представляват граници на периодите на фундаменталната честота. За маркери обикновено се взимат семплите с максимална амплитуда. Тъй като този сигнал е с много по-изчистен спектър, процедурата за pitch marking е по-точна върху него, отколкото върху оригиналният сигнал.

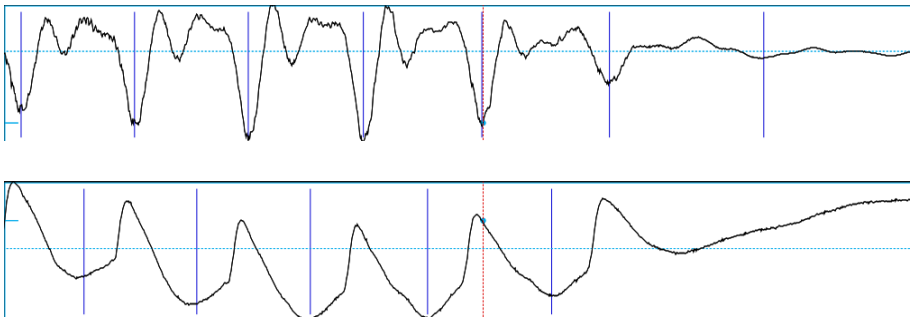


Таблица 1: *Pitch-detection методът, приложен върху сигнал от глотограф (2) е по-точен, отколкото върху звуковия сигнал (1).*

Областите, в които има такива маркери, са гласни или звучни съгласни. За останалите допълваме списъкът с маркери така, че максималното разстояние между две съседни точки да бъде по-малко или равно на 20 милисекунди. Между всеки две съседни граници от анотацията на сигнала намираме семпъла s с най-голяма абсолютна стойност (т.е. точката на максимална амплитуда на сигнала), както и най-малкия маркер p, който е по-голям от s. p използваме като граница на полуфонът. За да получим крайните граници на полуфоните, закръгляме надолу всяка граница от анотацията до маркер или най-близкият свободен такъв, ако на даден

маркер вече има граница. Закръглянето надолу е от особена важност за класа на експлозивните фонеми ("д", "б", "т" и др.). Те се учленяват чрез пълно прегражданe на въздушната струя и в момента на преодоляване се чува ясна "експлозия", която е кратка по продължителност. Чрез закръглянето надолу си осигуряваме, че експлозията ще попадне в десен полуфон, освен ако целият фон не е прекалено кратък. Ако не направим закръглянето до маркер, то последващата модификация на парчетата дава артефакти в синтезираната реч - "замазване" и неразбираемост на части от сигнала. Причината за това се дължи на избрания алогоритъм, описан в Секция 8.

Def. Фреймове ще наричаме застъпващи се отрязъци от сигнала с еднаква дължина и с начални точки, разположени на равни интервали (например с дължина 30 ms и начала на точки от сигнала 0 ms, 15 ms, 30 ms, и т.н.).

Тъй като се застъпват, когато говорим за точка и съдържащият я фрейм, ще имаме предвид този, с най-малка начална точка. Изчисляваме първите 12 MFC коефициенти на сигнала във фреймове от 30 милисекунди.

За всеки от така получените полуфони x записваме във фонетичната база следната информация:

- $id(x) \in L$ - фонетичен етикет (идентификатор на алофонът плюс "флаг" дали полуфонът е ляв или десен.)
- $d_s(x)$,$d_e(x) \in \mathbb{R}$ - отместване в сигнала на изречението за начало и край на полуфонът.
- $c_l(x)$,$c_r(x) \in L$ - фонетични етикети съответно на ляв и десен съсед в изречението (или $\$ \notin L$ ако такъв няма).
- $p_l(x)$,$p_r(x) \in \mathbb{R}$ - стойност на фундаменталната честота в херци в началото и края на единицата в херци.
- $MFCBegin(x)$,$MFCEnd(x)$ - вектори от MFC коефициенти във фреймовете, съдържащи началото и краят на единицата. Извличането им е описано в Секция 6.3.
- $PitchMarks(x) \in 2^{\mathbb{R}}$ - всички маркери, намиращи се в рамките на единицата от оригиналното, необогатено множество.

За допълнителна ефективност и по-добро използване на процесорния кеш на всеки фон даваме пореден номер (идентификатор), като на фони с един и същи етикет даваме поредни идентификатори. По-късно сортираме фоните по този идентификатор. Записваме цялата информация в двоичен формат.

6 Характеристични функции

6.1 Фундаментална честота

Според source-filter модела, речта се състои от два независими един от друг компоненти - източник на вибрации (осцилатор, гласните струни) и линеен акустичен филтър (гласовият тракт). Фундаменталната честота (pitch) е честотата на трептене на осцилатора. Понеже искаме да постигнем максимално близък сигнал до някакъв оригинал, при този модел на речта очевидно не можем да постигнем каквато и да е близост без източникът да трепти с една и съща честота. Също така, след намиране на най-подходящата поредица от фонеми, за да произведем желания сигнал, т.е с желаната интонация, ще ни е необходима допълнителна модификация на сигнала. Всяка такава модификация, обаче, прави речта по-неестествено звучаща. Желателно е тази модификация да бъде сведена до минимум и да използваме фони, които са максимално близо до желаните такива. Затова:

$$g_1(x_i, y_i) = \left|\log_{10} \frac{p_l(x_i)}{p_l(y_i)}\right| + \left|\log_{10} \frac{p_r(x_i)}{p_r(y_i)}\right| \tag{6.1}$$

Ако z е беззвучен фон, т.е. няма фундаментална честота, използваме стойността на най-близкия звучен фон в същото изречение.

В естествената реч прекъсванията във фундаменталната честота на звучни фони са рядко срещани. За да осигурим "по-гладък" преход и съответно по-малка необходима модицикация на фоните, въвеждаме и

$$f_1(x_i, y_i, x'_i, y'_i) = \left|\log_{10} \frac{p_l(y_i)}{p_r(y'_i)}\right| \tag{6.2}$$

Използваме логаритъм поради две причини. Това наподобява човешкият слух, т.к. например разликата между тон 220 херца и 440 херца се възприема за същата като между 440 и 880. Втората причина е, че за нас е по-важно доколко един сигнал ще се модифицира, за да се получи желаната единица. С други думи, важен е коефициентът на скалиране.

6.2 Функция на контекста

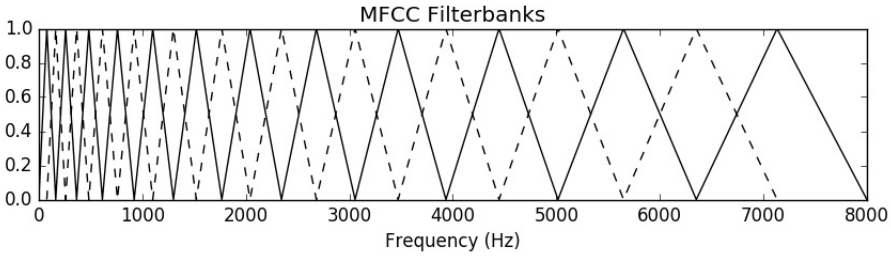
Въпреки че има някакви граници между отделните фони в дадено изречение, тези граници никога не могат да бъдат ясни, т.е. те са винаги размити. Въпреки че във вътрешността на даден фон звукът е чист, то към краищата винаги остава някакъв преход между съседните такива. Затова смятаме за "най-важна" функцията на контекста, т.е:

$$f_2(x_i, y_i, x'_i, y'_i) = \begin{cases} 0, & \text{if } c_r(y_i) = c_l(y'_i), \\ 1, & otherwise \end{cases} \tag{6.3}$$

В примерът в Секция 4.1 тази функция сама по себе си би била достатъчна, за да получим най-подходящият речеви сигнал.

6.3 MFC Коефициенти

Гласовият тракт, макар и да е трудно постижимо да бъде в идентична конфигурация при две отделни произношения на фонема, трябва да бъде максимално близък. По-точно, имайки source-filter модела предвид, бихме искали максимално близки филтри при целевото и синтезираното произношение. Mel-Frequency Cepstrum коефициентите са един начин да оценим тази разлика.



Фигура 4: *Разпределение на триъгълните филтри, използвани за пресмятане на MFC коефициенти, върху честотната скала.*

Мел скалата, представена от [11] е субективна (логаритмична) скала, която има за цел да отрази връзката между фундаменталната честота на даден звук и възприетия тон. В общият случай преобразуването до и от Мел скала се извършва по следният начин:

$$\begin{aligned} Mel(h) &= n \log_{10}(1 + \frac{h}{p}) \\ Mel^{-1}(m) &= p(10^{\frac{m}{n}} - 1) \\ n, p &\in \mathbb{R} \end{aligned} \tag{6.4}$$

Използваме $n = 2595, p = 700$ т.к. това са стойностите, избрани в Praat [8].

Ще дадем кратко описание на получаването на MFCC. Нека имаме периодична функция $s(t)$ - сигнал.

