多重周波数解像度のSTFTスペクトルを用いた敵対的DNN音声合成*

◎齋藤 佑樹，高道 慎之介，猿渡 洋（東大院・情報理工）

1 はじめに

統計的パラメトリック音声合成において，STRAIGHT [1] や WORLD [2] をはじめとした高品質なボコーダは重要な役割を果たしてきた．しかし，Deep Neural Network (DNN) に基づく音声合成 [3] の表現力が高くなるにつれ，ボコーダ処理が合成音声品質の低下の主要因になりつつある．

この音質劣化を避けるために，近年ではボコーダを用いない統計的パラメトリック音声合成の手法が提案されている．本稿で対象とするShort-Term Fourier Transform (STFT) スペクトルを用いた音声合成 [4] では，DNN音響モデルはテキスト特徴量から合成音声の対数振幅スペクトルを生成する．その後，Griffinらのアルゴリズム [5] を用いて，生成された対数振幅スペクトルから位相情報を復元し，合成音声波形を生成する．STFTスペクトルを用いた音声合成は，ボコーダによる音質劣化を回避するのみならず，スペクトルの領域で適用される音声強調 [6] などの技術を統合した音声合成技術 [7] を実現する．しかし，従来のボコーダ特徴量を用いた音声合成と同様に，音響モデルから生成されるスペクトルの過剰な平滑化 [4, 8] が発生し，音質が劣化する．これまでに我々は，ボコーダ特徴量を用いた音声合成の音質を改善させる手法として，敵対的学習 [9] の枠組みに基づく音響モデル学習法 (敵対的DNN音声合成) [10] を提案し，その有効性を確認している．敵対的DNN音声合成では，自然音声パラメータと合成音声パラメータの分布の違いを補償することで，過剰な平滑化の影響を緩和する．敵対的DNN音声合成は，STFTスペクトルを用いた音声合成に拡張可能だが，特徴量の次元数の多さや分布の複雑さにより，音響モデルの学習が困難となる．

STFTスペクトルを用いた音声合成の音質を改善させる手法として，本稿では，低周波数解像度のSTFTスペクトルを用いた敵対的DNN音声合成を提案する．低周波数解像度における識別モデルは，周波数方向のaverage poolingの結果として得られる低周波数解像度の対数振幅スペクトルを用いて，自然音声と合成音声を識別する．音響モデル学習時の損失関数は，元の周波数解像度における自然音声と合成音声の対数振幅スペクトルの二乗誤差と，低周波数解像度における識別モデルを詐称するための損失の重み付き和として定義される．低周波数解像度の対数振幅スペクトルは，フィルタバンクを模倣した特徴量とみなすことができるため，敵対的学習により，自然音声と合成音声のスペクトル包絡の違いを補償できる．さらに，この枠組みを拡張し，低周波数解像度と元の周波数解像度の両方における識別モデルを用いた多重周波数解像度の音響モデル学習法も新たに提案する．実験的評価では，(1) 低周波数解像度のSTFTスペクトルを用いた敵対的DNN音声合成が，ハイパーパラメータの設定に対して頑健に音質改善効果をもたらすこと，及び，(2) 低，多重，及び，元の周波数解像度を用いた合成音声の評価結果から，低周波数解像度の利用が音質改善に最も有効であることを示す．

2 従来手法

2.1 STFTスペクトルを用いたDNN音声合成 [4]

DNN音響モデルは，テキスト特徴量から対数振幅スペクトルを予測するように学習される．音響モデル学習時の損失関数は，自然音声の対数振幅スペクトル系列 $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \cdots, \boldsymbol{y}_t^\top, \cdots, \boldsymbol{y}_T^\top]^\top$ と，合成音声の対数振幅スペクトル系列 $\hat{\boldsymbol{y}} = [\hat{\boldsymbol{y}}_1^\top, \cdots, \hat{\boldsymbol{y}}_t^\top, \cdots, \hat{\boldsymbol{y}}_T^\top]^\top$ の間の二乗誤差として次式で与えられる．

$$L_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{T}(\hat{\boldsymbol{y}} - \boldsymbol{y})^\top (\hat{\boldsymbol{y}} - \boldsymbol{y}) \tag{1}$$

