GEORGIA INSTITUTE OF TECHNOLOGY

SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2026 Fall 2013
Lab #7: Synthesis of Vowel Sounds & Spectral Analysis with DFT

Date: 21-24, October 2013

Each Lab assignment in ECE2026 consists of three parts: Pre-Lab, In-lab Tasks, and Take-home Questions. It requires you to come into lab prepared. Be sure to read the entire lab carefully before arriving.

Pre-Lab: You should read the Pre-Lab section of the lab and go over all exercises in this section before going to your assigned lab session. Although you do not need to turn in results from the Pre-Lab, doing the exercises therein will help make your in-lab experience more rewarding and go more smoothly with less frustration and panicking.

In-lab Tasks and Verification: There are a number of designated tasks for each student to accomplish during the lab session. Students are encouraged to read, prepare for and even attempt at these tasks beforehand. These tasks must be completed **during your assigned lab time** and the steps marked *Instructor Verification* must also be signed off **during the lab time**. One of the laboratory instructors must verify the appropriate steps by signing on the **Instructor Verification** line. When you have completed a step that requires verification, simply put a plastic cup on top of your PC and demonstrate the step to one of the TAs or the professor. (You can also use the plastic cups to indicate if you have a more general question, i.e. you can use it to get our attention even if you don’t have an Instructor Verification ready.)

Take-home Questions: At the end of each lab sheet below all the verification steps, several questions are to be answered by the student, who can choose to complete the answers while in the lab or after leaving the lab.

The lab sheet with all verification signatures and answers to the questions needs to be turned in to the Lab-grading TA at the beginning of the next lab session.

---

1 Introduction

The objective of this lab is to expose the student with some primitive concept in speech synthesis. The lab is designed to bring out some hands-on experience in synthesizing speech vowel sounds using sum of sinusoids, and to obtain understanding of a key speech characteristic called the pitch, and as an option, to make an interesting attempt to relate speech synthesis to vocal exercises that a vocalist performs regularly. Another objective of the lab is for the student to learn the basics of spectral analysis using Discrete Fourier Transform.

2 Pre-Lab

2.1 Articulatory Apparatus – the Basics (Optional Reading)

Speech synthesis has been an interesting scientific and engineering endeavor for centuries. The challenge in making machines talk has captured numerous ingenious minds for a long time. Figure 1 is an illustration of a mechanical speech synthesizer designed by Wheatstone in the 19th Century. The "talking machine" consists of a bellows which blows air as a person's lungs do. A reed at a certain point along the air passage then vibrates as what would happen in a clarinet, causing some “essential sound” to be formed. (Think of a piece of paper that flaps back and forth in some gusty wind.) This “essential sound” does not bear particular characteristics that allow us to interpret the identities of the sound as components of the speech pronunciation. The "resonator" at the right of the figure is a flexible (configurable) tube made of leather. This tube can be manipulated by a skilled hand to form various configurations to produce desired speech sounds. As enlarged in Figure 2, key parts of the machine are thus similar to a musical instrument; one "learns" to play it by controlling its key mechanism, such as the holes in a flute. So, the heart of the "talking machine" is this leather tube.

One can also envision the tube as our vocal cavity, the so-called vocal tract. Indeed, when we speak we adjust our vocal tract to produce different speech sounds. The question is how the tube configuration is related to the speech identity.

Let's look at our articulatory apparatus. Figure 3 is a sagittal view of our main articulators, including the glottis, the vocal fold or vocal cords, the tongue, the vocal tract, the velum and the nasal passage, and finally the lips. We do not need to fully understand the detailed role each of these articulators play in producing a speech sound. For this lab, we shall focus on the synthesis of vowel-like sounds (a, e, I, o, u, …) and the most relevant articulatory features in this context are the shape of the vocal tract and the frequency at which the vocal cord vibrates.