- Разбиваме сигнала на фреймове по 50 милисекунди $s_i(t)$. За всеки фрейм $s_i(t)$ прилагаме трансформация на Фурие $FT(s_i)$ и получаваме комплексния спектър $S_i(f)$. Нека $P_i(f) = |S_i(f)|^2$, т.е. P_i е спектърът на енергията на сигнала във фрейма s_i.
- Фиксираме минимална честота f_{min} и максимална честота f_{max}, например $f_{min} = 100Hz, f_{max} = 8000Hz$.
- На равни разстояния между $Mel(f_{min})$ и $Mel(f_{max})$ взимаме точки $m_i, i = 0, ..., n$, като $m_0 = f_{min}$ и $m_n = f_{max}$ (обичайно $n = 24$).

- За всяка двойка (m_i, m_{i+1}), $i = 0, ..., n-1$ дефинираме триъгълен филтър MF_i за честотите между $f_l = Mel^{-1}(m_i)$ и $f_h = Mel^{-1}(m_{i+1})$ с връх в $f_c = \frac{1}{2}(Mel^{-1}(m_i) + Mel^{-1}(m_{i+1}))$. По-точно:

$$MF_i(f) = \begin{cases} \frac{f - f_l}{f_c - f_l}, f_l \leq f \leq f_c \\ \frac{f_h - f}{f_h - f_c}, f_c < f \leq f_h \\ 0, f < f_l \\ 0, f > f_h \end{cases} \tag{6.5}$$

- За всеки MF_i пресмятаме коефициентите C_k^i:

$$C_k^i = \sum_{f=1}^{N} MF_i(f) P_i(f) \cos\left(\frac{\pi k(f - 0.5)}{N}\right) \tag{6.6}$$

  Стойностите на хармоничните честоти в така получения дискретен кепстър се наричат MFC коефициенти.

- Използваме коефициентите $C_2^i, ..., C_{13}^i$. Същите се използват обичайно и в автоматичното разпознаване на реч [15]. Така за всеки фрейм получаваме характеристичен вектор от 12 стойности.

За характеристики на фоните използваме MFCC векторите на първия и последен фрейм, застъпващи се с границите на y. Ще ги обозначаваме съответно $MFCBegin(y)$ и $MFCEnd(y)$. Дефинираме:

$$f_3(x, y, x', y') = \sum_i (MFCBegin(y)_i - MFCEnd(y')_i)^2 \tag{6.7}$$

Очевидно при два съседни фона y и y', крайният фрейм на y и началният фрейм на y' ще съвпадат, т.е. $f_2(x, y, x', y') = 0$.

6.4 Продължителност

Имайки зададена от потребителя целева продължителност на даден фон и разполагайки с ограничена база, съвсем естествено е след процеса на избор на фони да се наложи допълнителна модификация на сигнала, за да се постигне желаната продължителност. Удължаването или скъсяването на фон включва дублициране и/или "изпускане" на части от сигнала. В Секция 8 ще опишем подробно избрания алгоритъм за пост-селекционна модификация. Действието на алгоритъма е от голяма важност за някои класове фонеми, например експлозивните звуци, заради тяхната неравномерна структура - затишие и краткотрайна "експлозия". Осигурили сме вече експлозията да се намира само в десни полуфони, но при прекомерно

удължаване или съкращаване е възможно експлозията да бъде изкривена, т.е. например да бъде изпусната част от нея, или пък дублицирана. За други класове фонеми, например гласните, промяната на продължителността е от по-малко значение, тъй като те се учленяват по по-прост начин. По-точно, при тях въздушната струя преминава почти свободно през гласовия тракт и съответно акустичният филтър е по-прост и не претърпява съществени изменения в рамките на даден фон. Същото важи и за проходните фонеми ("с", "в", "ф" и др.). Дефинираме:

$$g_2(x,y) = max(\left|\ln \frac{Duration(x)}{Duration(y)}\right|, \left|\ln \frac{Duration(y)}{Duration(x)}\right|) \tag{6.8}$$

където $Duration(x) = d_e(x) - d_s(x)$. Използваме логаритъм, тъй като важен е коефициентът на разпъване или свиване, а не абсолютната разлика в продължителностите.

6.5 Енергия

Нека $s(t)$ е сигнал. Енергия E на s в интервала (t_1, t_2) ще наричаме:

$$E(s)(t_1,t_2) = \int_{t_1}^{t_2} s(t)^2 dt \tag{6.9}$$

Дефинираме енергията за единица време в рамките на y:

$$Power(y) = \frac{E(s)(d_s(y), d_e(y))}{Duration(y)} \tag{6.10}$$

и съответната характеристична функция:

$$g_3(x,y) = \frac{Power(y)}{Power(x)} \tag{6.11}$$

Високата енергия съответства на по-силен сигнал (по-силен звук). За дадена позиция трябва да търсим фонема с максимално близка енергия. Въпреки че това е от особена важност е за ударени фонеми/срички, тази функция не допринася особено за качеството върху оценъчната част от корпуса, тъй като стилът на прочит на изреченията не е особено разнороден. По-точно, макар и да се срещат разнообразни откъм височина или скорост на прочитане изречения, по време на записът на корпуса средната енергия на сигнала е държана близко до някаква основна стойност. За разлика от предните характеристични функции тук не използваме логаритъм - възприемането на силата на звука е известно изключение от иначе логаритмичните осезания.

7 Оптимизиране параметрите на модела

7.1 Мотивация

Споменахме вече, че класическият начин на трениране на CRF не е приложим за тази задача. По-точно, обичайната процедура е чрез метода на максималното правдоподобие да се намери единственият максимум на функцията на правдоподобие за даден корпус (на практика - логаритъм от нея). Нека имаме извадка от фонетичният корпус $C = \{(\vec{x_1}, \vec{y_1}), ..., (\vec{x_N}, \vec{y_N})\}$. Функцията на правдоподобие за C е:

$$\begin{aligned} Likelihood(C) &= \prod_{i=1}^{N} P(\vec{y_i}|\vec{x_i}) \\ LogLikelihood(C) &= \ln Likelihood(C) = \sum_{i=1}^{N} \ln P(\vec{y_i}|\vec{x_i}) \end{aligned} \tag{7.1}$$

За случая на CRF [3] имаме:

$$\begin{aligned} P(\vec{y}|\vec{x}) &= \frac{1}{Z(x)} \sum_{i=1}^{i=|\vec{x}|} \exp(\sum_k \lambda_k f_k + \sum_m \mu_m g_m) \\ Z(x) &= \sum_{\vec{y'} \in \vec{Y}} \exp(\sum_k \lambda_k f_k + \sum_m \mu_m g_m) \end{aligned} \tag{7.2}$$

Ще припомним, че C се състои изцяло от естествени произношения и че $\vec{x}_j = \vec{y}_j, \forall j = 1, ..., N$. Тогава за дадени $x_i, x_{i+1} \in \vec{x}_j$ и съответстващите им $y_i, y_{i+1} \in \vec{y}_j$ имаме:

$$\begin{aligned} f_1(x_i, y_i, x_{i+1}, y_{i+1}) &\geq 0 \\ f_2(x_i, y_i, x_{i+1}, y_{i+1}) &= 0 \\ f_3(x_i, y_i, x_{i+1}, y_{i+1}) &= 0 \text{ [4]} \\ g_1(x_i, y_i) &= 0 \\ g_2(x_i, y_i) &= 1 \end{aligned} \tag{7.3}$$

Повечето функции естествено получават стойност 0, а стойностите на f_2 също са близки до 0. При така зададените характеристични функции стойността на $Likelihood(C)$ не зависи от техните стойности, а оттам и от параметрите на модела. Очевидно прилагането на оптимизационна процедура не би довело до смислен резултат. По-формално, проблемът е

[3] За простота пишем f_i, g_i вместо $\frac{1}{f_i}, \frac{1}{g_i}$

[4] Освен в редките случаи, когато границата между единиците съвпадне с граница на фрейм.

във факта, че входните и изходни описания в корпуса са изоморфни, т.е. че $\vec{x}_i = \vec{y}_i$ или:

$$\begin{aligned} &P(\vec{y}|\vec{x}) = P(\vec{y}|\vec{y}) = P(\vec{x}|\vec{x}) = P(\vec{x}) = P(\vec{y}) \\ &\implies LogLikelihood(C) = \sum_{i=1}^{N} \ln P(\vec{x_i}) \end{aligned} \tag{7.4}$$

За да можем да използваме функцията на правдоподобие като целева функция, са ни необходими друг тип данни - по-точно двойки вектори различни входни и изходни описания. Всъщност имаме нужда от изкуствено (ръчно) създаден корпус, състоящ се от примерни конкатенации на фони от нашата база, ръчно определени за естествени. Разполагайки с такива данни, тогава параметрите на вероятностното пространство биха могли да бъдат намерени и чрез метода на максималното правдоподобие, например.