ここで，t はフレームインデックス，T は総フレーム数，$\boldsymbol{y}_t = [y_t(1), \cdots, y_t(F)]^\top$ はフレーム t における対数振幅スペクトル，F は周波数ビン数である．合成音声波形生成時には，Griffinらの位相復元アルゴリズム [5] を用いて，生成された対数振幅スペクトルから位相情報を復元する．

2.2 敵対的DNN音声合成 [10]

敵対的DNN音声合成では，敵対的学習 (Generative Adversarial Network: GAN) [9] の枠組みに基づき，合成音声と自然音声の特徴量の分布間距離を最小化することで合成音声の品質を改善する．学習時には，自然音声の特徴量と合成音声の特徴量を識別する識別モデル $D(\cdot)$ と，音声合成の音響モデルを交互に更新する．識別モデルの更新に用いる識別損失 (cross-entropy 関数) $L_{\mathrm{D}}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ は，次式で与えられる．

$$L_{\mathrm{D}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{D},1}(\boldsymbol{y}) + L_{\mathrm{D},0}(\hat{\boldsymbol{y}}) \tag{2}$$

$$L_{\mathrm{D},1}(\boldsymbol{y}) = -\frac{1}{T}\sum_{t=1}^{T} \log D(\boldsymbol{y}_t) \tag{3}$$

$$L_{\mathrm{D},0}(\hat{\boldsymbol{y}}) = -\frac{1}{T}\sum_{t=1}^{T} \log\left(1 - D(\hat{\boldsymbol{y}}_t)\right) \tag{4}$$

ここで，$L_{\mathrm{D},1}(\boldsymbol{y})$ と $L_{\mathrm{D},0}(\hat{\boldsymbol{y}})$ はそれぞれ自然音声と合成音声に対する損失である．識別モデルは，式 (2) の識別損失の勾配を用いたbackpropagationにより更新され，自然音声に対して1を，合成音声に対して0を出力するように学習される．識別モデルの更新後，次式に示す損失関数を最小化するように音響モデルを更新する．

$$L_{\mathrm{G}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \omega_{\mathrm{D}} \frac{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]}{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{ADV}}]} L_{\mathrm{ADV}}(\hat{\boldsymbol{y}}) \tag{5}$$

ここで，$L_{\mathrm{ADV}}(\hat{\boldsymbol{y}}) = L_{\mathrm{D},1}(\hat{\boldsymbol{y}})$ は識別モデルを詐称するための敵対損失であり，合成音声特徴量の分布を自然音声に近づける効果を持つ．ω_{D} は，敵対損失の影

*Adversarial DNN-Based Speech Synthesis Using Multi-Frequency Resolution STFT Spectra, by SAITO, Yuki, TAKAMICHI, Shinnosuke, and SARUWATARI, Hiroshi (The University of Tokyo).

響を調整するハイパーパラメータである．$\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]$ と $\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{ADV}}]$ はそれぞれ L_{MSE} と L_{ADV} の期待値であり，2 つの損失関数のスケールを調整する役割を持つ．

3 提案手法

3.1 低周波数解像度の STFT スペクトルを用いた敵対的 DNN 音声合成

2.2 節の手法は，STFT スペクトルを用いた音声合成の枠組みに適用できる．しかし，対数振幅スペクトルは従来のボコーダ特徴量と比較して高次元であり，分布形状も複雑となるため，敵対的学習による分布補償の有効性が低下すると予想される．本稿では，元の周波数解像度における対数振幅スペクトル $\boldsymbol{y}$ を低周波数解像度スペクトル $\boldsymbol{y}^{(\mathrm{L})}$ に圧縮し，低周波数解像度において自然音声と合成音声を識別する識別モデル $D^{(\mathrm{L})}(\cdot)$ を導入する．周波数解像度の圧縮を行う average pooling 関数を $\boldsymbol{\phi}(\cdot)$ とすると，フレーム t における低周波数解像度スペクトルの f 番目の周波数ビンの要素 $y_t^{(\mathrm{L})}(f)$ は次式で計算される．

