Um Sistema TTS Baseado em Redes Neurais Profundas Usando Parâmetros Síncronos de *Pitch*

Ranniery Maia e Rui Seara

***Resumo*— Em sistemas de síntese de fala baseados em redes neurais profundas (*deep neural networks*, DNN), o treinamento é realizado com vetores de parâmetros acústicos usualmente extraídos do sinal de fala a partir de quadros de tamanhos fixos. Este artigo apresenta formas de usar parâmetros acústicos obtidos síncronamente com o *pitch* em tais sistemas, com o objetivo de melhorar a qualidade do sinal sintético. Resultados experimentais mostram que o uso de atributos lingúisticos obtidos de quadros de tamanhos fixos, juntamente com parâmetros acústicos extraídos de forma síncrona com o *pitch*, produzem melhores resultados em termos de medidas objetivas de qualidade.**

***Palavras-Chave*— *Deep learning*, redes neurais profundas, síntese de fala.**

***Abstract*— In speech synthesis systems based on deep neural networks (DNN), training is usually conducted by using acoustic feature vectors extracted from the speech signal at a fixed frame rate. This paper presents some approaches to use pitch-sychronous acoustic features in speech synthesizers based on DNN, with the goal to improve synthetic speech quality. Experimental results show that the use of frame-based linguistic features, along with pitch-synchronously extracted acoustic parameters, produce better results in terms of objective quality measures.**

***Keywords*— Deep learning, deep neural networks, speech synthesis.**

I. Introdução

A síntese de fala através de modelos estatísticos e paramétricos (*statistical parametric speech synthesis*, SPSS) [1] apresenta certas vantagens quando comparada com as técnicas de síntese de fala que usam busca e concatenação de unidades fonéticas [2]. Dentre tais vantagens, podemos citar a capacidade de transformar a fala sintética em uma fala com estilo diferente (por exemplo, emoção, gênero) através da manipulação de parâmetros estatísticos com ou sem o uso de uma pequena base de dados. Além disso, a técnica de SPSS possibilita a construção automática de sistemas texto-fala (*text-to-speech*, TTS) com o mínimo de procedimentos heurísticos.

Recentemente, avanços importantes em SPSS vêm sendo alcançados com o uso de *deep learning* [3]–[6]. A vasta e recente literatura na área mostra que a aplicação de *deep learning* não somente melhora a qualidade da síntese através de uma representação mais precisa dos parâmetros acústicos usados para gerar o sinal de fala, mas também com a aplicação de métodos inovadores para modelagem acústica em sistemas TTS. Dentre tais métodos, podemos citar a extração de parâmetros usando redes neurais profundas auto-decodificadoras [7] e o uso de atributos extraídos de camadas *bottlenecks* [8]. Por fim, alguns modelos baseados em redes neurais complexas podem gerar amostras do sinal de fala diretamente no domínio do tempo, alcançando qualidade superior, quando comparados aos sistemas TTS que usam busca e concatenação de unidades [9].

Dentre os tipos de redes neurais mais comuns aplicadas a SPSS até o momento, tanto as redes neurais artificiais profundas na configuração *feed-forward* (*deep neural networks*, DNN) quanto as arquiteturas que levam em conta memória, tais como as redes neurais recorrentes (*recurrent neural networks*, RNN), possuem vantagens quando comparadas aos modelos semi-Markov ocultos (*hidden semi-Markov models*, HSMM) [10]. Dentre elas, podemos citar o maior poder das redes neurais em representar dados vetoriais cujos elementos não são estatisticamente independentes ou possuam alta dimensionalidade [11]. Além disso, as DNN (bem como as RNN) também proporcionam melhor representação de alguns parâmetros acústicos típicos usados em sistemas TTS baseados em HSMM (HSMM-TTS), devido principalmente à superação de uma das limitações conhecidas das árvores de decisão, no caso a fragmentação ou quantização dos dados.

Uma característica importante do uso de redes neurais para síntese (quando comparadas aos HSMM) é o fato de não haver necessidade de os parâmetros de entrada e saída no treinamento serem extraídos de quadros de tamanho fixo, visto que as redes neurais não operam no estilo máquina de estados, como é o caso dos HSMM. Com base nessa característica, este artigo apresenta um estudo do uso de parâmetros síncronos com o *pitch*, bem como o seu impacto em termos de qualidade do sinal de fala sintética. A vantagem da análise do sinal de fala de forma síncrona está no fato de os parâmetros dos modelos digitais de fala serem diretamente estimados a cada período de *pitch*. Um típico exemplo seria a resposta de fase contínua do sinal de fala, que pode ser extraída nos momentos de fechamento da glote (*glottal closure instants*, GCI) [12]. Outro exemplo seria a envoltória espectral suave de tempo curto obtida sem o efeito harmônico da frequência fundamental, que pode ser extraída usando marcas de *pitch* [13].