7.2 Алгоритъм

За решение на задачата предлагаме вариант на търсене в решетка (grid search). Ще отбелязваме с $\vec{x}_1, ..., \vec{x}_n$ изреченията от корпуса, избрани за трениране. Нека имаме $\theta = (\lambda_1, ..., \lambda_r) \in \mathbb{R}^r$ - параметрите на модела като точка в r-мерното Евклидово пространство. Нека също така:

$$\begin{aligned} &C : \mathbb{R}^r \times Y^* \times X^* \to \mathbb{R} \quad && C(\theta, \vec{x}, \vec{y}) = \theta . \sum_{i=0}^{|\vec{x}|} \vec{f}(x_i, y_i, y_{i+1}) \\ &C_{min} : \mathbb{R}^r \times X^* \to Y^* \quad && C_{min}(\theta, \vec{x}) = \underset{\vec{y} \in Y}{\operatorname{argmin}} C(\theta, \vec{x}, \vec{y}) \\ &F_{\vec{x}} : X^* \to \mathbb{R} && F_{\vec{x}}(\theta) = Target(\vec{x}, C_{min}(\theta, \vec{x})) \\ &F : \mathbb{R}^r \to \mathbb{R} && F(\theta) = \sum_{\vec{x} \text{ е от корпуса}} F_{\vec{x}}(\theta) \end{aligned} \tag{7.5}$$

Целта ни е да намерим $\hat{\theta} = \underset{\theta}{\operatorname{argmin}} F(\theta)$.

За да определим решетката за търсене, трябва да вземем предвид няколко неща. Нека сме намерили $C_{min}(\theta, \vec{x}_i) = \vec{y}^1$ и имаме фиксирана посока на търсене $\delta \in \mathbb{R}^r$. За удобство ще записваме стъпката при търсене като

$$k.\delta, |\delta| = 1, k > 0$$

Да предположим, че следващият най-добър път по посоката δ е $\vec{y}^2 \neq \vec{y}^1$. Следователно има някакъв минимален праг в цената (а оттам и в размера на стъпката), който трябва да бъде "превъзмогнат", за да настъпи промяна в стойността на F. Нещо повече, този праг зависи както от θ, така и от

δ. Следователно ако при търсенето да използваме предварително фиксирана константа като стъпка за придвижване в пространството, то ще направим евентуално повече пресмятания на F от необходимото и също така бихме могли да "прескочим" глобалния минимум на функцията. Т.к. кодомейнът на C_{min} е крайно множество, то съществува и някаква граница, отново зависеща от θ и δ, отвъд която по-нататъшно следване на дадено δ не би довело до промяна стойността на F.

В следващата секция ще видим как можем да определим тези прагове. При фиксирани параметри θ и посока δ, минималният размер на стъпката, водещ до промяна в стойността на F ще отбелязваме с $MinD(\theta, \delta) \in \mathbb{R}^+$.

Използваният тук алгоритъм за оптимизиране на θ е следният алчен вариант на търсене в решетка. $MaxPasses$ и N_k са параметри на процедурата, задаващи горна граница на броя пресмятания на F, които процедурата ще извърши. $MaxPasses$ задава брой "завъртания" по възможните посоки на търсене, докато N_k налага ограничение на броят стъпки, извършени по дадена посока. Посоката на търсене δ избираме последователно от множеството на единичните вектори, успоредни на координатните оси $E = \{\pm e_i\}_{i=1}^{r}$, като целенасочено избираме положителна и отрицателна посока по дадена координатна ос една след друга. С други думи, извършваме търсенето по дадена ос последователно в двете посоки.

В псевдокод процедурата изглежда така:

1. $MaxPasses \in \mathbb{N}, N_k \in \mathbb{N}$
2. $\hat{\theta} \in \mathbb{R}^r, \hat{\theta} \neq (0, 0, ..., 0)$
3. $BestValue = F(\hat{\theta}), i = 0, j = 0$
4. Избери посока $\delta \in \mathbb{R}^r$
5. $\theta_i = \hat{\theta}$
6. $j = j + 1$
7. $n_k = 0$
8. Докато $MinD(\theta_i, \delta) > 0$ и $n_k < N_k$:
   - 8.1. $\theta_{i+1} = \theta_i + \delta.MinD(\theta_i, \delta)$
   - 8.2. $\hat{\theta} = \begin{cases} \theta_{i+1} & \text{, ако } F(\theta_{i+1}) < BestValue \\ \hat{\theta} & \text{, иначе} \end{cases}$
   - 8.3. $BestValue = F(\hat{\theta})$
   - 8.4. $n_k = n_k + 1$
   - 8.5. $i = i + 1$
9. Ако $j \leq MaxPasses$ GOTO стъпка 4
10. Stop

След завършване на процедурата $\hat{\theta}$ съдържа намерената най-добра стойност на параметрите θ. Очевидно по този начин не можем да гарантираме намиране на глобалния минимум върху решетката, тъй като не обхождаме цялата. Обхождането на цялата решетка смятаме за непосилно с толкова наивен алгоритъм, т.к. броят точки в общият случай е поне $\Omega(|C|^r)$.

7.3 Избор на решетка

В тази секция ще опишем как можем да намерим следващата стъпка $MinD(\theta, \delta)$ в процедурата на търсене. Ще предложим приближение на $MinD(\theta, \delta)$, вместо намирането на същинското такова.

Преди това ще направим някои опростявания в допълнение към означенията от предната секция.

$$\vec{f}(x_i, y_i, y_{i+1}) = (g_1(x_i, y_i), ..., g_k(x_i, y_i), f_1(y_i, y_{i+1}), ..., f_l(y_i, y_{i+1}))^T$$
$$\vec{f_\sigma}(\vec{x}, \vec{y}) = \sum_{i=0}^{|\vec{x}|-1} \vec{f}(x_i, y_i, y_{i+1})$$

където $y_{|\vec{x}|} = \$, \$ \notin X \cup Y$ и $\forall i \in \mathbb{N}, y \in Y : f_i(y, \$) = 0$. Така имаме:

$$C(\theta, \vec{x}, \vec{y}) = \theta.\vec{f_\sigma}(\vec{x}, \vec{y})$$

Твърдение: Нека $\vec{x} \in \vec{X}, \theta, \delta \in \mathbb{R}^n$ са фиксирани, $\hat{\vec{y}} = C_{min}(\theta, \vec{x})$ и $\vec{y} \neq \hat{\vec{y}}$. Тогава следните са еквивалентни:

1. $\exists k > 0 : C(\theta + k\delta, \vec{x}, \vec{y}) \leq C(\theta + k\delta, \vec{x}, \hat{\vec{y}})$
2. $\delta f_\sigma(\vec{x}, \vec{y}) \leq \delta f_\sigma(\vec{x}, \hat{\vec{y}})$

Това означава, че по посоката δ можем да достигнем промяна в минималния път със стъпка $MinD(\theta, \delta) = k$ точно тогава, когато $\hat{\vec{y}} \neq C_{min}(\delta, \vec{x})$. Нещо повече, както ще видим по-късно, това ни позволява да приближим k с произволна точност.

И така, нека имаме 1, тогава:

$$C(\theta + k\delta, \vec{x}, \vec{y}) \leq C(\theta + k\delta, \vec{x}, \hat{\vec{y}})$$
$$(\theta + k\delta).\vec{f_\sigma}(\vec{x}, \vec{y}) \leq (\theta + k\delta).\vec{f_\sigma}(\vec{x}, \hat{\vec{y}})$$
$$\theta.\vec{f_\sigma}(\vec{x}, \vec{y}) + k\delta.\vec{f_\sigma}(\vec{x}, \vec{y}) \leq \theta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) + k\delta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}})$$
$$k\delta.\vec{f_\sigma}(\vec{x}, \vec{y}) - k\delta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) \leq \theta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) - \theta.\vec{f_\sigma}(\vec{x}, \vec{y})$$

откъдето

$$k\delta.(\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) - \vec{f_\sigma}(\vec{x}, \vec{y})) \geq \theta.\vec{f_\sigma}(\vec{x}, \vec{y}) - \theta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) \quad (7.6)$$

Но $\hat{\vec{y}} = C_{min}(\theta, \vec{x}) = \underset{\vec{y}}{\operatorname{argmin}}\, \theta.\vec{f_\sigma}(\vec{x}, \vec{y})$ и

$$\begin{aligned}
&\Longrightarrow \theta.\vec{f_\sigma}(\vec{x}, \vec{y}) - \theta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) \geq 0 \\
&\overset{(7.6)}{\Longrightarrow} k\delta.(\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) - \vec{f_\sigma}(\vec{x}, \vec{y})) \geq 0 \\
&\Longrightarrow k\delta.\vec{f_\sigma}(\vec{x}, \vec{y}) \leq k\delta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}})
\end{aligned}$$

Обратно, нека 2, т.е. $\delta f_\sigma(\vec{x}, \vec{y}) \leq \delta f_\sigma(\vec{x}, \hat{\vec{y}})$ и нека вземем например:

$$k = \frac{\theta.\vec{f_\sigma}(\vec{x}, \vec{y}) - \theta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}})}{\delta\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) - \delta\vec{f_\sigma}(\vec{x}, \vec{y})} + 1$$

Да разгледаме цените на $\vec{y}$ и $\hat{\vec{y}}$ при $\theta + k\delta$.

$$\begin{aligned}
&C(\theta + k\delta, \vec{x}, \vec{y}) - C(\theta + k\delta, \vec{x}, \hat{\vec{y}}) = \\
&= (\theta + k\delta)f_\sigma(\vec{x}, \vec{y}) - (\theta + k\delta)f_\sigma(\vec{x}, \hat{\vec{y}}) = \\
&= \theta(f_\sigma(\vec{x}, \vec{y}) - f_\sigma(\vec{x}, \hat{\vec{y}})) - k(\delta f_\sigma(\vec{x}, \hat{\vec{y}}) - \delta f_\sigma(\vec{x}, \vec{y})) = \\
&= \theta(f_\sigma(\vec{x}, \vec{y}) - f_\sigma(\vec{x}, \hat{\vec{y}})) - \theta(f_\sigma(\vec{x}, \vec{y}) - f_\sigma(\vec{x}, \hat{\vec{y}})) + \delta f_\sigma(\vec{x}, \vec{y}) - \delta f_\sigma(\vec{x}, \hat{\vec{y}}) = \\
&= \delta f_\sigma(\vec{x}, \vec{y}) - \delta f_\sigma(\vec{x}, \hat{\vec{y}}) \\
&\Leftrightarrow C(\theta + k\delta, \vec{x}, \vec{y}) - C(\theta + k\delta, \vec{x}, \hat{\vec{y}}) \leq 0 \\
&\Leftrightarrow C(\theta + k\delta, \vec{x}, \vec{y}) \leq C(\theta + k\delta, \vec{x}, \hat{\vec{y}})
\end{aligned}$$

С което твърдението е доказано.

