UNIVERSITY OF CAMBRIDGE

MPhil in Machine Learning and, Machine Intelligence 2018-2019

Module Coursework Feedback

Module Title: Statistical Speech Synthesis

Module Code: MLMI10

Candidate Number: F606F

Coursework Number: 1

I confirm that this piece of work is my own unaided effort and adheres to the Department of Engineering's guidelines on plagiarism. √

Date Marked: Click here to enter a date. Marker's Name(s): Click here to enter text.

Marker's Comments:

This piece of work has been completed to the following standard *(please circle as appropriate)*:

	Distinction			**Pass**			**Fail (C+ - marginal fail)**		
Overall assessment (circle grade)	**Outstanding**	**A+**	**A**	**A-**	**B+**	**B**	**C+**	**C**	**Unsatisfactory**
Guideline mark (%)	90-100	80-89	75-79	70-74	65-69	60-64	**55-59**	**50-54**	**0-49**
Penalties	**10% of mark for each day, or part day, late (Sunday excluded).**								

The assignment grades are given **for information only**; results are provisional and are subject to confirmation at the Final Examiners Meeting and by the Department of Engineering Degree Committee.

MLMI10: Statistical Speech Synthesis
Practical Report: Parametric Speech Synthesis

Experiments: Synthesis and Trajectories

1. Original waveform

The `audacity` is used to examine the spectrogram from the file of `utt1.wav`. From `utt1.txt`, it is known that the waveform is speaking "Keep the hatch tight and the watch constant".

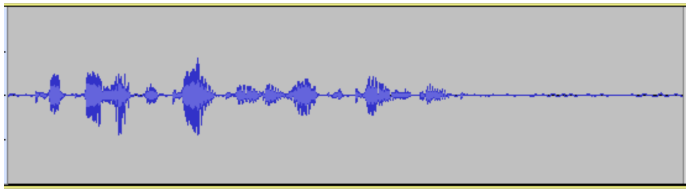


Figure 1: Spectrogram of the original waveform

It is observed that the original waveform which contains both voiced and unvoiced regions is not very smooth and continuous but quite discrete. For the unvoiced region, the amplitude is not always zero, there exists some noise in the recording.

2. Model parameters trajectories

Based on the labels generated from `txt2lab.sh`, i.e. `utt1.lab`, the trojectories of the model parameters can be generated by `lab2traj.sh`. The trajecotries of the model parameters include `utt1.mcep`, `utt1.apf` and `utt1.f0.txt`.

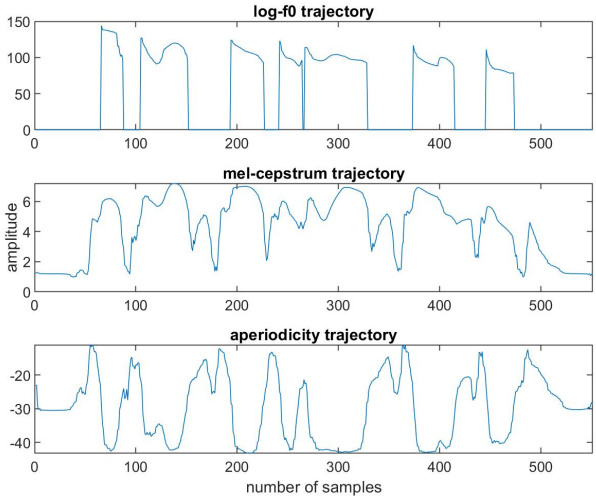


Figure 2: Trajectories of the model parameters

The script `load_traj.m` is used to examine the trajectories of `mcep` and `apf` with the size of 60 and 21 respectively. The `f0.txt` is examined by the matlab function `fscanf`. The number of samples is 551. Their trajectories (including the 1^{st} dimension of mel-cepstrum parameters and aperiodity parameters) are shown in Figure 2.
The number of voiced regions in Figure 1 is around 7 which matches the number of non-zero regions in log-f0 trajectory. Given that the term "aperiodity" in physics means being damped sufficiently to reach equilibrium without oscillation, it is sensible to conclude that the troughs in aperiodity trajectory correspond to the non-zero regions in log-f0 trajectory, and the peaks in aperiodity trajectory correspond to the zeros around the non-zero regions in log-f0 trajectory. It is more difficult to observe the pattern of the mel-cepstrum parameter trajectory and its relationship with other parameters trajectories because it is generated via many operations, such as Fourier transform, discrete cosing transform, mel-scale, etc, the trajectory becomes non-linear.