Nas demais seções, este artigo está organizado da seguinte forma. Na Seção II, apresentamos uma breve descrição da técnica de síntese de fala baseada em DNN. A Seção III discute algumas formas de implementar parâmetros síncronos com o *pitch* em sistemas DNN-TTS. A Seção IV trata dos aspectos relacionados à implementação de DNN-TTS para o idioma Português falado no Brasil. Na Seção V, são mostrados

Ranniery Maia e Rui Seara, LINSE - Laboratório de Circuitos e Processamento de Sinais, Departamento de Engenharia Elétrica e Eletrônica, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil (e-mails: rmaia@linse.ufsc.br, seara@linse.ufsc.br).

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo no. 165836/2015-6.

alguns experimentos de síntese de fala usando parâmetros síncronos com o *pitch*; e na Seção VI estão as conclusões deste estudo.

II. Síntese de Fala por Redes Neurais Profundas

Apesar de haver várias formas na qual um sistema TTS pode ser implementado usando *deep learning* (veja [14] para um estudo mais detalhado), neste trabalho utilizamos o sistema ilustrado na Fig. 1, que é semelhante ao descrito em [4]. Nesse caso, o objetivo da DNN é aproximar uma função que mapeie um conjunto de vetores linguísticos de entrada $\mathcal{L}$ em um conjunto de vetores de parâmetros acústicos de saída $\mathcal{O}$, isto é, $\mathcal{O} = f(\mathcal{L})$. O conjunto de vetores de saída da rede $\mathcal{O} = \{\boldsymbol{o}_0, \ldots, \boldsymbol{o}_{T-1}\}$ contêm os parâmetros acústicos extraídos dos sinais de fala concatenados aos correspondentes coeficientes dinâmicos,

$$\boldsymbol{o}_t = \left[\boldsymbol{y}_t^\top \quad \Delta^{(1)}\boldsymbol{y}_t^\top \quad \cdots \quad \Delta^{(M)}\boldsymbol{y}_t^\top\right]^\top \tag{1}$$

onde $\boldsymbol{y}_t$ é um vetor contendo parâmetros que podem ser usados para reconstruir o sinal de fala usando modelos paramétricos de síntese [15], $[\cdot]^\top$ indica transposição de matriz, T é o números de vetores usados no treinamento, e M é a ordem de coeficientes dinâmicos. Os vetores de entrada da rede $\mathcal{L} = \{\boldsymbol{l}_0, \ldots, \boldsymbol{l}_{T-1}\}$, por sua vez, contêm atributos linguísticos gerados a partir da base de textos, em que cada vetor de entrada $\boldsymbol{l}_t$ contém três subvetores, isto é,

$$\boldsymbol{l}_t = \left[\boldsymbol{l}_t^{(b)\top} \quad \boldsymbol{l}_t^{(n)\top} \quad \boldsymbol{l}_t^{(d)\top}\right]^\top \tag{2}$$

com $\boldsymbol{l}_t^{(b)}$, $\boldsymbol{l}_t^{(n)}$ e $\boldsymbol{l}_t^{(d)}$ representando, respectivamente, os atributos binários, numéricos e de duração.

Durante a síntese, uma sequência de atributos linguísticos é gerada do texto a ser sintetizado $\tilde{\mathcal{L}} = \left\{\tilde{\boldsymbol{l}}_0, \ldots, \tilde{\boldsymbol{l}}_{N-1}\right\}$, onde N é o número de quadros da sentença a ser sintetizada. A variável $\tilde{\mathcal{L}}$ é então passada pela DNN, de forma que na saída tenhamos uma sequência correspondente de parâmetros acústicos $\tilde{\mathcal{O}} = f\left(\tilde{\mathcal{L}}\right) = \{\tilde{\boldsymbol{o}}_0, \ldots, \tilde{\boldsymbol{o}}_{N-1}\}$. Como $\tilde{\mathcal{O}}$ é nesse caso assumido ser uma sequência de médias, a sequência acústica final é obtida ao aplicar um dos algoritmos dados em [16] sobre $\tilde{\mathcal{O}}$, juntamente com a variância global, resultando na sequência final de parâmetros acústicos $\tilde{\mathcal{Y}} = \{\tilde{\boldsymbol{y}}_0, \ldots, \tilde{\boldsymbol{y}}_{N-1}\}$. Por fim, a forma de onda é produzida com $\tilde{\mathcal{Y}}$ ao assumir um modelo paramétrico digital de produção do sinal de fala [15].