Следствие: Нека $\dot{\vec{y}} = \underset{\vec{y} \in \vec{Y}}{\operatorname{argmax}} C(\delta, \vec{x}, \vec{y})$. Тогава $\nexists k > 0, \vec{y} \neq \dot{\vec{y}} : C(\theta + k\delta, \vec{x}, \vec{y}) \leq C(\theta + k\delta, \vec{x}, \dot{\vec{y}})$.

С други думи, ако $\underset{\vec{y} \in Y}{\operatorname{argmin}} C(\theta, \vec{x}, \vec{y}) = \dot{\vec{y}} = \underset{\vec{y} \in \vec{Y}}{\operatorname{argmax}} C(\delta, \vec{x}, \vec{y})$, то по-нататъшно следване на посоката δ не води до промяна в стойността на F.

Така получаваме горна граница за k

$$k'' = \frac{\theta.\vec{f_\sigma}(\vec{x}, \dot{\vec{y}}) - \theta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}})}{\delta.(\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) - \vec{f_\sigma}(\vec{x}, \dot{\vec{y}}))}, \vec{y} \in \vec{Y}, \vec{y} \neq \hat{\vec{y}} \tag{7.7}$$

а една долна граница на $\hat{k}$ е например:

$$k' = \frac{\min\limits_{\vec{y} \in \vec{Y}, \vec{y} \neq \hat{\vec{y}}} (\theta.\vec{f_\sigma}(\vec{x}, \vec{y}) - \theta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}))}{\max\limits_{\vec{y} \in \vec{Y}, \vec{y} \neq \hat{\vec{y}}} |\delta.(\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) - \vec{f_\sigma}(\vec{x}, \vec{y}))|} \leq \frac{\theta.\vec{f_\sigma}(\vec{x}, \vec{y}) - \theta.\vec{f_\sigma}(\vec{x}, \hat{\vec{y}})}{|\delta.(\vec{f_\sigma}(\vec{x}, \hat{\vec{y}}) - \vec{f_\sigma}(\vec{x}, \vec{y}))|} \tag{7.8}$$

За да намерим следваща стъпка в процедурата на търсене ще апроксимираме $\hat{k}$ в интервала (k', k'') ако то съществува. И така, нека $\vec{x}$ е фик-

сирано и:

$$M(\theta) = \{C_{min}(\theta, \vec{x}) | \vec{x} \in \vec{X}\} \subset 2^{Y^*}$$

$$\Delta_\theta(\theta') = \begin{cases} 0, & \text{ако } M(\theta') = M(\theta), \\ 1, & \text{ако } M(\theta') \neq M(\theta) \end{cases}$$

Очевидно $\Delta_\theta(k') = 0$ и $\Delta_\theta(k'') = 1$ точно тогава, когато $\hat{k} > 0$ съществува.

$\hat{k}$ можем да апроксимираме чрез процедура, подобна на двоично търсене:

$$\begin{aligned} \dot{k}_0 &= \frac{l_0 + r_0}{2}, l_0 = k', r_0 = k'' \\ \dot{k}_n &= \frac{l_n + r_n}{2}, l_n = \begin{cases} l_{n-1}, & \Delta_\theta(k_{n-1}) = 1 \\ k_{n-1}, & \text{иначе}, \end{cases}, r_n = \begin{cases} k_{n-1}, & \Delta_\theta(k_{n-1}) = 0 \\ r_{n-1}, & \text{иначе} \end{cases} \end{aligned} \tag{7.9}$$

Така $\dot{k} = \dot{k}_n + O(\frac{1}{2^n})$. Избираме $\dot{k} = \dot{k}_i$ за някое достатъчно голямо i.

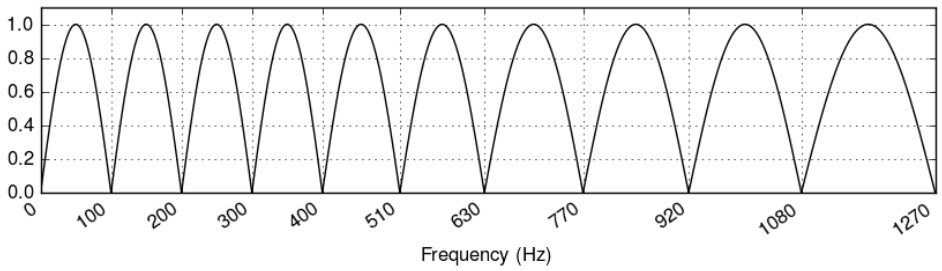
На практика се оказва, че често интервалът е (k', k'') е достатъчно голям, за да е необходимо $i = 30$ или повече. Това прави търсенето сравнително бавно, тъй като в процедурата 7.2 е необходимо поне $MaxPasses \geq$ #(характеристични функции).

В крайна сметка, в процедурата 7.2 в стъпка 8 избираме $MinD(\theta, \delta) = \dot{k}$.

7.4 Избор на функцията Target

Изборът на Target функция е от съществено значение за поведението на синтезатора. Най-важното за тази функция е да отразява колко естествено звучи изходният сигнал. Проблемът за естественото звучене на даден сигнал е обект на изследване на психоакустиката. Въпреки че съществуват много начини за измерване близостта на сигнали, никой от тях не е в състояние да отрази субективната оценка на човек за естествено звучене. [13] прави сравнение на няколко обективни мерки и тяхната корелация с оценките за качество на даден (синтетичен) сигнал, дадени от хора. Резултатите, за съжаление, показват, че няма особено висока корелация между близостта на сигналите и степента на "естественост" на синтетичната реч. Според [13] повечето от използваните мерки не са подходящи за предсказване субективното качество на синтетична реч. Изключение прави PESQ алгоритъмът (Perceptual Evaluation of Speech Quality), разработен за автоматична оценка на качеството на реч, пренесена по телекомуникационни канали (VoIP или телефония).

В тази дипломна работа ще разгледаме няколко възможни функции Target и ефектът от оптимизирането им. Това включва както такива върху времевия домейн, така и спектрални мерки. Резултатите са представени в секция Секция 9. Ще са ни необходими някои означения. Нека:



Фигура 5: *Критични честотни ленти от 1 до 10*

- $s_{\vec{x}(t)}$ е сигналът на $\vec{x}$
- $s^i_{\vec{x}}(t), i = 0, ..., M$ са фреймове от $s_{\vec{x}}$ с дължина d (например 30 милисекунди) и стъпка ν (например 10 милисекунди)
- $P(s^i_{\vec{x}})$ е комплексният спектър на $s^i_{\vec{x}}(t)$, т.е. $P(s^i_{\vec{x}})(f) \in \mathbb{C}$

Log-Spectrum (LS). Първата мярка е разстояние между лог-спектрите на двата сигнала, пресметната при $d = 20$ милисекунди и $\nu = 10$. По-точно:

$$D_{LS}(\vec{x}, \vec{y}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{f} (\log_{10} \frac{|P(s^i_{\vec{x}})(f)|}{|P(s^i_{\vec{y}})(f)|})^2 \tag{7.10}$$

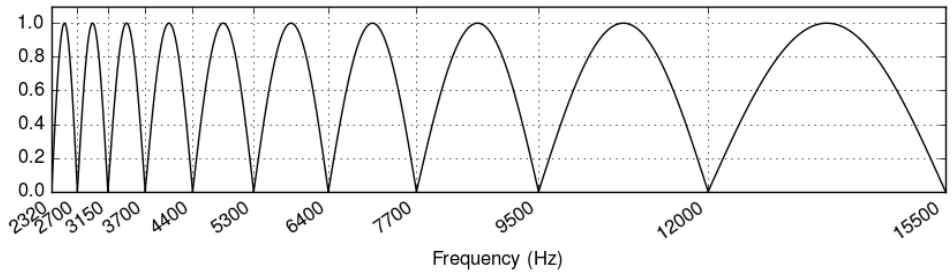
Тази мярка, разбира се, е възможно най-наивната. Очевиден проблем с нея е, че не взима предвид дори малки отклонения в спектрите. По-точно, дори и за съвсем малка разлика, например промяна във фундаменталната честота с няколко херца (нещо, което човек трудно може да долови), води до голяма разлика в разстоянието.