3. Generated waveform

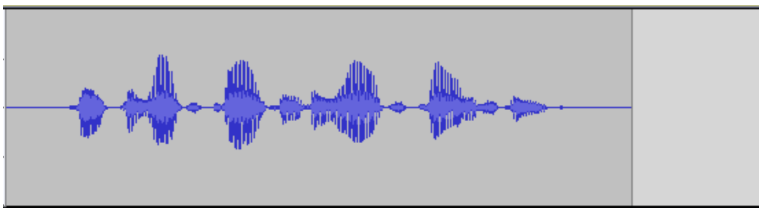


Figure 3: Spectrogram of the generated waveform

Compared to the original waveform from Figure 1, the audio of the generated waveform shown in Figure 3 sounds unnatural, there are more noise in the voiced regions.
When comparing their spectrograms, the original waveform has more noise in the unvoiced regions, but the generated one has no noise. In the voiced regions, there are more separations between generated dark blue spikes in the generated waveform. But the dark blue spikes are more compact in the original waveform. This may explain what was heard from the audio.

4. Generated waveform with zero log-f0 trajectory

A script `replace0.py` is written to replace the non-zero elements in `utt1.f0.txt` with zeros. With the modified log-f0 trajectory, a waveform is generated and its spectrogram is shown in Figure 4.

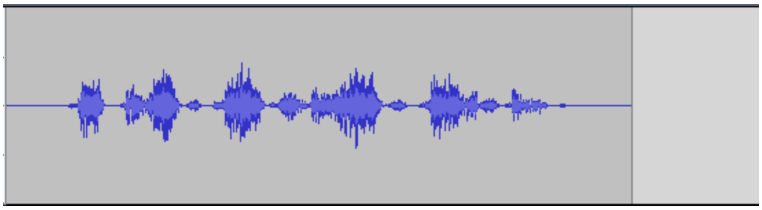


Figure 4: Spectrogram of the generated waveform with zero f0 trajectory

Compared to the previously generated waveform, the new waveform sounds more noisy and the voice sounds more flatten. This maybe because the amplitudes of the dark blue spikes are not too much greater than those of the light blue regions in the zero f0 trajectory, when compared with the greater separations shown in the previous generated trajectory.

5. Generated waveform with forced alignment

By using `load_htkdata.m` to examine `utt1.cmp`, the number of samples in the original trajectory is found to be 548 which are smaller than that of generated trajectory (551).

To generate the distance between the synthesis trajectories, the following procedure is adopted to achieve the best alignment. The command `alignsignals` is used to align both trajectories roughly, then the command `edr` is used to add and reduce zeros in the unvoiced regions so that individual voiced regions can be aligned properly. Lastly, the `alignsignals` is used again to align again. The results are shown in Figure 5.

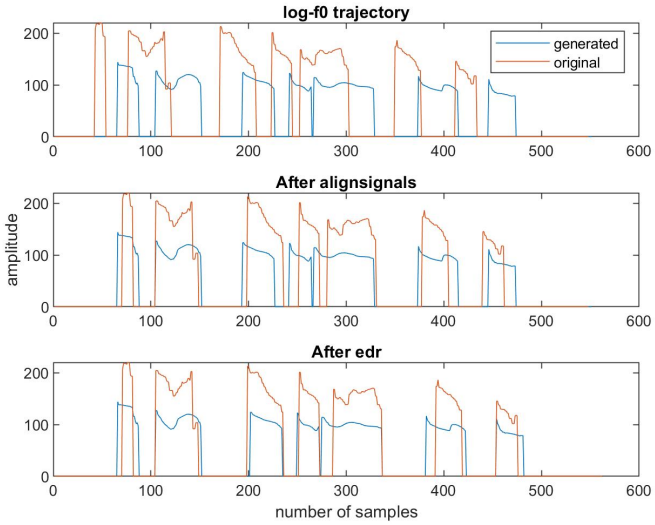


Figure 5: Trajectories of the log-f0 parameters

The procedure helps to align signals such that one's unvoiced regions will not match with other's voiced regions, and vice versa, so that the distance between can be minimized.
With the use of `lab2traj.sh`, it is possible to generate trajectories based on forced alignment of the waveform data with the acoustic model. The log-f0 trajectory aligned by the given script is shown in Figure 6.