III. DNN-TTS com Parâmetros Síncronos

Uma das vantagens de realizar análise síncrona com o *pitch*, para sintetizadores da família SPSS é obter uma estimativa da envoltória espectral de tempo curto sem o efeito da frequência fundamental (F_0) [17], [18]. Além disso, pode-se também usar nesse caso a resposta de fase do sinal, pois, quando a análise é feita nos GCI, facilita-se a obtenção dos coeficientes que representam a resposta de fase, porque em tais instantes o componente de fase linear da resposta de fase é teoricamente zero [19].

Vale lembrar aqui que a ideia é usar somente parâmetros síncronos durante o treinamentos dos sistemas DNN-TTS. Para que a geração da forma de onda sintética seja feita de forma

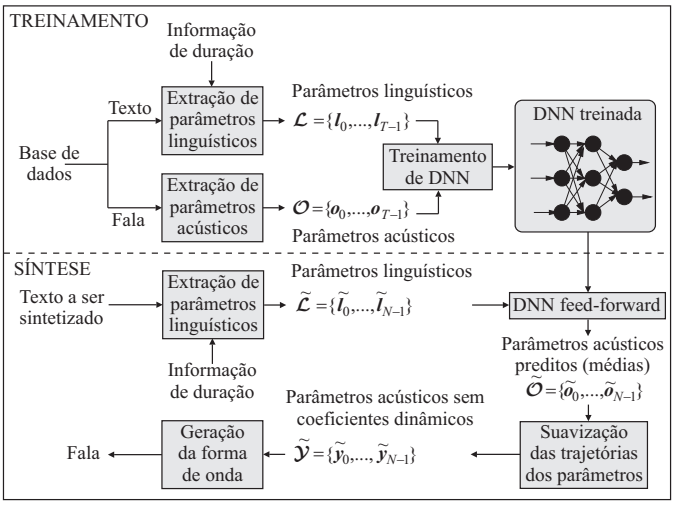

Fig. 1. Sistema TTS usando DNN, com as etapas de treinamento e síntese.

síncrona, é necessário que haja o conhecimento das marcas de *pitch* de antemão. Isso pode ser feito com o uso de um modelo de prosódia externo, estando esse fora do escopo deste trabalho de pesquisa.

Portanto, para treinar uma DNN-TTS com parâmetros síncronos, é necessário: 1) que os parâmetros acústicos sejam extraídos de forma síncrona com o *pitch*; 2) que o vetor $\boldsymbol{l}_t^{(d)}$ represente a informação de duração dos quadros de tamanhos não-uniformes que começam e terminam de acordo com as GCI.

A. Extração de Parâmetros Síncronos com o Pitch

Visando utilizar a informação de envoltória espectral suave, sem o efeito da F_0, bem como informação de fase, o modelo de análise e síntese de fala baseado no cepstro complexo [20] pode ser usado. Inicialmente, GCI $\{p_0, \ldots, p_{Z-1}\}$ são detectados do sinal de fala [21], no qual Z é o número de GCI. Em seguida, a resposta em frequência do sinal de fala $s(n)$ no instante p_z é obtida através de

$$S_z\left(e^{\jmath\omega}\right) = \sum_{n=p_{z-1}}^{p_{z+1}} s(n) w\left(n - p_{z-1}\right) e^{-\jmath\omega n} \tag{3}$$

onde $w(n)$ é uma janela usada para selecionar o sinal de fala entre p_{z-1} e p_{z+1}. Finalmente, o cepstro complexo é computado usando

$$\hat{h}_z(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left\{\log\left|S_z\left(e^{\jmath\omega}\right)\right| + \jmath\theta_z(\omega)\right\} e^{\jmath\omega n} d\omega \tag{4}$$

onde $\left|S_z\left(e^{\jmath\omega}\right)\right|$ e $\theta_z(\omega)$ são, respectivamente, as respostas de amplitude e de fase contínua de $s(n)$ no instante p_z. Para fins de modelagem acústica em SPSS, o cepstro complexo é decomposto em seus componentes de fase mínima e de fase residual, conforme discutido em [20]. Assim, o cepstro de fase mínima no instante p_z é dado por