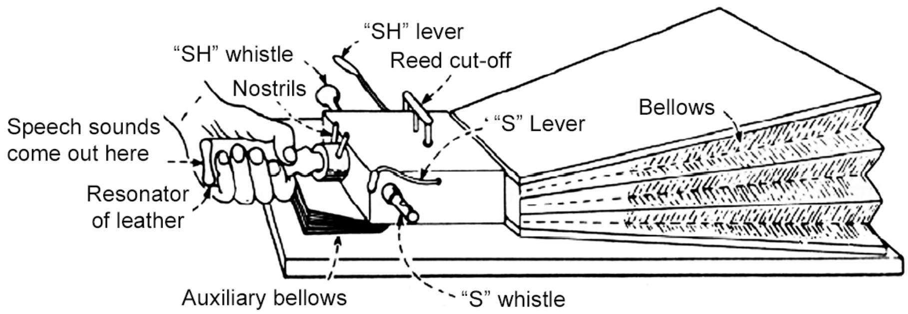


Figure 1 A mechanical talking machine by Wheatstone

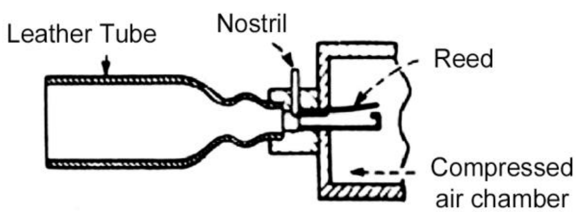


Figure 2 Enlargement of the leather tube and the reed in Wheatstone's talking machine

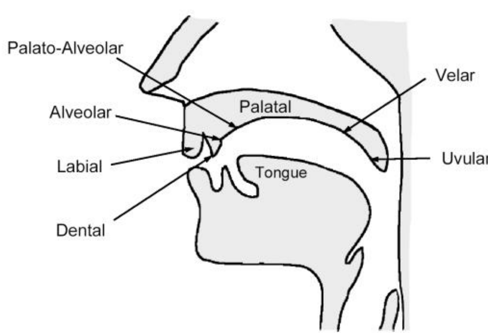


Figure 3 Human articulatory apparatus

The theory of acoustic tube and its resonance provides a rigorous analytical framework for us to find out the kind of sound a tube of a particular configuration will produce. Figure 4 is a model of the vocal tract as an acoustic tube. The air flow from the lung is forced through the narrow passage of the vocal cords. It generates what is called a Bernouli oscillation. Essentially, it works as follows. As the air speeds through the passage, it creates a vacuum (a low pressure area). (Try to blow air between two parallel pieces of paper and see what happens.) This temporary vacuum pulls the vocal cords together, causing a blockage to the air passage (the so-called glottal closure). When the air is stopped, the vacuum disappears and the muscles connected to the vocal cords then retract the vocal cords to an open position. The air blows through again and the process repeats – Bernouli oscillation thus occurs.

If we are to measure the air volume (or pressure) at the vocal tract side of the vocal cords, it will look like a periodic train of pulses, carrying no specific speech identity (except what is called the voicing attribute, signifying that the vocal cords have been vibrating). Figure 5 shows an example of the glottal pulses – pulses of air volume flowing through the glottis. There is an important feature embedded in the glottal wave though; it is the frequency of the pulses, which is related to the frequency of pulsating of the vocal cords. This frequency is called the “fundamental frequency” or the pitch frequency. Women with high pitch voices have high fundamental frequencies; Earl Jones, the actor, has a rather low fundamental frequency. A talker uses the muscle connected to the vocal cords and the lung pressure to produce the desired pitch. A vocalist sings a music note by adjusting the pitch to be in tune with the designated key.