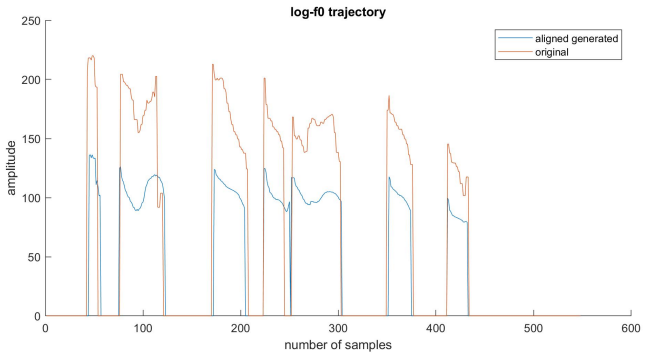


Figure 6: Trajectories of the log-f0 parameters

The final result of Figure 5 and the result of Figure 6 look quite similar. But the script `lab2traj.sh` does subsample the voiced regions of the generated trajectory intentionally so that the durations of the voiced regions match. And the start times of the voiced regions in the generated trajectory is shifted to match those of the original trajectory.
In general, these two log-f0 trajectories are formed by voiced and unvoiced regions. The elements of unvoiced regions are zeros, and the voiced regions are usually started from the peaks. The generated trajectory is more smooth but the original one is more discrete.

6. Generated waveform with original log-f0 trajectory replacement

Using the generated log-f0 trajectory in the top plot of Figure 7, the waveform can be generated and its spectrogram is illustrated in Figure 8(a). This is the waveform without any replacement.

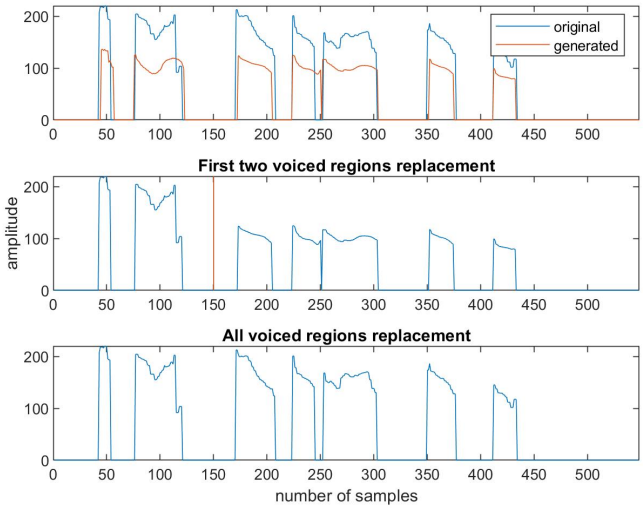
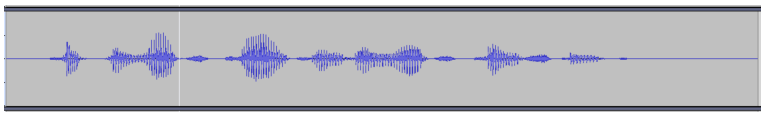
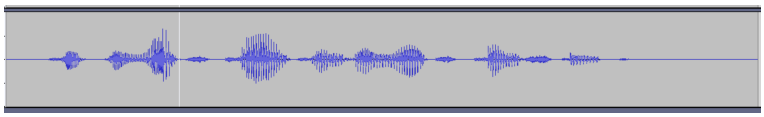


Figure 7: Replacements of log-f0 trajectories

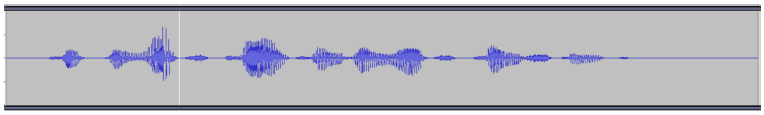
To investigate how replacing the synthesized trajectories with the original waveform parameters affects the waveform, several experiments are conducted. Firstly, the first two voiced regions of the generated log-f0 parameter are replaced by those of the original log-f0 parameter as shown in the middle plot of Figure 7. Using this modified log-f0 paramter, a waveform is generated and shown in Figure 8(b). Secondly, the whole generated log-f0 parameter is replaced by the original log-f0 parameter as shown in the bottom plot of Figure 7. Then another waveform is generated and shown in Figure 8(c).
Based on these three waveforms, how the replacement of the elements impacts the waveform can be concluded.



((a)) Waveform without replacement



((b)) Waveform with first 2 voiced regions replacement



((c)) Waveform with full replacement

Figure 8: Spectrogram of the forced alignment data with replacement. The regions before the white line are the first two voiced regions.

The results show that the replacement of log-f0 trajectory's voiced regions can directly manipulate the shape of the voiced regions in the spectrogram of the waveform. But the waveform illustrated in Figure 8(c) does

not look similar to that in Figure 1 even with the original log-f0 trajectory replacement, this is because both the mel-cepstrum and aperiodicity parameters trajectories are not yet replaced.

Experiments: Trajectory Generation [1]

The code written for this experiment is located in `/homes/ycl60/MLMI10/Final.ipynb`.