$$\hat{h}_z^{(m)}(n) = \begin{cases} \hat{h}_z(n), & n = 0 \\ \hat{h}_z(n) + \hat{h}_z(-n), & 1 \leq n \leq C \end{cases} \tag{5}$$

enquanto os parâmetros de fase são

$$\phi_z(n) = \hat{h}_z(-n-1), \quad 0 \leq n < C. \tag{6}$$

Note que, mesmo com parâmetros acústicos extraídos de forma não-periódica, o mapeamento entrada-saída da DNN durante o treinamento pode ser feito de forma periódica, pois as durações dos estados dos HSMM geradas pelo alinhamento fonético são em geral fornecidas por quadros cujos tamanhos são múltiplos de 5 ou 10 ms. Para isso, os parâmetros acústicos síncronos com o *pitch*, extraídos em cada um dos instantes $\{p_0, \ldots, p_{Z-1}\}$, podem ser transformados da seguinte forma:

$$\boldsymbol{y}_t = \{\boldsymbol{y}_z \mid p_z \leq tK < p_{z+1}\}, \;\; t = 0, \ldots, N-1 \tag{7}$$

onde K é a duração de um quadro em sistemas HSMM-TTS. Perceba que (7) simplesmente repete os vetores do instante de *pitch* anterior até o próximo GCI.

B. Informação de Duração na Entrada da Rede

Em sistemas estado-da-arte DNN-TTS, a informação de duração é usualmente codificada a cada período de duração fixa, ao se considerar a posição do quadro de duração fixa dentro do estado do HSMM e do fonema[1]. Como a ideia aqui é substituir, durante o treinamento, as janelas periódicas de tamanho fixo por segmentos que comecem e terminem de acordo com os GCI, podemos determinar $\boldsymbol{l}_t^{(d)}$ da seguinte forma:

$$\boldsymbol{l}_t^{(d)} = \begin{bmatrix} (p_z - t_{s_i}) \,/\, (t_{s_f} - t_{s_i}) \\ (t_{s_f} - p_z) \,/\, (t_{s_f} - t_{s_i}) \\ (p_z - t_{p_i}) \,/\, (t_{p_f} - t_{p_i}) \\ (t_{p_f} - p_z) \,/\, (t_{p_f} - t_{p_i}) \end{bmatrix} \tag{8}$$

onde os tempos t_{s_i}, t_{s_f}, t_{p_i}, t_{p_f} são, respectivamente, os tempos de início e fim do estado do HSMM e do fonema, nos quais p_z está inserido.

IV. DNN-TTS para o Idioma Português

Para a aplicação da estrutura DNN-TTS (apresentada na Seção II) a um idioma específico, é necessário somente que seja definida a informação linguística a ser representada pelos vetores de entrada da rede neural, isto é, $\mathcal{L} = \{\boldsymbol{l}_0, \ldots, \boldsymbol{l}_{T-1}\}$. Assumindo que já existe um sistema HSMM-TTS em funcionamento para o referido idioma, a tarefa torna-se ainda mais simples, consistindo em selecionar perguntas do conjunto de questões usado no processo de *context clustering* durante o treinamento do sistema HSMM-TTS. Neste trabalho, as informações que compõem os vetores $\boldsymbol{l}_t^{(b)}$ e $\boldsymbol{l}_t^{(n)}$ são extraídas de [22]. A seguir, fazemos um resumo dessas informações.

A. Atributos Binários

Para o caso do idioma Português falado no Brasil, a informação binária $\boldsymbol{l}_t^{(b)}$ pode consistir em grupos fonéticos, indicadores silábicos e classificadores semânticos e/ou morfológicos, tais como

- tipo de fonema: sonoro, surdo, fricativo.
- tonicidade silábica: silaba tônica ou átona;
- classificação das palavras: substantivo, artigo, palavras de conteúdo.
- tipos de frases quanto à entonação: interrogativa, afirmativa.

B. Atributos Numéricos

Os parâmetros de entrada numéricos são aqueles relacionados a quantidades e distâncias, que de acordo com [22] são:

- número de fones na sílaba;
- número de sílabas na palavra, frase e sentença;
- posição da sílaba na palavra, frase e sentença;
- número de sílabas tônicas na frase antes e depois da sílaba corrente;
- distância da sílaba corrente até a próxima sílaba tônica antes e depois, na frase corrente;
- número de palavras na frase e sentença;
- posição da palavra corrente na frase corrente e sentença;
- distância entre a palavra corrente e a próxima palavra de conteúdo antes e depois, na frase corrente;
- posição da frase corrente na sentença;
- número de sílabas e palavras na frase anterior, posterior e corrente;
- número de sílabas, palavras e frases na sentença.