$$y_t^{(\mathrm{L})}(f) = \frac{1}{w} \sum_{i=-p+1+(f-1)s}^{-p+1+(f-1)s+w} y_t(i) \tag{6}$$

ここで，p は zero-padding のサイズ，w は average pooling の窓幅，s はストライド幅を表す．ただし $i < 1$ もしくは $i > F$ のときの $y_t(i)$ は 0 とする．低周波数解像度スペクトルのビン数 $F^{(\mathrm{L})}$ は，次式で計算される．

$$F^{(\mathrm{L})} = \frac{F + 2p - w}{s} + 1 \tag{7}$$

上記の過程は，スペクトル包絡の特徴を表すフィルタバンクのパラメータを STFT スペクトルから抽出する枠組みと類似している．低周波数解像度スペクトルを用いた敵対的 DNN 音声合成における音響モデル学習時の損失関数は，次式で与えられる．

$$L_{\mathrm{G}}^{(\mathrm{Multi})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \omega_{\mathrm{D}}^{(\mathrm{L})} \frac{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]}{\mathbb{E}_{\hat{\boldsymbol{y}}^{(\mathrm{L})}}[L_{\mathrm{ADV}}]} L_{\mathrm{ADV}}\left(\hat{\boldsymbol{y}}^{(\mathrm{L})}\right) \tag{8}$$

ここで，合成音声の低周波数解像度スペクトルは，$\hat{\boldsymbol{y}}^{(\mathrm{L})} = \boldsymbol{\phi}(\hat{\boldsymbol{y}})$ として計算され，$\omega_{\mathrm{D}}^{(\mathrm{L})}$ は第二項の敵対損失の影響を調整するハイパーパラメータである．式 (8) の損失関数は，元の周波数解像度における二乗誤差と，低周波数解像度における敵対損失の重み付き和とみなせる．低周波数解像度スペクトルの分布は，元の周波数解像度よりも単純化されるため，敵対的学習の困難性を緩和することが期待できる．また，低周波数解像度スペクトルにおける分布間差異を最小化するため，音韻性の復元による音質改善を期待できる．音響モデルの更新後には，2.2 節の手法と同様に，低周波数解像度における識別モデルを更新する．この更新で最小化される損失関数は，式 (2) における $\boldsymbol{y}$ 及び $\hat{\boldsymbol{y}}$ をそれぞれ $\boldsymbol{y}^{(\mathrm{L})}$ と $\hat{\boldsymbol{y}}^{(\mathrm{L})}$ で置き換えたものと等価である．

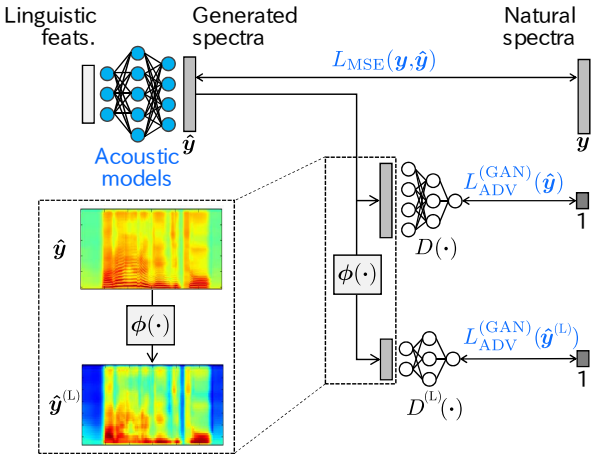

Fig. 1　多重周波数スペクトルを用いた敵対的 DNN 音声合成の音響モデル学習時の損失関数の計算手順．$\boldsymbol{\phi}(\cdot)$ は，元の周波数解像度における対数振幅スペクトルを周波数方向に圧縮する average-pooling である．