Log-Spectrum-Critical (LSC). Идеята за критична лента (critical band) за дадена честота е предложена от H. Fletcher в [17] и доразвита в [18]. Най-общо казано, честотите от кричината лента на дадена честота могат да я "маскират", т.е. да затруднят нейното отчитане (възприятие). Bark скалата е подобна на Мел скалата - идеята е близки разстояния (в скалата) да съответстват на близко възприятие. [19] разделя честотите до около 15.5 kHz на 24 честотни ленти (критични ленти на слуха). Техните стойности са експериментално установени.

Segmental Signal-To-Noise Ratio (SegSNR). Измерва силата на даден сигнал спрямо шума. Изчислява се директно във времевия домейн:

$$SegSNR(\vec{x}, \vec{y}) = -\frac{1}{M} \sum_{i=1}^{M} \log_{10}(\sum_{t} \frac{s^i_{\vec{x}}(t)^2}{(s^i_{\vec{x}}(t) - s^i_{\vec{y}}(t))^2})$$

$$D_{SegSNR}(\vec{x}, \vec{y}) = \frac{1}{SegSNR(\vec{x}, \vec{y})}$$



Фигура 6: *Критични честотни ленти 15 до 24*

Ще отбележим, че тази мярка изисква много точна синхронизация във времето на двата сигнала - като продължителност и като фазово отместване на честотните компоненти. Обикновено се използва за оценяване на качеството на алгоритми за кодиране/декодиране на звукови сигнали, или при преноса на сигнал по шумен канал.

Weighted Spectral Slope Distance (WSS). Предложена от [16] и използвана с голям успех в ASR. Също така и [13] показва значима, макар и незадоволителна, корелация със субективно оцененото качество на синтетична реч. Използва формата на спектъра, т.е. "наклонът" в определени честотни ленти от спектъра. Целта на тази метрика е да не бъде чувствителна към абсолютните височини на пиковете в спектъра, но в същото време да отразява относителните им разлики. Дефинира се като:

$$WSS(s, s') = K(SPL(s) - SPL(s')) + \sum_f W(f)(SL(s, f) - SL(s', f))^2$$

$$SL(s, f) = |E_s(f + 1)| - |E_s(f)|$$

Тук $K \in \mathbb{R}$ е константа, а $SPL(s)$ е мярка за ефективното налягане на речевия сигнал спрямо някаква базова стойност. Базовата стойност зависи от средата и апаратурата при записа на сигнала. В нашия случай можем да предположим, че $SPL(s) = SPL(s')$, тъй като речевият корпус е съставен в единствено звукозаписно студио и е от единствен говорител.

$W(f)$ е теглова функция, която трябва да вземе предвид дали енергията на честотната лента f е връх или долина (локален максимум или минимум) на енергийния спектър и дали това е най-големият връх в спектъра. Според [16] точният вид на функцията няма голямо значение, стига тези условия да са изпълнени. Тук ще използваме подобна $W(k)$ като в [16] - осреднена стойност между тегла, изчислени поотделно спрямо s и

s'. Нека:

$$W(f) = \frac{W_s(f) + W_{s'}(f)}{2}$$
$$W_s(k) = \frac{20}{20 + E_{s,max} - E_s f} \frac{1}{1 + Peak(s)(f) - E_s(f)}$$
$$E_S(f) = \sum_{k \in f} |P(s)(k)|$$
$$E_{s,max} = \max_f E_s(f)$$

$Peak(s)(f)$ е енергията на най-близкия спектрален "връх". Намира се с търсене "наляво", ако $SL(s, f) < 0$, и "надясно", ако $SL(s, f) > 0$.

Mel Cepstral Distance (MFCC). Предложена от [14]. Базира се на същите MFC коефициенти, описани и използвани в секция Секция 6.3 като характеристична функция. Дефинира се като:

$$D_{MFCC}(\vec{x}, \vec{y}) = \sum_{i=1}^{M} \sqrt{\sum_j (MFC(s^i_{\vec{x}})j - MFC(s^i_{\vec{y}})_j)^2}$$

Това разстояние се използва в автоматичното разпознаване на реч (ASR) [15].

8 TD-PSOLA

Тъй като разполагаме с крайна фонетична база, не можем винаги да постигнем желаната интонация и продължителност на синтетичната реч. Затова е необходимо да прибегнем до метод, чрез който от полуфоните, с които разполагаме, да получим такива с подходяща фундаментална честота и продължителност. Един такъв метод е TD-PSOLA [12] (Time-domain pitch-synchronous overlap-add). В тази секция ще дадем описание на използвания алгоритъм.

PSOLA методите разчитат на предварително поставени маркери във входния сигнал, които съответстват на границите на периоди в псевдопериодичния сигнал (pitch marks). В озвучени части на сигнала (звучни полуфони), те са разположени на разстояние, точно съответстващо на локалната стойност на фундаменталната честота. За граници на периодите обикновено се избират семплите с максимална амплитуда.

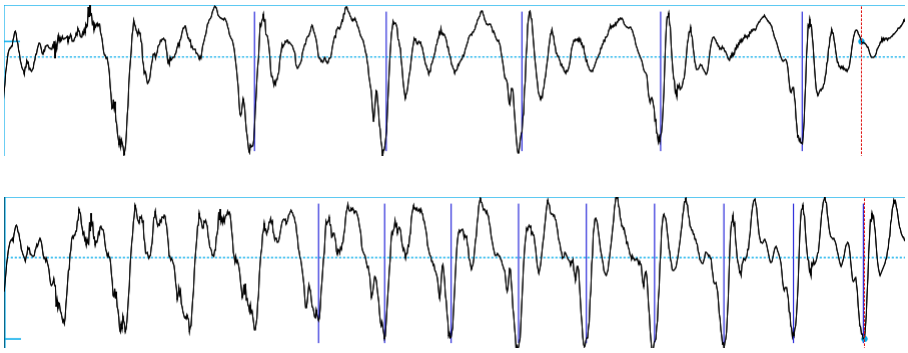


Таблица 2: *TD-PSOLA, приложена върху фонема "а" с фундаментална честота* $138Hz$*, при* $r_p = 2$ *и* $r_d = 1$*. При по-голям* r_p *някои от отрязъците ще трябва да се копират повече от 2 пъти.*

За неозвучените части тези маркери могат да се поставят на фиксирано разстояние един от друг. Нека имаме изходния сигнал $S(t), t \in (0, T)$ и списък от така поставени маркери $p_i, i = 0, ..., N \in \mathbb{N}$. За всеки маркер p_i се извлича кратък отрязък $s_i(t)$ от сигнала с дължина 2 пъти локалният фундаментален период, като средата му съвпада с p_i. Алгоритъмът едновременно модифицира фундаменталната честота и продължителността, като на входните маркери p_i съпоставя нова поредица от маркери $p'_j, j = 0, ..., M$ на разстояние, съответстващо на желаната фундаментална честота. За всеки маркер p'_i от отрязъка $s_i(t), t = p_i - T_i, ..., p_i + T_i$ от оригиналния сигнал получаваме $s'_i(t) = h(t).s_i(t)$, където $h(t) = \frac{1}{2} + cos(2\pi\frac{t}{T_i})$, т.е. оригиналният отрязък, взет с Hanning прозорец с дължина $2.T_i$. Така ако увеличим фундаменталната честота с коефициент $r_p = 1.3$, например, това ще доведе до естествено намаляване на продължителността с коефициент $r_d = 0.7$. Това може да се компенсира, като се копира един и същи сигнал $s_i(t)$ повече от веднъж, така че да се получи максимално близък по продължителност изходен сигнал. Този подход е и причината PSOLA алгоритмите да водят до чувствително изкривяване на сигнала, когато някой от коефициентите на скалиране (на фундаменталната честота r_p или на продължителността r_d) е по-малък от 0.5 или по-голям от 2. Ако един и същи беззвучен $s_i(t)$ се копира прекалено много пъти, това води до изкуствено "озвучаване" на част от изходния сигнал. Един начин това да се избегне е като за всяко последващо копиране отрязъкът се обръща, т.е. $s'_i(t) = s_i(T_i - t)$, или пък в беззвучните участъци се избира променлив "период". Избраният тук подход е друг. При търсенето на подходяща поредица от фонеми, за "възможни" съответствия на дадено x разглеждаме само y такива, че:

$$0.5 \leq \frac{Duration(y)}{Duration(x)} \leq 2$$

С други думи, налагаме експлицитно ограничението. Тъй като избраната звукова единица е полуфон, на практика това ограничение не намалява значително кандидатите, съответстващи на дадено x. Така премахваме случаите на прекомерна модификация, а оттам и изкривяване, породени от TD-PSOLA алгоритъма.

9 Експерименти и резултати

9.1 База за сравнение

Като контролна база ще използваме един наивен подход за намиране на най-подходяща редица полуфони. Нека имаме входно описание $\vec{x}$, за него избираме такова $\vec{y}$, т.ч. броят "парчета" от корпуса да е минимален. Така броят на "връзки" между парчета, които не се срещат наготово в корпуса, ще бъде минимален. Този подход би бил подходящ, ако броят изречения, които системата трябва да може да възпроизведе, е ограничен, или пък разполагаме с много голям корпус. Например в системи за навигация по пътищата или автоматизирани съобщения в градския транспорт.