Os atributos numéricos são geralmente normalizados entre 0 e 1. Os valores máximos e mínimos de cada atributo podem ser obtidos da base a ser usada no treinamento do DNN-TTS.

V. Experimentos

A. Preparação da Base de Dados Usada

Para implementar o DNN-TTS proposto para o idioma Português falado no Brasil, a base de dados *Constituicao 1.0*, fornecida pelo grupo do projeto *FalaBrasil* [23] é utilizada. Essa base consiste em nove horas de áudio amostrado a 22,05 kHz, gravado em um ambiente controlado, por um locutor masculino. Além do áudio, o texto correspondente também é disponibilizado.

Para a obtenção de etiquetas fonéticas e de contexto, uma ferramenta de preparação de base de dados para a síntese (baseada no método descrito em [24]) é implementada. O processo consiste na criação de etiquetas fonéticas iniciais usando o módulo de conversão grafema-fonema de um processador de textos, seguido por procedimentos de alinhamento da base de dados. Para isso, um dicionário específico é usado, em que para cada palavra a ser alinhada existe uma versão com ou sem uma pausa logo a seguir. Dessa forma, consegue-se não só alinhar a base, mas também detectar as pausas. Por fim, as novas etiquetas fonéticas com informação de início e fim, e pausas nos lugares corretos, são usadas para produzir as etiquetas de contexto usadas em sistemas HSMM-TTS. Do total da base de dados, selecionamos uma hora, coletando as sentenças que obtêm os maiores escores de log-verossimilhança durante o procedimento final de alinhamento.

[1] As informações de duração de estado HSMM e de fonema são fornecidas pelo alinhamento forçado usando um sistema HSMM-TTS equivalente.

B. Entradas e Saídas das DNN

Dois conjuntos de entrada são preparados, isto é,

- $\mathcal{L}_1 = \{\boldsymbol{l}_0, \ldots, \boldsymbol{l}_{T-1}\}$: conjunto de vetores extraídos a cada quadro de 5 ms, em que cada subvetor duração $\boldsymbol{l}_t^{(d)}$ contém informação das posições do correspondente quadro dentro do estado do HSMM e do fonema;
- $\mathcal{L}_2 = \{\boldsymbol{l}_0, \ldots, \boldsymbol{l}_{z-1}\}$: conjunto de vetores síncronos com os GCI, em que cada subvetor duração $\boldsymbol{l}_t^{(d)}$ está conforme descrito na Seção III-B.

Os atributos linguísticos são obtidos usando uma versão para o idioma Português do *Festival Speech Synthesis System* [25], enquanto o alinhamento fonético, ao nível dos estados dos HSMM, são gerados por um sistema HSMM-TTS treinado com $\mathcal{O}_1$ descrito a seguir. Ao final, cada vetor $\boldsymbol{l}_t$ possuirá 888 elementos.

Os parâmetros acústicos são extraídos segundo o modelo de análise e síntese de fala usando o cepstro complexo [20], discutido na Secção III. Detecção de GCI e extração de F_0 são feitos usando a ferramenta *Reaper* [26]. A frequência F_0 é extraída dos sinais de fala a cada 5 ms. Cada vetor de parâmetros $\boldsymbol{o}_t$ é composto de 45 coeficientes mel-cepstrais, $\log(F_0)$, 39 coeficientes de fase, e *flag* de decisão sonoro/surdo. Os coeficientes de velocidade e aceleração também são concatenados, ou seja $M = 2$, exceto pelo *flag* sonoro/surdo, de forma que a dimensão de $\boldsymbol{o}_t$ é 256. Ao final, dois conjuntos de parâmetros são obtidos. Assim,

- $\mathcal{O}_1 = \{\boldsymbol{o}_0, \ldots, \boldsymbol{o}_{T-1}\}$: conjunto de vetores periódicos a cada 5 ms, obtidos a partir de $\mathcal{O}_2$ através de (7);
- $\mathcal{O}_2 = \{\boldsymbol{o}_0, \ldots, \boldsymbol{o}_{z-1}\}$: conjunto de vetores síncronos com o *pitch*.