3.2 多重周波数解像度の STFT スペクトルを用いた敵対的 DNN 音声合成

3.1 節で提案する学習法は，元の周波数解像度における識別モデル $D(\cdot)$ も考慮した学習法に拡張できる．多重周波数解像度スペクトルを用いた敵対的 DNN 音声合成における音響モデル学習時の損失関数は，次式で与えられる．

$$L_{\mathrm{G}}^{(\mathrm{Multi})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \omega_{\mathrm{D}} \frac{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]}{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{ADV}}]} L_{\mathrm{ADV}}(\hat{\boldsymbol{y}}) + \omega_{\mathrm{D}}^{(\mathrm{L})} \frac{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]}{\mathbb{E}_{\hat{\boldsymbol{y}}^{(\mathrm{L})}}[L_{\mathrm{ADV}}]} L_{\mathrm{ADV}}\left(\hat{\boldsymbol{y}}^{(\mathrm{L})}\right) \tag{9}$$

元の周波数解像度における敵対損失に対する重み ω_{D} を 0 に設定すると，この損失関数は低周波数解像度スペクトルを用いた場合（式 (8)）と等価になる．提案手法における音響モデル学習時の損失関数の計算手順を図 1 に示す．ここで，元の周波数解像度及び低周波数解像度における識別モデルは，それぞれ独立に学習される．

3.3 考察

先行研究として，Kaneko ら [11] は，敵対的学習を用いた STFT スペクトルのためのポストフィルタを提案している．ポストフィルタを用いた手法では，音声特徴量の生成に加えてポストフィルタの処理が必要となるが，提案手法では学習時と同様の処理で音声特徴量を生成可能である．また，[11] では STFT スペクトルを帯域分割し，各帯域で独立に敵対的学習によるポストフィルタを構築するため，スペクトル全体としての構造や相関を無視している．一方で，提案手法では元の周波数解像度での生成誤差を考慮しつつ，異なる周波数解像度での分布の違いを補償するため，スペクトル全体としての整合性を保った学習が可能である．

提案手法は，ボコーダ特徴量を用いた音声合成における敵対的学習の枠組みを，STFT スペクトルを用いた音声合成に拡張した手法と解釈できる．同様に，

音声波形を直接的に生成する音声合成 [12, 13] における敵対的学習の提案も期待できる．

4 実験的評価

4.1 実験条件

実験的評価に用いるデータとして，JSUT コーパス [14] の一部から抽出した女性話者による 4007 文の発話音声を利用し, 3808 文を学習に，199 文を評価に用いる．学習データのサンプリング周波数は 16 kHz であり，フレーム長は 400 サンプル (25 ms)，フレームシフトは 80 サンプル (5 ms)，FFT 長は 1024 サンプルである．FFT 分析時の窓には，Hamming 窓を用いる．学習時には，実数値を取るコンテキストラベルと対数振幅スペクトルを平均 0，分散 1 となるように正規化し，無音区間の 90%を削除する．