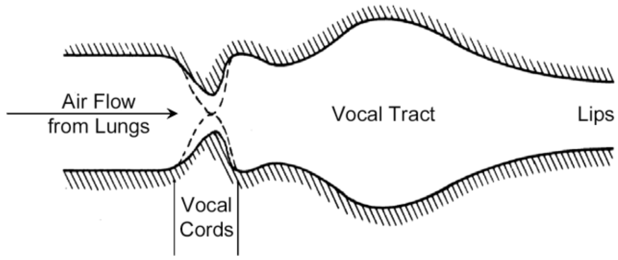


Figure 4 An acoustic tube model for speech production

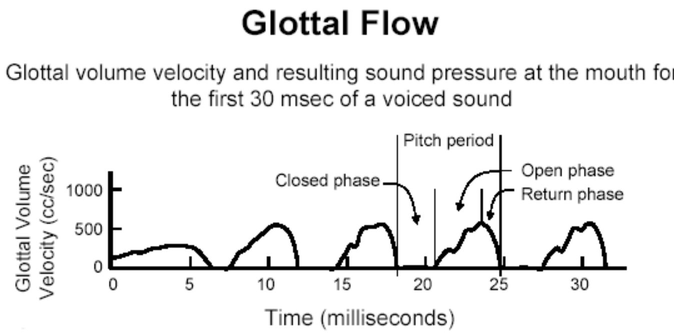


Figure 5 Glottal pulses

Beyond the vibrating vocal cords, it is the duty of the vocal tract to give speech identity to the acoustic signal. The duty is accomplished by varying the shape of the vocal tract.

Now, let's go back to the question of the relationship between the vocal tract configuration and the speech sound identity. Analysis of this relationship is often performed with a simplified model, called a multi-sectional concatenated uniform tube model of vocal tract. Figure 6 illustrates this model. In the figure, the vocal tract is conceived as a concatenation of a number of sections of uniform acoustic tubes. The glottis end is sometimes called the excitation end – that is where the driving air from the lung enters the vocal tract. The lips constitute the "radiation end" where the sound is emitted into the air, causing the air molecules to move.

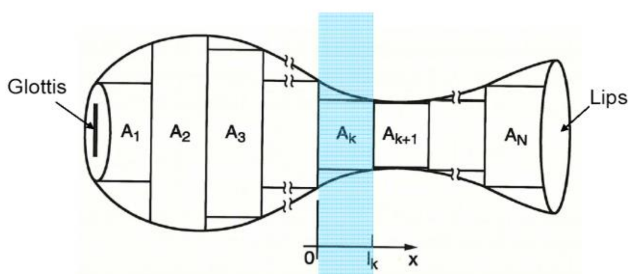


Figure 6 Multi-sectional concatenated uniform tube model

An acoustic tube is a resonator. What it means is that if an acoustic impulse (in the real world it probably does not exist but you can try to envision it as something like a sudden and very short blow of a huge amount of air) is given at the tiny opening of the glottis end, a few sinusoids will come out at the lips. These frequencies are considered the “modes,” or resonant frequencies, of the tube. When the sectional tubes are so-called “lossless”, meaning that the energy that is being put in the tubes does not dissipate or get absorbed by the walls of the tubes, these sinusoids will continue forever (again, only in an ideal world). Figure 7 illustrates several examples of simple acoustic tubes (maximum 2 uniform sections) and their resonant frequencies. At the top of the left column is a resonator which is a single uniform tube with length equal to 17.5 centimeters, which is considered the average length of human's vocal tract. Such a lossless uniform acoustic tube has resonant frequencies that are odd multiples of 500Hz as shown. The rest in the figure are 2-sectional tubes with various geometrical dimensions, characterized by the ratio of the lengths and the ratio of the cross-sectional areas. The set of resonant frequencies, or commonly known as the formants in speech signals, is noted in the figure next to the corresponding tubes. Modes, or formants, are the frequencies where most of the signal energy is concentrated.