Това можем да постигнем като в разпределението 4.12 използваме единствена характеристична функция $f_0(y, y')$, дефинирана по следния начин:

$$f_0(y, y') = \begin{cases} 0, \text{ ако } (y, y') \text{ се среща в корпуса} \\ 1, \text{ иначе} \end{cases}$$

Тогава минималният път ще бъде този, в който има най-малко непоследователни фони, т.е. с най-много последователни естествени парчета. Останалите стъпки от процеса на синтез остават същите. Запазваме използването на TD-PSOLA алгоритъма, както и ограничението върху продължителност и фундаментална честота.

За да видим дали има смисъл от използването на няколко мерки, пресмятаме корелацията между техните стойности.

	LS	LSC	MFCC	SegSNR	WSS
LS	1	0.908	0.561	-0.175	0.097
LSC	0.908	1	0.561	-0.236	0.106
MFCC	0.561	0.561	1	-0.111	0.112
SegSNR	-0.175	-0.236	-0.111	1	0.009
WSS	0.097	0.106	0.112	0.009	1

Таблица 3: *Корелация между стойностите на функциите Target в синхронизирани по време фреймове. Фреймовете са извлечени от генерираните от Baseline сигнали.* [6]

[6] За SegSNR, стойностите са отношението Signal-To-Noise в даден фрейм. Отрицателна

Както може да се очаква, корелацията между LS и LSC е значителна. Затова е възможно в експериментите да се получат много близки оптимални параметри за двете мерки. За останалите Target функции корелацията е сравнително ниска, с други думи вероятно бихме получили различни параметри.

Таблица 4 съдържа някои статистики за Target функциите.

	mean	median	std dev
LS	972.472	673.403	1013.58
LSC	26.362	14.43	38.128
MFCC	21.756	15.211	22.337
SegSNR	4.556	4.474	0.88
WSS	40.458	30.872	35.386

Таблица 4: *Стойностите са изчислени върху синхронизирани по време фреймове, извлечени от генерираните от Baseline сигнали от тестовата част на корпуса.*

Тъй като характеристичните функции са независими една от друга и стойностите им са в напълно различни скали и порядъци, е необходимо някакво тяхно нормализиране. В противен случай и получените коефициенти биха били трудни за интерпретация. Ето защо за всяка функция f предварително пресмятаме максимума по модул върху всевъзможните ѝ стойности от корпуса Max_f. От тук нататък за стойност на $f(y, y')$ ще взимаме $\frac{f(y,y')}{Max_f} \in (-1, 1)$.

9.2 Резултати

Проведено е трениране за всяка от споменатите функции Target при $MaxPasses = 3$ #(брой характеристични функции) и всеки от следните параметри:

- E1) характеристични функции $f_1, ..., f_3, g_1, ..., g_3$, $N_k = 100$.
- E2) характеристични функции $f_1, ..., f_3, g_1, ..., g_3$, $N_k = 20$.
- E3) характеристични функции $f_0, ..., f_3, g_1, ..., g_3$, $N_k = 20$.
- E4) характеристични функции $f_0, ..., f_3, g_1, ..., g_3$, $N_k = 20$, редът на обхождане на функциите при търсене е обърнат.

Всички оригинални и съответните им синтетични сигнали, настоящият текст, както и точните стойности на всички изчисления са достъпни на `http:`

корелация тук означава положителна за Noise-To-Signal = $\frac{1}{Signal - To - Noise}$, което всъщност минимизираме.

`//spork-izam.rhcloud.com/samples`. Сигналите са групирани по експеримент, оптимизирана функция Target и част от корпуса, върху която са генерирани (съответно train или eval за тренировъчна и оценъчна част).

Стойността на N_k в E3 е мотивирана от фактът, че E1 и E2 дават идентични (като изходни сигнали) резултати, а голямо N_k увеличава значително времето за оптимизиране.

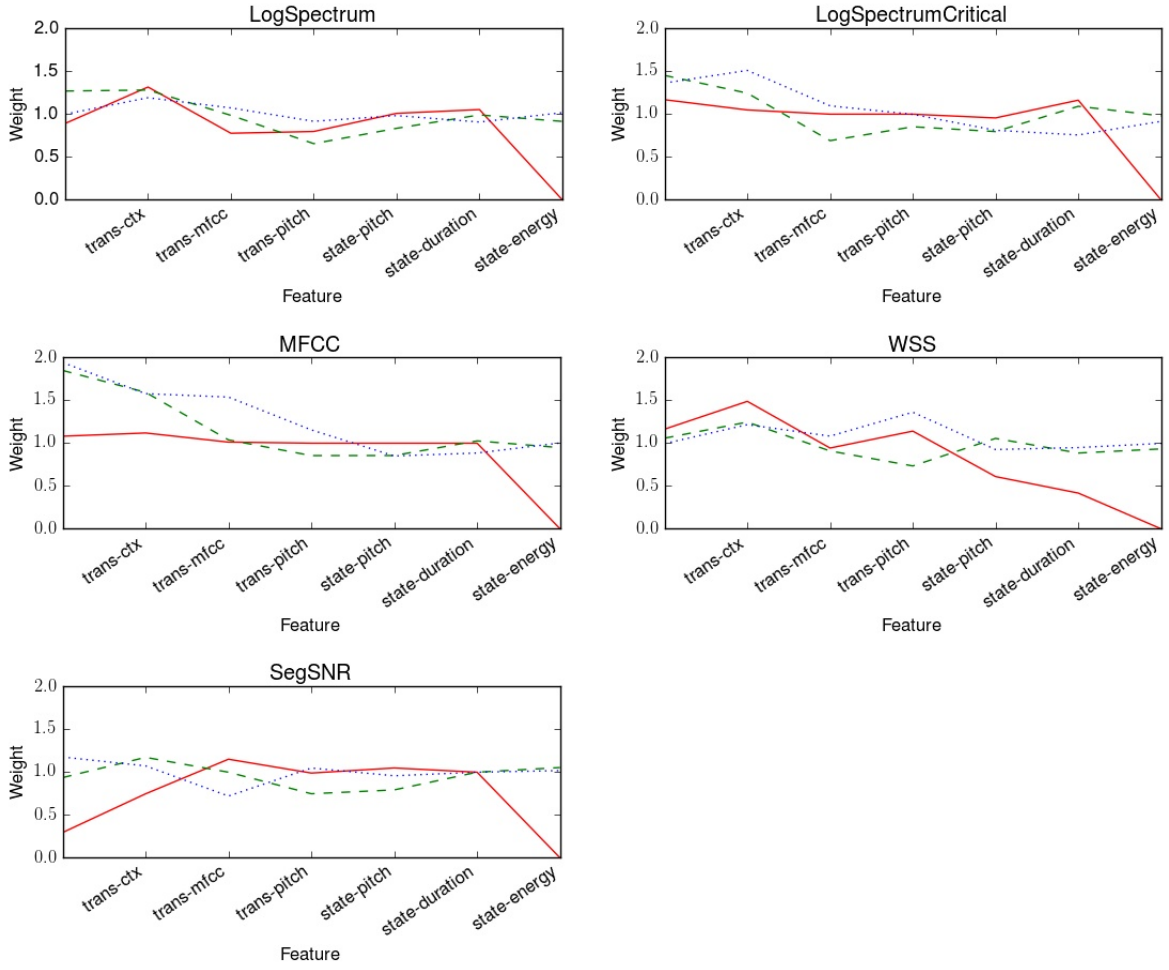


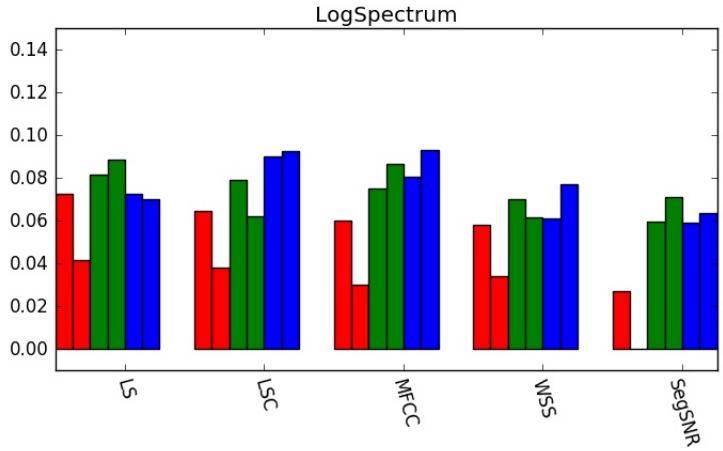
Таблица 5: *Оптимални тегла получени от E2 (червена непрекъсната линия), E3 (зелена прекъсната линия) и E4 (синя точкова линия). Стойност 0 за trans-baseline за E2 отразява факта, че не е включена в множеството на характеристичните функции.*