音響モデルと識別モデルの DNN アーキテクチャは，すべて Feed-Forward である．音響モデルの入力は，439 次元のコンテキストラベル，3 次元の継続長特徴量に加え，先行研究 [4] と同様の連続 F_0 と U/V を含む 444 次元のベクトルである．F_0 の抽出には，STRAIGHT ボコーダ [1] を用いる．F_0 と継続長特徴量を予測する DNN は，別途構築する．音響モデルは 513 次元の対数振幅スペクトルをフレーム毎に予測する．音響モデルの隠れ層数は 3，隠れ素子数は 1024，隠れ層及び出力層の活性化関数はそれぞれ Rectified Linear Unit (ReLU) [15] 及び線形関数である．元の周波数解像度における識別モデルの隠れ層数は 3，隠れ素子数は 512，隠れ層及び出力層の活性化関数はそれぞれ ReLU 及び sigmoid 関数である．低周波数解像度における識別モデルの隠れ層数及び活性化関数は元の周波数解像度におけるものと同じだが，入力される低周波数解像度スペクトルのビン数 $F^{(\mathrm{L})}$ に応じて隠れ素子数を変化させる．以降の評価では，式 (7) における zero-padding のサイズを $p = 6$，ストライド幅を $s = w/2$ として設定し，average pooling の窓幅 w を 14, 30, 70 と変化させる．それぞれの窓幅に対応する低周波数解像度スペクトルのビン数 $F^{(\mathrm{L})}$ は 74，34，14 であり，隠れ素子数は 128，64，32 である．

まず，音響モデルの初期化として，反復回数 25 回の二乗誤差最小化に基づく学習 [4] を行う．次に，識別モデルの初期化として，自然音声の対数振幅スペクトルと初期化後の音響モデルから生成された合成音声の対数振幅スペクトルを識別するような反復回数 5 回の学習を行う．その後，初期化された音響モデルと識別モデルを用いて，反復回数 25 回の敵対的 DNN 音声合成の学習を行う．最適化アルゴリズムとして，学習率 0.01 の AdaGrad [16] を用いる．

4.2 主観評価

主観評価として，我々のクラウドソーシングによる評価システムを用いて，合成音声の音質に関するプリファレンス AB テストを実施する．各評価における受聴者数は 25 人であり，1 人あたり 10 サンプルの音声の音質を評価する．以降の評価において，“Baseline” は，従来の二乗誤差最小化に基づく学習 [4] を意味する．すなわち，提案手法における損失関数（式 (9)）において，ω_{D} と $\omega_{\mathrm{D}}^{(\mathrm{L})}$ を両方 0 に設定したものと等価である．

Table 1　音質に関するプリファレンススコアと p 値 (元の周波数解像度を用いた敵対的 DNN 音声合成)

ω_{D}	Score	p-value	ω_{D}
0.0	**0.700** vs. 0.300	$< 10^{-10}$	0.5
1.0	0.280 vs. **0.720**	$< 10^{-10}$	0.0
0.5	0.496 vs. **0.504**	8.6×10^{-1}	1.0

Table 2　音質に関するプリファレンススコアと p 値 (低周波数解像度を用いた敵対的 DNN 音声合成における w の影響)

(a) “Baseline” と敵対的 DNN 音声合成の比較

	Score	p-value	
$w = 14$	**0.568** vs. 0.432	2.3×10^{-3}	Baseline
$w = 30$	**0.572** vs. 0.428	1.2×10^{-3}	Baseline
$w = 70$	**0.528** vs. 0.472	2.1×10^{-1}	Baseline

(b) 敵対的 DNN 音声合成での比較

	Score	p-value	
$w = 14$	0.488 vs. **0.512**	5.9×10^{-1}	$w = 30$
$w = 30$	**0.532** vs. 0.468	1.5×10^{-1}	$w = 70$
$w = 70$	0.472 vs. **0.528**	2.1×10^{-1}	$w = 14$

4.2.1 元の周波数解像度を用いた敵対的 DNN 音声合成の評価

まず，元の周波数解像度を用いた敵対的 DNN 音声合成（即ち，従来手法 [10] を直接適用させた手法）の有効性を調査する．ここでは，$\omega_{\mathrm{D}}^{(\mathrm{L})} = 0$ に固定し，“Baseline”（即ち，$\omega_{\mathrm{D}} = 0.0$）と，提案手法において $\omega_{\mathrm{D}} = 0.5, 1.0$ として設定させた手法を比較する．評価結果を Table 1 に示す．$\omega_{\mathrm{D}} = 0.0$ とした従来手法と比較して，敵対的 DNN 音声合成を用いることによる音質の劣化が確認できる．故に，ボコーダ特徴量を用いた音声合成において有効であった手法 [10] を STFT スペクトルを用いた音声合成に適用するだけでは，合成音声の音質は改善しないことを示した．