1. Product of experts

The trajectory of the 4^{th} dimension of the mel-cepstrum parameter can be computed by the Product of experts (PoE) model with the following equation,

$$\hat{\mathbf{x}}_{1:T} = \mu_{\mathbf{q}} = (\mathbf{W}'\bar{\mathbf{\Sigma}}_{\mathbf{q}}^{-\mathbf{1}}\mathbf{W})^{-1}\mathbf{W}'\bar{\mathbf{\Sigma}}_{\mathbf{q}}^{-\mathbf{1}}\bar{\mu}_{\mathbf{q}} \tag{1}$$

, where $\bar{\mu}_{\mathbf{q}}$ is a $3T \times 1$ mean parameter vector, $\bar{\mathbf{\Sigma}}_{\mathbf{q}}$ is a $3T \times 3T$ covariance parameter matrix of the state sequence q determined by the state duration probability density functions, $\mathbf{W}$ is a $3T \times T$ window matrix which transforms the static parameters to the dynamic paramters and T is the number of samples. And the vectors and matrices above are formed by the following procedures.
Read the file `utt1.cmp.expt` to find the static, delta and delta-delta parameters of the mean ($\mu_{\mathbf{s}} = [\Delta^{(0)}\mu_s, \Delta^{(1)}\mu_s, \Delta^{(2)}\mu_s]$) and variance ($\mathbf{\Sigma_s} = diag[\Delta^{(0)}\sigma_s^2, \Delta^{(1)}\sigma_s^2, \Delta^{(2)}\sigma_s^2]$) for each state s in each phone i. And read the file `utt1.dur.expt` to see the number of times (n_s) the state s parameters ($\mu_{\mathbf{s}}$ and $\mathbf{\Sigma_s}$) needed to concatenate to form each phone i parameters ($\mu_{\mathbf{i}}$ and $\mathbf{\Sigma_i}$). And finally form the parameters of whole sequence ($\bar{\mu}_{\mathbf{q}}$ and $\bar{\mathbf{\Sigma}}_{\mathbf{q}}$) by the concantenations of all phone i paramters ($\mu_{\mathbf{i}}$ and $\mathbf{\Sigma_i}$).
Thus, the mean parameter vector can be expressed as,

$$\bar{\mu}_{\mathbf{q}} = \{\mu_{\mathbf{i}}\}_{i=1}^{N} = \left\{ \{\mu_{\mathbf{s}}^{\mathbf{n_s}}\}_{s=2}^{6} \right\}_{i=1}^{N} = \left\{ \{\{\mathbf{\Delta}^{(\mathbf{d})}\mu_{\mathbf{s}}\}_{\mathbf{d=1}}^{\mathbf{3}\ \mathbf{n_s}}\}_{s=2}^{6} \right\}_{i=1}^{N} \tag{2}$$

The covariance parameter matrix can be expressed as,

$$\bar{\mathbf{\Sigma}}_{\mathbf{q}} = diag\{\mathbf{\Sigma_i}\}_{i=1}^{N} = diag\left\{ \{\mathbf{\Sigma_s^{n_s}}\}_{s=2}^{6} \right\}_{i=1}^{N} = diag\left\{ \{\mathbf{diag}\{\mathbf{\Delta}^{(\mathbf{d})}\sigma_{\mathbf{s}}^{\mathbf{2}}\}_{\mathbf{d=1}}^{\mathbf{3}\ \mathbf{n_s}}\}_{s=2}^{6} \right\}_{i=1}^{N} \tag{3}$$

Given the weights shown in the practical handout, $\mathbf{W}$ should be computed in the form of,

$$\begin{bmatrix} 1 & 0 & 0 & \dots & \dots & \dots & \dots \\ 0 & 0.1 & 0.2 & \dots & \dots & \dots & \dots \\ -0.28 & -0.14 & 0.28 & \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & 0 & \dots & \dots & \dots \\ -0.1 & 0 & 0.1 & 0.2 & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \dots & \dots \\ 0 & 0 & \dots & \dots & 0.28 & -0.14 & -0.28 \end{bmatrix}$$

Therefore, the computed trajectory $\hat{\mathbf{x}}_{1:T} = \mu_{\mathbf{q}}$ is a vector of $T \times 1$ vector.
The trajectory computed by Equation 1 is shown in the top plot of Figure 9. When compared to the generated trajectory with the forced alignment and the original trajectory in the middle and bottom plot of Figure 9 respectively, the computed trajectory has a smaller number of samples. The top plot of figures 10 shows the trajectory computed in this section has a relatively long stable time at the beginning.