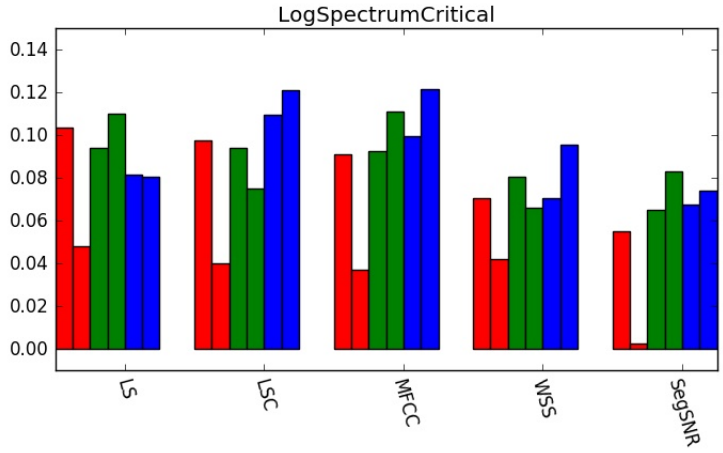
Интересно е да забележим, че коефициентите от експериментите са сравнително близки. Също така, не се наблюдават големи разлики в тег-

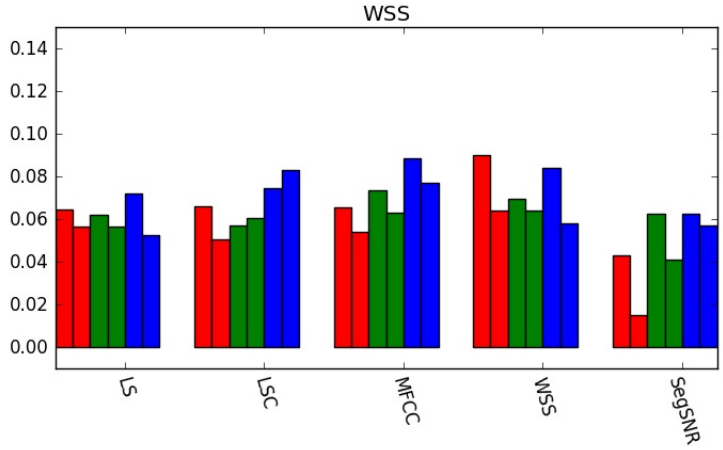
лата на отделните характеристични функции. При неподходящи функции бихме очаквали някои от тях да са със значително по-големи тегла от други. Тъй като търсенето е проведено както в положителна, така и отрицателна посока за всяка характеристична функция, неподходяща такава би получила несъразмерно ниско или дори отрицателно тегло.

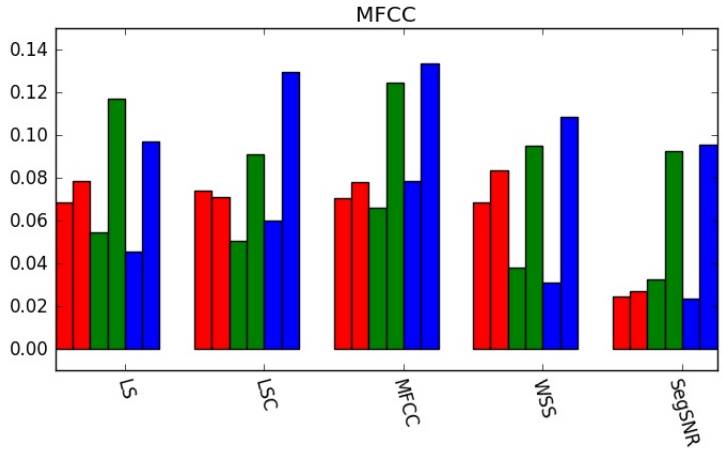
Забележката за различните по скали стойности е валидна и за Target функциите. За да ги представим по обозрим начин, за всяка от тях пресмятаме максималната стойност за съответната функция и част от корпуса при оптимизирани по отделно всяка Target функция. Тъй като максимумът се достига при базата за сравнение, той не е отбелязан. Приложение Б съдържа точните стойности на Target функциите.

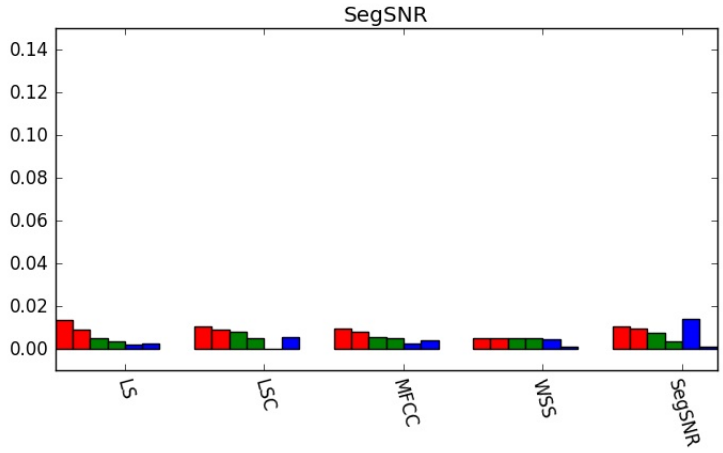
Следващите графики представят резултатите от експериментите. Всяка графика съдържа **подобрението** в стойността на съответната Target функция като част от нейния максимум (ординатата). По абсцисата са изброени оптимизираните Target функции. За всяка функция е пресметнато подобрението за E2 (в червено), E3 (в зелено), E4 (в синьо) последователно за тренировъчната и оценъчна части от корпуса (съответно лява и дясна част).




LogSpectrumCritical
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
LS
LSC
MFCC
WSS
SegSNR


WSS
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
LS
LSC
MFCC
WSS
SegSNR


MFCC
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
LS
LSC
MFCC
WSS
SegSNR



Един от недостатъците на алгоритъма на търсене е очевиден. Тъй като процедурата има "greedy" част, т.е. при всеки избор на посока фиксираме съответната координата, то не можем да гарантираме намирането на глобален минимум. Затова се наблюдават аномалии, при които за дадена Target функция е достигнат неин по-малък локален минимум при оптимизирането на друга такава (например WSS има по-голямо подобрение при минимизирана MFCC). Друг интересен ефект е, че има разлики в достигнатия минимум между E3 и E4, т.е. редът на обхождане на функциите наистина оказва някакво влияние, макар и на пръв поглед малко.

9.3 Възможни подобрения

В тази секция ще опишем накратко някои методи, които биха могли допълнително да подобрят качеството на синтетичната реч.

В други системи често се използват алгоритми, които да "изгладят" прекъсванията при връзките между непоследователни фони. [20] предлагат няколко алгоритъма за изглаждане както във времевия, така и в честотния домейн. Един вариант е линейна интерполация на спектрите на фреймове, например с дължина 2 фундаментални периода, взети от двете страни на границата между два фона. Проблемът тук е при кои граници да се приложи алгоритъмът. Например при краткотрайни експлозивни фони интерполацията би влошила качеството. Като цяло алгоритъмът не се смята за подходящ при беззвучни фони [21], но при ограничено използване би могъл да бъде полезен и за представената система.

Методът на оптималното свързване (optimal coupling [22]) е вариант, който не променя сигнала. Неговата идея е границите на фоните да бъдат "плаващи", т.е. да се изместят така, че да се получи минимално прекъсване в спектрите на съседните фреймове около границата. Като цяло методът е успешно използван на много места, но може да се окаже неподходящ, когато се приложи върху много кратки фони, тъй като би могъл да

"отреже" важна част от едната.

За да се справим с кратките фони, един подход е те да бъдат "прилепени" към техни съседи като така се получи нова фонетична единица (наричана в литературата дифон - diphone). Условното случайно поле е изключително гъвкав модел и може относително лесно да се измени така, че да позволява едновременното използване на фони и дифони. [24] и [23] показват на практика как CRF може да се използва за научаването на много по-общи взаимодействия в подобен тип задачи, в частност и съпоставянето на повече от един етикет на дадена позиция. Такъв по-общ модел би могъл да прецени за подходящо изпускането на даден фон или вмъкването на нов такъв.

Друг аспект, който би могъл да се подобри, е използването на друга Target функция с по-добра корелация с оценката за качеството на реч, дадена от хора. [25] представя няколко варианта за комбиниране на различни мерки (някои от които използвани и в тази дипломна работа) в една композитна такава. [25] показва, че подобна сложна мярка има по-висока корелация с оценката за качество, дадена от хора. Изследването, обаче, е извършено в контекста на подобрение качеството на естествена реч при наличието на изкривяващи сигнала фактори (speech enhancement). Такъв фактор е например компресията на речевия сигнал, но т.к. и тук проблемът за синтез на близък речеви сигнал се разглежда като вид изкривяване на оригинала, изследването остава релевантно. За да се възползваме от подобна сложна мярка, обаче, вероятно имаме нужда от повече данни, тъй като самото научаване параметрите на сложната мярка изисква отделен анализ.

Разбира се, използването на повече данни по принцип води до по-добро качество на изходния сигнал. Ще отбележим, че в повечето системи за трениране се използват много по-големи масиви от данни. Например [26] (конкатенативен синтезатор за турски език) използва общо 20 часа реч от професионален диктор.