4.2.2 低周波数解像度を用いた敵対的 DNN 音声合成の評価

次に，提案手法における average pooling の窓幅 w の影響を調べるために，$\omega_{\mathrm{D}} = 0$ 及び $\omega_{\mathrm{D}}^{(\mathrm{L})} = 1$ とし，“Baseline” と，提案手法において $w = 14, 30, 70$ とした手法を比較する．評価結果を Table 2 に示す．Table 2(a) より，w の設定に依らず，低周波数解像度を用いた敵対的 DNN 音声合成による音質の改善が確認できる．また，Table 2(b) より，提案手法は w の設定に対して頑健に動作することを確認できる．以降の評価では，他の設定に比べてわずかにスコアが高い $w = 30$ を利用する．

さらに，敵対的 DNN 音声合成における敵対損失に対する重みを調整するハイパーパラメータの影響も調査する．ここでは，$\omega_{\mathrm{D}} = 0$ とし，“Baseline”（即ち，$\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.0$）と，提案手法において $\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5, 1.0$ とした手法を比較する．評価結果を Table 3 に示す．評価結果より，低周波数解像度を用いた敵対的 DNN 音声合成は，average pooling の窓幅のみならず，損失関数のハイパーパラメータの設定に対しても頑健に音質改善効果をもたらすことを示した．

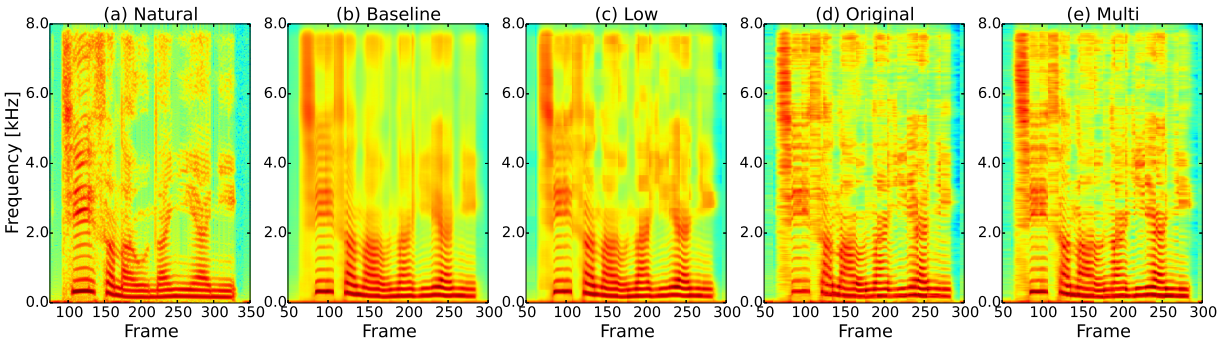

Fig. 2　自然音声と合成音声の対数振幅スペクトル．合成音声の継続長は自然音声と異なるため，自然音声の横軸を調整して表示している．

Table 3　音質に関するプリファレンススコアと p 値 (低周波数解像度を用いた敵対的 DNN 音声合成における $\omega_{\mathrm{D}}^{(\mathrm{L})}$ の影響)

$\omega_{\mathrm{D}}^{(\mathrm{L})}$	Score	p-value	$\omega_{\mathrm{D}}^{(\mathrm{L})}$
0.0	0.456 vs. **0.544**	4.9×10^{-2}	0.5
1.0	**0.588** vs. 0.412	7.6×10^{-5}	0.0
0.5	**0.504** vs. 0.496	8.6×10^{-1}	1.0

Table 4　音質に関するプリファレンススコアと p 値 (種々の周波数解像度を用いた敵対的 DNN 音声合成)