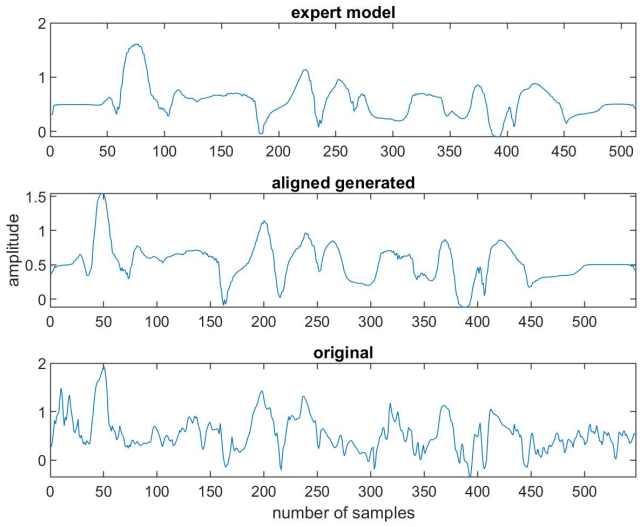
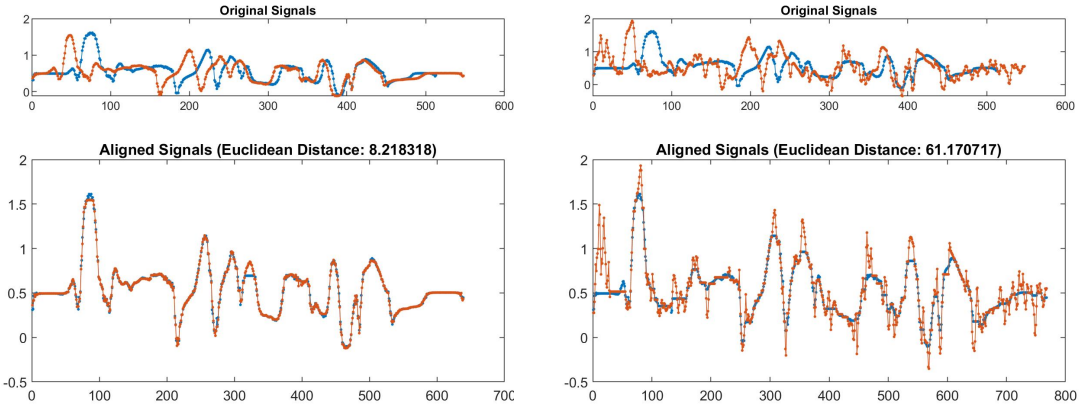


Figure 9: Trajectories of the c_3 parameters. Top: trajectory computed via PoE model. Middle: trajectory from Section – Synthesis and Trajectories Part 5. Bottom: trajectory from utt1.cmp.

In this section, the dynamic time warping (DTW) is used to align the signals for further comparison. From Figure 10, it is observed that the distance between the original and the PoE-computed trajectories is greater than that between the one with forced alignment and the PoE-computed trajectory. This may indicate that the PoE-computed trajectory is closer to the one with forced alignment than the actual trajectory. And these generated trajectories are too smooth when compared with the actual trajectory.



((a)) ((b))

Figure 10: Distance between trajectories. The left plot (a) shows how the force-align trajectory aligns with the PoE-computed trajectory and the distance between them. The right plot (b) shows how the original trajectory aligns with the PoE-computed trajectory and the distance between them.

This trajectory generation process takes approximately 3.83 seconds. It is possible to make it more efficient by calculating the phone frame trajectory ($\mu_{\mathbf{q},\mathbf{i}}$) individually, and lastly concatenating the resulting trajectories to form the whole the trajectory ($\mu_{\mathbf{q}}$). This method only requires a shorter processing time around 0.334 seconds because the inversion of multiple smaller matrices needs less times than that of a bigger matrix. But the resulting trajectory will become different, and the distance between the trajectory generated by this method and other trajectories will also increase.
Besides, instead of using the pseudo-inverse, the reciprocal of variance can be computed to form the diagonal

element of the convariance matrix inversion since the covariance matrix is a diagonal matrix. With the pseudo-inverse, the inversion takes around 4 seconds; but with the reciprocal method, it now takes 0.001 seconds. This shows that this little change can accelerate the process, and still keep the same trajectory shape.