C. Treinamento dos Sistemas

Inicialmente, um sistema HSMM-TTS, que denominamos modelo $\boldsymbol{\Lambda}_1$, é treinado considerando as etiquetas de contexto e fonéticas extraídas da base de dados, juntamente com $\mathcal{O}_1$. Em seguida, os HSMM treinados são então usados para alinhar a base de dados ao nível de cada estado dos HSMM. Usando essas informações de alinhamento, $\mathcal{L}_1$ e $\mathcal{L}_2$ são produzidos. Com isso, dois DNN-TTS são treinados. Dessa forma,

1) $\boldsymbol{\Lambda}_2$: treinado com $\{\mathcal{L}_1, \mathcal{O}_1\}$;
2) $\boldsymbol{\Lambda}_3$: treinado com $\{\mathcal{L}_2, \mathcal{O}_2\}$.

O treinamento ocorre usando a ferramenta *TensorFlow* [27] em um *MacOs* Versão 10.13.3, com processador duplo *3,06 GHz Intel Core i3* e 8 GB de memória. O tempo de treinamento de cada DNN-TTS dura em torno de 15 horas, enquanto o HSMM-TTS despende 3 horas para ser treinado. Os passos de treinamento do HSMM-TTS são aqueles sugeridos pelas ferramentas disponíveis em [28]. Para o nosso caso, no entanto, o peso dos parâmetros da fase $\phi(n)$ na probabilidade de emissão dos estados dos HSMM é ajustado para zero.

Todas as DNN têm 5 camadas ocultas com 1024 neurônios cada, função de ativação sigmoide e camada de saída linear. As dimensões dos vetores de entrada e saída são 888 e 256, respectivamente. Para o treinamento das DNN, é utilizado o método de otimização Adam [27] e taxa de aprendizagem de 10^{-3}. O *batch size* escolhido é de 256 como um compromisso entre consumo de memória, precisão e tempo de treinamento. O número de épocas máximo é ajustado para 50 e não é utilizado um conjunto de dados para validação.

TABELA I

RESULTADOS DE AVALIAÇÃO OBJETIVA COM 20 SENTENÇAS DE TESTE (APROXIMADAMENTE 10 MINUTOS). AS DURAÇÕES SÃO OBTIDAS ATRAVÉS DE ALINHAMENTO FONÉTICO USANDO O MODELO $\boldsymbol{\Lambda}_1$

	Situação	MCD (dB)	RMS F_0	% VUV errados	RMS fase
1	HSMM: $\tilde{\mathcal{O}}_1 \leftarrow \left\{\tilde{\mathcal{L}}_1, \boldsymbol{\Lambda}_1\right\}$	5,39	**17,19**	3,94	3,44
2	DNN: $\tilde{\mathcal{O}}_2 \leftarrow \left\{\tilde{\mathcal{L}}_1, \boldsymbol{\Lambda}_2\right\}$	**5,18**	17,82	**3,39**	**3,37**
3	DNN: $\tilde{\mathcal{O}}_3 \leftarrow \left\{\tilde{\mathcal{L}}_1, \boldsymbol{\Lambda}_3\right\}$	5,35	18,31	3,87	3,38
4	DNN: $\tilde{\mathcal{O}}_4 \leftarrow \left\{\tilde{\mathcal{L}}_2, \boldsymbol{\Lambda}_2\right\}$	6,86	21,88	17,30	3,32
5	DNN: $\tilde{\mathcal{O}}_5 \leftarrow \left\{\tilde{\mathcal{L}}_2, \boldsymbol{\Lambda}_3\right\}$	5,30	18,56	3,64	3,21

D. Avaliação Objetiva

Para avaliar os sistemas treinados, são usadas as seguintes medidas de distorção entre parâmetros acústicos naturais e gerados pelos sistemas: distância mel-cepstral (MCD) em decibeis [29]; valor eficaz (*root mean square*, RMS) da F_0 em quadros sonoros de 5 ms; percentual de erros de classificação sonoro/surdo (VUV) dos quadros de 5 ms; e RMS dos parâmetros de fase. Os resultados para todas as situações testadas estão mostrados na Tabela I. Pode-se observar que a melhor situação em termos de menor distorção espectral e erros sonoros/surdos ocorre quando $\tilde{\mathcal{O}}_2 \leftarrow \left\{\tilde{\mathcal{L}}_1, \boldsymbol{\Lambda}_2\right\}$, apesar de o sistema HSMM-TTS ter obtido menor distorção de F_0, o que é esperado quando se usa DNN, devido à prática da interpolação para obtenção de valores nas regiões surdas da F_0 [4]. Já a menor distorção de fase ocorre na situação $\tilde{\mathcal{O}}_5 \leftarrow \left\{\tilde{\mathcal{L}}_2, \boldsymbol{\Lambda}_3\right\}$, que utiliza entrada síncrona com o *pitch*. Vale lembrar que as situações 4 e 5 necessitam de um modelo externo de prosódia, pois para gerar vetores com atributos linguísticos de forma síncrona para um texto qualquer no momento da síntese, é necessário que se tenha conhecimento das marcas de *pitch* de antemão antes da geração de parâmetros acústicos. Portanto, se consideramos somente os casos práticos (1, 2 e 3), $\boldsymbol{\Lambda}_2$ também obtém a menor distorção de fase. A Fig. 2 mostra, para uma dada frase de teste, as trajetórias naturais do quarto coeficiente mel-cepstral e F_0, junto com as correspondentes versões sintéticas produzidas nas situações 1, 2 e 3. Percebe-se que, no caso do cepstro, o melhor modelo é $\boldsymbol{\Lambda}_2$, ao passo que para o caso da F_0, $\boldsymbol{\Lambda}_1$ é melhor.