	Score	p-value	
Low	**0.808** vs. 0.192	$< 10^{-10}$	Multi
Multi	0.492 vs. **0.508**	7.2×10^{-1}	Original
Original	0.192 vs. **0.808**	$< 10^{-10}$	Low

4.2.3　種々の周波数解像度を用いた敵対的 DNN 音声合成の評価

最後に，低，多重，及び、元の周波数解像度を用いた敵対的 DNN 音声合成を比較するために，以下の 3 手法の合成音声を評価する．

Original: $(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})}) = (1.0, 0.0)$ とした提案手法

Low: $(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})}) = (0.0, 1.0)$ とした提案手法

Multi: $(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})}) = (1.0, 1.0)$ とした提案手法

評価結果を Table 4 に示す．評価結果より，低周波数解像度を用いた場合の音質が最も高く，元の周波数解像度及び多重周波数解像度を用いた場合の音質は同程度であることが確認できる．この結果を議論するために，自然音声と合成音声の対数振幅スペクトルを Fig. 2 に示す．”Baseline” (Fig. 2(b)) において平滑化されていたスペクトルが，低周波数解像度における分布補償 (Fig. 2(c)) により復元されていることを確認できる．一方で，元の周波数解像度及び多重周波数解像度を用いた場合 (Fig. 2(d), (e)) では，スペクトルは復元されているものの，フレーム間の不連続性が生じている．これは，系列モデリング [17, 18] や条件付き GAN [19, 20] などにより緩和できると考えられる．

5　おわりに

本稿では，STFT スペクトルを用いた敵対的 DNN 音声合成を提案し，実験的評価により，低周波数解像度を用いた敵対的 DNN 音声合成が音質改善に最も有効であることを示した．今後は，元の周波数解像度の効果的な利用法を検討する．

謝辞: 本研究は，セコム科学技術振興財団，及び JSPS 科研費 16H06681，17H06101 の支援を受けた．

参考文献

[1] Kawahara et al., *Speech Commun.*, Vol. 27, No. 3–4, pp. 187–207, 1999.
[2] Morise et al., *IEICE Trans. on Inf. and Syst.*, Vol. E99-D, No. 7, pp. 1877–1883, 2016.
[3] Zen et al., *Proc. ICASSP*, pp. 7962–7966, 2013.
[4] Takaki et al., *Proc. INTERSPEECH*, pp. 1128–1132, 2017.
[5] Griffin et al., *IEEE Trans. on ASLP*, Vol. 32, No. 2, pp. 236–243, 1984.
[6] Xu et al., *IEEE/ACM Trans. on ASLP*, Vol. 23, No. 1, pp. 7–19, 2015.
[7] 宇根 他, 情報処理学会研究報告, 2017-SIG-SLP-118, pp. 1–6, 2017.
[8] Toda et al., *Proc. INTERSPEECH*, pp. 1632–1636, 2016.
[9] Goodfellow et al., *Proc. NIPS*, pp. 2672–2680, 2014.
[10] Saito et al., *IEEE/ACM Trans. on ASLP*, Vol. 26, No. 1, pp. 84–96, 2018.
[11] Kaneko et al., *Proc. INTERSPEECH*, pp. 3389–3393, 2017.
[12] Oord et al., *arXiv:1609.03499*, 2016.
[13] Mehri et al., *arXiv:1612.07837*, 2016.
[14] Sonobe et al., *arXiv:1711.00354*, 2017.
[15] Glorot et al., *Proc. AISTATS*, pp. 315–323, 2014.
[16] Duchi et al., *The Journal of Machine Learning Research*, Vol. 12, pp. 2121–2159, 2011.
[17] Hochreiter et al., *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
[18] Zen et al., *Proc. ICASSP*, pp. 4470–4474, 2015.
[19] Mirza et al., *arXiv:1411.1784*, 2014.
[20] Yang et al., *Proc. ASRU*, pp. 685–691, 2017.