2. Global variance model

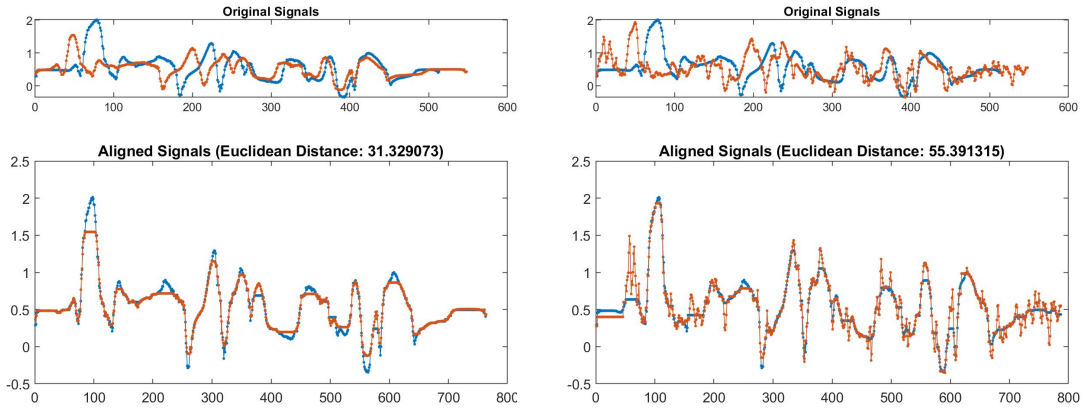
In the paper [1], it has introduced a quantity named global variance (GV) to reduce the distance between the generated trajectory and the actual one. This model can be implemented with the following frameworks.

a. Expert within a product of experts framework

This framework involves using optimizer to maximize the log probabilities to predict the trajectory, which states as follows.

$$\hat{\mathbf{x}}_{1:T} = \arg\max_{\mathbf{x}_{1:T}} \left\{ \log \mathcal{N}(\mathbf{x}_{1:T}, \mu_{\mathbf{q}}, \mathbf{\Sigma_q}) + \alpha \log \mathcal{N}(\frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_t - \mu)^2; \mu_{\mathbf{gv}}, \mathbf{\Sigma_{gv}}) \right\} \tag{4}$$

, where $\alpha = 3T$, $\mu = \frac{1}{T}\sum_{t=1}^{T} x_t$, $\mathbf{\Sigma}_q^{-1} = \mathbf{W}'\bar{\mathbf{\Sigma}}_{\mathbf{q}}^{-\mathbf{1}}\mathbf{W}$, $\mu_{\mathbf{gv}}$ is the mean parameter of the GV model and $\mathbf{\Sigma_{gv}}$ is the covariance parameter of the GV model.
The optimized trajectory with GV is shown to have a greater distance to the system-generated trajectory but a smaller distance to the actual trajectory, relative to the computed trajectory without GV, in Figure 11.
This shows the introduction of GV in PoE model can help the generated trajectory be more like the actual trajectory, and make it less smooth relative to the original generated trajectory.



((a)) ((b))

Figure 11: Distance between trajectories. The left plot (a) shows how the force-align trajectory aligns with the GV-computed trajectory and the distance between them. The right plot (b) shows how the original trajectory aligns with the GV-computed trajectory and the distance between them.

b. Constraint within the optimization approach

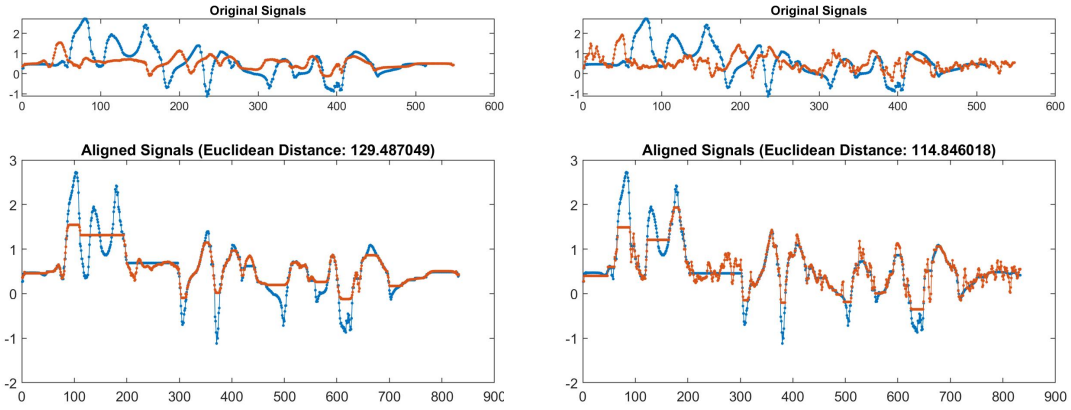
Some research studies [2] also have suggested a quicker way to predict trajectory without optimizing the function, but with the use of a Lagrange multiplier λ. However, the prediction of trajectory (given by Equation 5) still involves calculating a large matrix inversion, so the method might not be quicker than that from the PoE model.