VI. CONCLUSÕES

Neste artigo, foi apresentado um estudo de aplicação de parâmetros síncronos com o *pitch* a sistemas DNN-TTS, com o objetivo de melhorar a qualidade do sinal de fala. A motivação está baseada nos seguintes fatos: 1) análise síncrona com o *pitch* proporciona estimativa da envoltória espectral suave de tempo curto e resposta de fase do sinal de fala, que melhoram o sinal produzido por sistemas do tipo SPSS; 2) as DNN não requerem que os parâmetros de entrada e saída sejam extraídos de quadros de tamanho fixo. Medidas objetivas de qualidade mostraram que, dentre as estruturas testadas, o

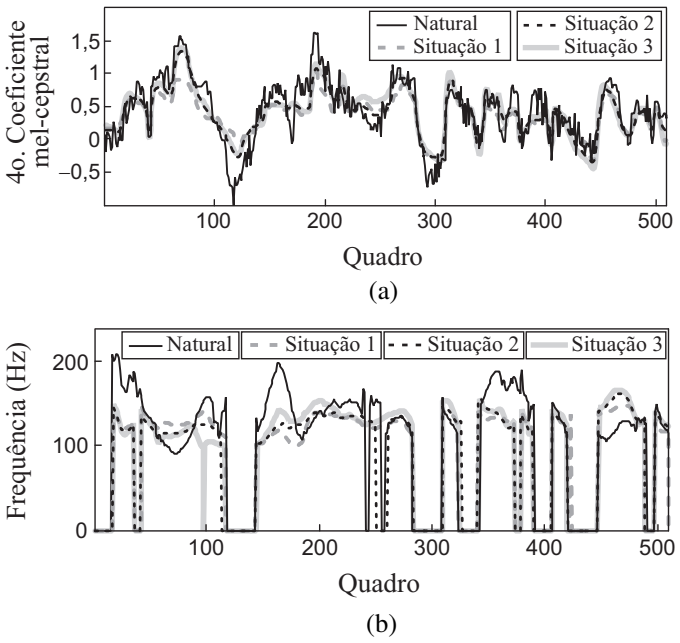

Fig. 2. Trajetórias natural e sintéticas, produzidas pelos modelos $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$ e $\mathbf{\Lambda}_3$, com entrada $\hat{\mathcal{L}}_1$. (a) Quarto coeficiente mel-cepstral. (b) Frequência F_0.

melhor resultado foi obtido quando os parâmetros acústicos extraídos de forma síncrona com os GCI foram usados com mapeamento entrada-saída para cada quadro de tamanho fixo, fornecido pelo alinhamento fonético. Isso decorre por dois fatores: a) a quantidade de dados para treinamento é maior quando quadros de 5 ms são utilizados; b) a entrada da rede é gerada a partir de alinhamento fonético produzido por HSMM. No futuro, pretendemos investigar métodos que eliminem a necessidade do uso de um sistema HSMM-TTS equivalente, o qual tem tornado-se até então requerido para obtenção de parâmetros linguísticos de entrada.

AGRADECIMENTOS

Os autores gostariam de agradecer ao grupo do projeto FalaBrasil, pela disponibilidade da base de dados *Constituicao 1.0* on-line, e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERÊNCIAS

[1] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Commun.*, vol. 51, no. 11, pp. 1039–1064, Nov. 2009.

[2] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, USA, May 1996, pp. 373–376.

[3] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, Canada, May 2013, pp. 7825–7829.

[4] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, Canada, May 2013, pp. 7962–7966.