10 Заключение

В тази дипломна работа представихме модул за конкатенативен синтез на реч за български език по предварително генерирано прозодично описание. Модулът разчита на пълно автоматично намиране на оптимални тегла за набор от характеристични функции. Показахме как статистическият модел Условно случайно поле е един естествен начин за решаване на тази задача. Проведените експерименти показват, че това е адекватен и лесно приложим метод за конструиране на конкатенативен синтезатор. Тъй като единственият необходим вход за трениране на модела е анотирани звукови записи, модулът може с лекота да се използва и за други езици или гласове.

Въпреки това важно за качеството е съответствието между входни записи и домейна на приложение. Проведените експерименти показват за-

доволително качество с малък обем входни данни, когато има консистентност между произношението на фонетичната база и целевия изход.

Литература

[1] Е. Добрева "Увод в общото езикознание"

[2] J. Allen, M. Sharon Hunnicutt, D. Klatt "From Text To Speech: The MITalk System", Cambridge University Press New York, NY, USA ©1987

[3] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, K. Tokuda "The HMM-based Speech Synthesis System (HTS) Version 2.0", Sixth ISCA Workshop on Speech Synthesis (SSW6), p. 294-299

[4] C. Bickley, K. Stevens, D. Williams "A Framework For Synthesis Of Segments Based On Pseudoarticulatory Parameters", Progress in Speech Synthesis, p. 211-220

[5] A. Hunt, A. Black "Unit selection in a concatenative speech synthesis system using a large speech database", Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on (Volume:1)

[6] J. Lafferty, A. McCallum, and F. C.N. Pereira "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning, p. 282-289

[7] J. M. Hammersley, P. Clifford "Markov Fields On Finite Graphs And Lattices", Unpublished manuscript (1971)

[8] P. Boersma, D. Weenink (2015) "Praat: doing phonetics by computer [Computer program]", Version 6.0.19, retrieved 13 June 2016 from http://www.praat.org

[9] G. David Forney, Jr. "The Viterbi Algorithm", Proceedings of the IEEE (Volume:61 , Issue: 3)

[10] Neli Hateva, Petar Mitankin, Stoyan Mihov "BulPhonC: Bulgarian Speech Corpus for the Development of ASR Technology", Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)

[11] S. S. Stevens, J. Volkmann and E. B. Newman "A Scale For The Measurement Of The Psychological Magnitude Pitch", J. Acoust. Soc. Am. 8, 185 (1937)

[12] E. Moulines, F. Charpentier "Pitch-Synchronous Waveform Processing Techniques For Text-To-Speech Synthesis Using Diphones", Speech Communication Volume 9, Issues 5–6, December 1990, Pages 453-467

[13] Dong-Yan Huang "Prediction of Perceived Sound Quality of Synthetic Speech", Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit And Conference 2011

[14] R. Kubichek "Mel-Cepstral Distance Measure For Objective Speech Quality Assessment", Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on (Volume:1)

[15] L. Muda, M. Begam, I. Elamvazuthi "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques", Journal Of Computing, Volume 2, Issue 3, March 2010

[16] D. Klatt "Prediction Of Perceived Phonetic Distance From Critical—band Spectra", Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82. (Volume:7)

[17] H. Fletcher "Loudness, its Definition, Measurement and Calculation", Jour. Acous. Soc. Amer., October, 1933

[18] H. Fletcher "Auditory Patterns", Rev. Mod. Phys. 12, 47 – Published 1 January 1940

[19] E. Zwicker "Subdivision of the Audible Frequency Range into Critical Bands", J. Acoust. Soc. Am. 33, 248 (1961)

[20] D. T. Chappell, J. H. L. Hansen "Spectral Smoothing For Concatenative Speech Synthesis"

[21] D. T. Chappell, J. H. L. Hansen "A Comparison Of Spectral Smoothing Methods For Segment Concatenation Based Speech Synthesis", Speech Communication Volume 36, Issues 3–4, March 2002, Pages 343–373

[22] A. Conkie, S. Isard "Optimal Coupling Of Diphones", In SSW2-1994, 119-122.

[23] C. Sutton, A. McCallum "An Introduction To Conditional Random Fields For Relational Learning"

[24] C. Sutton, K. Rohanimanesh, A. McCallum "Dynamic Conditional Random Fields: Factorized Probabilistic Models For Labeling And Segmenting Sequence Data", Journal of Machine Learning, 8(Mar):693-723 2007

[25] Y. Hu, P. C. Loizou "Evaluation Of Objective Quality Measures For Speech Enhancement", IEEE Transactions on Audio, Speech, and Language Processing (Volume:16 , Issue: 1)

[26] H. SAK, T. GÜNGÖR, Y. SAFKAN "A Corpus-Based Concatenative Speech Synthesis System For Turkish", Turkish Journal of Electrical Engineering and Computer Sciences 14(2):209-223 - January 2006

A Описание на използваните фонетични етикети

Етикет	Описание
Гласни	
A	ударено ‘а’
a	неударено ‘а’, неударено ‘ъ’
O	ударено ‘о’
o	неударено ‘о’,
U	неударено ‘у’
Y	ударено ‘у’
e	ударено ‘ъ’
I	ударено ‘е’, неударено ‘е’
j	ударено ‘и’, неударено ‘и’
Съгласни	
b	‘б’
v	‘ж’
g	‘г’
d	‘д’
w	‘в’
z	‘з’
0	‘дж
9	‘дз
k	‘к’
l	‘л’
m	‘м’
n	‘н’
p	‘п’
r	‘р’
s	‘с’
t	‘т’
f	‘ф’
h	‘х’
c	‘ц’
4	‘ч’
6	‘ш’
Други	
_	пауза, тишина

Б Получени точни стойности за функциите Target

E?	Metric	baseline	LogSpectrum	LogSpectrumCritical	MFCC	WSS
baseline-test	LogSpectrum	973.581	0	0	0	0
baseline-test	LogSpectrumCritical	26.2292	0	0	0	0
baseline-test	MFCC	21.7693	0	0	0	0
baseline-test	SegSNR	0.0218	0	0	0	0
baseline-test	WSS	41.4439	0	0	0	0
baseline-eval	LogSpectrum	979.237	0	0	0	0
baseline-eval	LogSpectrumCritical	26.7631	0	0	0	0
baseline-eval	MFCC	22.2008	0	0	0	0
baseline-eval	SegSNR	0.0219	0	0	0	0
baseline-eval	WSS	40.6569	0	0	0	0
e2-test	LogSpectrum	910.528	947.261	917.199	902.962	914.96
e2-test	LogSpectrumCritical	23.6711	24.7895	24.3724	23.5078	23.8344
e2-test	MFCC	20.1533	21.2295	20.2803	20.2736	20.2323
e2-test	SegSNR	0.0216	0.0216	0.02172	0.02153	0.02162
e2-test	WSS	38.7113	39.6532	37.7114	38.7584	38.7234
e2-eval	LogSpectrum	945.147	982.781	949.181	941.708	953.384
e2-eval	LogSpectrumCritical	25.6959	26.6899	25.6391	25.4727	25.7659
e2-eval	MFCC	20.6221	21.5955	20.348	20.4532	20.4639
e2-eval	SegSNR	0.0217	0.02169	0.02179	0.0217	0.02172
e2-eval	WSS	38.5911	40.0446	38.0475	38.3542	38.4535
e3-test	LogSpectrum	905.514	915.39	896.507	900.387	894.25
e3-test	LogSpectrumCritical	24.117	24.5282	23.7631	23.8068	23.7629
e3-test	MFCC	20.9443	21.0612	20.6644	20.3266	20.579
e3-test	SegSNR	0.02172	0.02167	0.02165	0.02171	0.02172
e3-test	WSS	38.5573	38.8409	39.0882	38.3976	38.8631
e3-eval	LogSpectrum	922.02	913.135	921.702	897.592	895.892
e3-eval	LogSpectrumCritical	24.994	24.533	24.749	23.7932	23.8146
e3-eval	MFCC	20.0901	20.1444	20.1833	19.4321	19.5975
e3-eval	SegSNR	0.02179	0.02182	0.02179	0.02179	0.02182
e3-eval	WSS	38.0554	38.9892	38.1861	38.0988	38.3538
e4-test	LogSpectrum	886.026	914.279	916.257	903.045	894.91
e4-test	LogSpectrumCritical	23.3491	24.3717	24.455	24.0865	23.6143
e4-test	MFCC	20.4616	21.0967	21.2591	20.7754	20.0601
e4-test	SegSNR	0.02183	0.02173	0.02153	0.02179	0.02178
e4-test	WSS	38.3559	37.9625	38.8501	38.4555	37.7648
e4-eval	LogSpectrum	891.723	907.109	920.259	913.671	891.079
e4-eval	LogSpectrumCritical	23.5213	24.203	24.7754	24.6063	23.512
e4-eval	MFCC	19.3281	19.7951	20.077	20.0494	19.2364
e4-eval	SegSNR	0.02178	0.02187	0.02188	0.02184	0.02181
e4-eval	WSS	37.2705	38.2882	38.3448	38.5201	37.518

Таблица 6: *Колоната Metric обозначава оценената функция, а останалите обозначават оптимизираната функция Target.*