$$\hat{\mathbf{x}}_{1:T} = (\mathbf{\Sigma}_{\mathbf{q}}^{-\mathbf{1}} - \lambda\mathbf{I})^{-1}(\mathbf{b} - \nu(\lambda)\mathbb{I}) \tag{5}$$

The global variance now becomes,

$$\nu(\lambda) = \frac{\lambda \mathbf{b}'(\boldsymbol{\Sigma}_\mathbf{q}^{-\mathbf{1}} - \lambda \mathbf{I})^{-1}\mathbb{I}}{T + \lambda \mathbb{I}'(\boldsymbol{\Sigma}_\mathbf{q}^{-\mathbf{1}} - \lambda \mathbf{I})^{-1}\mathbb{I}} \tag{6}$$

, where $\mathbf{b} = {\boldsymbol{\Sigma}_\mathbf{q}}^{-1}\mu_\mathbf{q}$, λ is one of the eigenvalues of $\boldsymbol{\Sigma}_\mathbf{q}^{-\mathbf{1}}$ such that $\boldsymbol{\Sigma}_\mathbf{q}^{-\mathbf{1}} - \lambda\mathbf{J}$ is definite positive, given $\mathbf{J} = \mathbf{I} - \frac{1}{T}\mathbb{I}\mathbb{I}'$, $\mathbf{I}$ is the identity matrix and $\mathbb{I}$ is a vector of ones of length T.



((a)) ((b))

Figure 12: Distance between trajectories. The left plot (a) shows how the force-align trajectory aligns with the constraint-computed trajectory and the distance between them. The right plot (b) shows how the original trajectory aligns with the constraint-computed trajectory and the distance between them.

From Figure 12, the performance of this method is not good. The result has greater distances to the actual trajectory and the generated trajectory, when compared to those of the PoE with and without GV.
The global variance of the generated trajectory in part (a) is approximately 0.166, and that of the expert is 0.170, their distance is around 0.004. The global variance in part(b) is 0.515. The GV distance between the experts and this method is 0.345. Given that the GV of PoE model is 0.0870 and the difference is 0.08256, it can be concluded that the PoE model with GV has the smallest GV distance and the constraint model has the largest GV distance, so that the method (a) is better than (b).
Some suggest that the generated trajectory by the constraint model can be put back into the optimizer as the new initial point, so that the optimizer can find the global optimum more quickly. After some testings, it turns out the new initial point does not accelerate the optimization speed that much, and the trajectory becomes the one found by GV-PoE model.
Another equation for the optimal trajectory has also been tested.

$$\hat{\mathbf{x}}_{1:T} = ({\boldsymbol{\Sigma}_\mathbf{q}}^{-1} - \lambda\mathbf{J})^{-1}\mathbf{b} \tag{7}$$

The global variance of the trajectory predicted by Equation 7 is 0.490 which is smaller than that from Equation 5 and closer to the actual trajectory. This is sensible because Equation 5 is just an approximation of Equation 7. But its performance is still worse than the GV-PoE model.

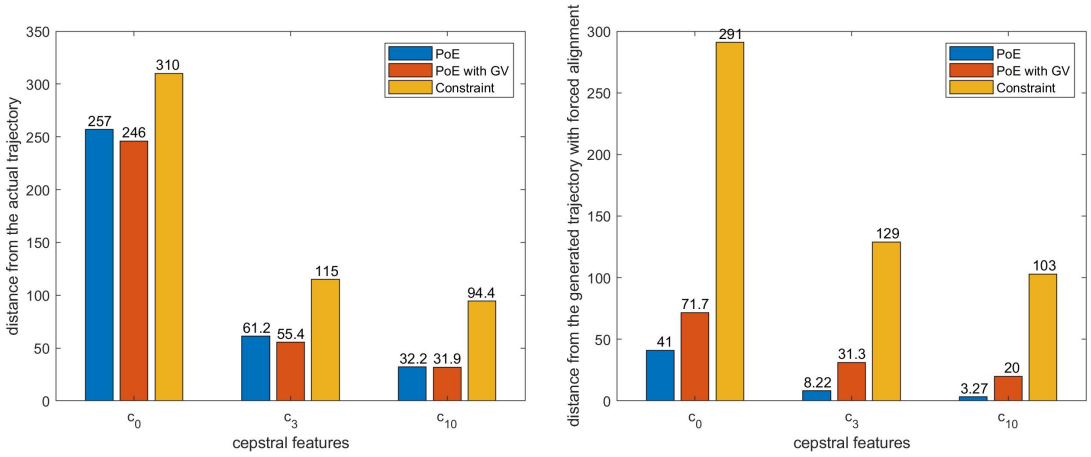


Figure 13: Different cepstral features trajectories distances with different prediction methods. The left plot (a) shows the distances to the actual trajectory. The right plot (b) shows the distances to the original generated trajectory.