[5] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. ISCA Interspeech*, Singapore, Sept. 2014, pp. 1964–1968.

[6] H. Zen, Y. Agiosmyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak, "Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices," in *Proc. ISCA Interspeech*, San Francisco, USA, Sept. 2016, pp. 2273–2277.

[7] S. Takaki and J. Yamagishi, "A deep auto-encoder based low-dimensional feature extraction from FFT spectral envelopes for statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Apr. 2016, pp. 5535–5539.

[8] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, Apr. 2015, pp. 4460–4464.

[9] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A.r Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, [Online]. Available: `https://arxiv.org/abs/1609.03499`.

[10] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Hidden semi-Markov model based speech synthesis," in *Proc. ISCA Interspeech*, Jeju, Korea, Oct. 2004, pp. 1393–1396.

[11] Q. Hu, Z. Wu, K. Richmond, J. Yamagishi, Y. Stylianou, and R. Maia, "Fusion of multiple parameterisations for DNN-based sinusoidal speech synthesis with multi-task learning," in *Proc. ISCA Interspeech*, Dresden, Germany, Sept. 2015, pp. 854–858.

[12] T. Drugman, B. Bozkurt, and T. Dutoit, "A comparative study of glottal source estimation techniques," *Comput. Speech Language*, vol. 26, no. 1, pp. 20–34, Jan. 2012.

[13] J. E. Markel and A. H. Gray, *Linear Prediction of Speech*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[14] Z.-H. Ling, S. Kang, H. Zen, A. W. Senior, M. Schuster, X. Qian, H. M. Meng, and L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 35–52, May 2015.

[15] J. R. Deller, Jr., J. H. L. Hansen, and J. G. Proaks, *Discrete-Time Processing of Speech Signals*, IEEE Press Classic Reissue, New York, NY, USA, 2000.

[16] K. Tokuda, T. Kobayashi, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, June 2000, pp. 1315–1318.

[17] K. K. Paliwal and P. V. S. Rao, "A modified autocorrelation method of linear prediction for pitch-synchronous analysis of voiced speech," *Signal Process.*, vol. 3, no. 2, pp. 181–185, Apr. 1981.

[18] S. Chandra and W. Lin, "Experimental comparison between stationary and nonstationary formulations of linear prediction applied to voiced speech analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 22, no. 6, pp. 403–415, Dec. 1974.

[19] T. Quatieri, *Discrete-time Speech Signal Processing: Principles and Practice*, Prentice Hall Press, Upper Saddle River, NJ, USA, 2001.

[20] R. Maia, M. Akamine, and M.J.F. Gales, "Complex cepstrum for statistical parametric speech synthesis," *Speech Commun.*, vol. 5, no. 55, pp. 606–618, June 2013.

[21] T. Drugman, M. Thomas, J. Gudnason, P. Naylor, and T. Dutoit, "Detection of glottal closure instants from speech signals: a quantitative review," *IEEE Trans. Audio, Speech, Language Process.*, vol. 3, no. 20, pp. 994–1006, Mar. 2012.

[22] R. Maia, H. Zen, K. Tokuda, T. Kitamura, and F. G. Resende Jr., "An HMM-based Brazilian Portuguese synthesizer and its characteristics," *IEEE J. Commun. Inf. Systems*, vol. 21, no. 2, pp. 58–71, Aug. 2006.

[23] "FalaBrasil: reconhecimento de voz para o português brasileiro," [Online]. Available: `http://www.laps.ufpa.br/falabrasil/`. Accessed Mar. 2017.

[24] R. Maia, J. Ni, S. Sakai, T. Toda, K. Tokuda, T. Shimizu, and S. Nakamura, "The NICT/ATR speech synthesis system for the Blizzard Challenge 2008," [Online]. Available: `http://festvox.org/blizzard/blizzard2008.html`. Accessed Apr. 2017.

[25] "The Festival Speech Synthesis System," [Online]. Available: `http://www.cstr.ed.ac.uk/projects/festival`. Accessed Apr. 2017.

[26] "REAPER: Robust Epoch And Pitch EstimatoR," [Online]. Available: `https://github.com/google/REAPER`. Accessed Mar. 2017.

[27] "TensorFlow: An open-source software library for Machine Intelligence," [Online]. Available: `https://www.tensorflow.org`. Accessed Apr. 2017.

[28] "The HMM-based Speech Ssynthesis Toolkit," [Online]. Available: `http://hts.nitech.ac.jp`. Accessed Apr. 2017.

[29] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.