The equal influence to all cepstral parameters by GV is expected. In Figure 13, it is observed that the introduction of GV to the PoE model generally reduces the distance to the actual trajectory. And the constraint model cannot improve the performance for all features. The use of either the GV-PoE or the constraint models increase the distance from the original generated trajectory.

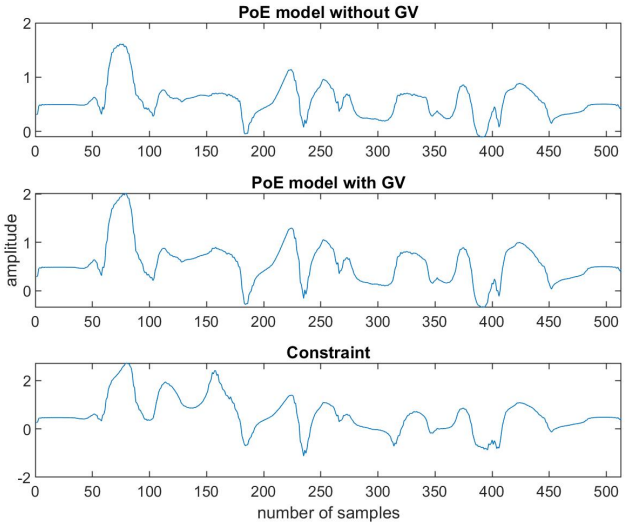


Figure 14: Trajectories computed by different methods

From Figure 14, it is observed that the results from PoE model and GV-PoE model are quite similar. The result of constraint model seems to approximate the correct shape but the peaks at the region of 100-200 th samples are too distinctive when compared with other generated trajectories.

3. Trends for all utterances

To investigate whether other utterances have similar trends, the c_3 cepstral features trajectories from `utt5` and `utt9` are extracted as well. They are then run via 3 different models to record their distance performance.

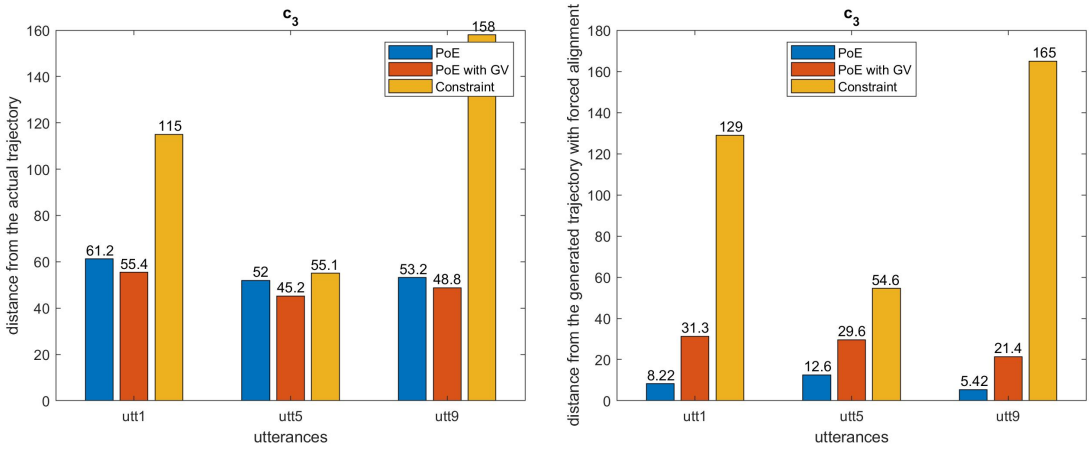


Figure 15: Cepstral parameter c_3 trajectories distances with different prediction methods and different utterances. The left plot (a) shows the distances to the actual trajectory. The right plot (b) shows the distances to the original generated trajectory.

As shown in Figure 15, other utterances, such as `utt5` and `utt9`, also show a similar trend. Compared with the PoE model, the PoE model with GV has a shorter distance from the actual trajectory but a longer distance from the original generated one. But the constraint model has longer distances to both the actual trajectory and the generated trajectory.

References

[1] H. Zen, M. J. F. Gales, Y. Nankaku, and K. Tokuda, "Product of experts for statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 794–805, 2012.

[2] M. Shannon and W. Byrne, "Fast, low-artifact speech synthesis considering global variance," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7869–7873, May 2013.