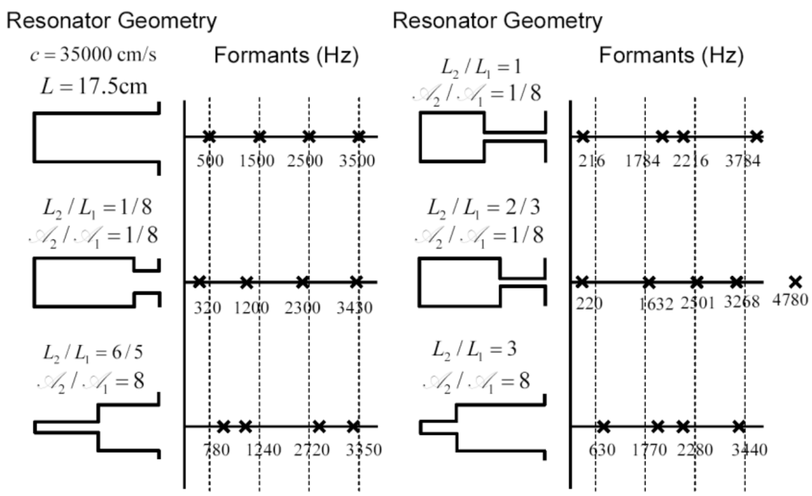


Figure 7 Example of 2-sectional acoustic tubes and their resonant frequencies.

2.2 Formant Frequencies of Vowel Sounds

Now, we get the idea that by changing the geometrical configuration of a tube, different sounds (with different distributions of energy spread over frequency) can be produced. When we speak, we move our jaws to open (enlarge the cross-sectional area of) or close (reduce the cross-sectional area of) the vocal tract (the acoustic tube). We also move our tongue to change the configuration of the vocal tract. These constitute the articulatory movement for producing the desired speech sound.

ARPABET Symbol	IPA Symbol	Typical Word	F1	F2	F3
IY	/i/	beet	270	2290	3010
IH	/I/	bit	390	1990	2550
EH	/ɛ/	bet	530	1840	2480
AE	/æ/	bat	660	1720	2410
AH	/Λ/	but	520	1190	2390
AA	/a/	hot	730	1090	2440
AO	/ɔ/	bought	570	840	2410
UH	/U/	foot	440	1020	2240
UW	/u/	boot	300	870	2240
ER	/ə/	bird	490	1350	1690

Table 1 Formants (Resonant frequencies) of various vowels

Table 1 summarizes the three formant frequencies present in a typical vowel sound. The presence of significant energy around a particular group of formant frequencies marks the identity of the vowel sound. Note that these are "typical" values. They do vary to some extent with people; after all, each person has his or her own voice characteristics, or "personality." Aside from these formant frequencies based on which we perceive the identity of a vowel sound, we may also observe a fourth formant at around 2800-3000 Hz, referred to as the "singing formant" when the vocal effort turns from speaking to singing. You may be interested in exploring further the acoustics of singing formant but it would be beyond the scope of this lab. We are mostly interested in synthesizing the sound with three sinusoids here. Nevertheless, you may ask: Will adding three sinusoids of different frequencies together produce a vowel sound? No, not yet. Again, the formants are the places along the frequency line that show a dominant presence of energy, resulted from the shape of the vocal tract. The role of the vocal cords is yet to be put to work.

In a commonly accepted model of the human articulatory apparatus, a sequence of glottal pulses (Fig. 5) which are produced by the vibrating vocal cords is regarded as the signal that drives the vocal tract which in turn changes the characteristics of the acoustic signal according to its configured shape. One may ask a simple but somewhat intriguing question at this point: Why do we have to drive the acoustic tube repetitively (with a sequence of glottal pulses)? One reason among perhaps many is that all natural resonators are dissipative and thus the signal will decay as time progresses. The vocal tract consists of soft tissue walls and is constantly damp with saliva. The acoustic energy will be quickly absorbed and the sound will not last very long. Therefore, the lung needs to keep releasing air into the glottis and the vocal tract if the speech sound is to be sustained. This train of repetitive driving pulses also gives the talker an important control factor in his or her speech act, namely, the pitch in the speech. We change the muscle tension around the vocal cords to alter the frequency of the Bernouli oscillation. Sometimes we do need to raise our pitch during speaking for additional communicational effects other than just the words.

In this lab, we'll not be constructing such an elaborate model of the articulatory apparatus. Instead, we shall emulate the dissipation and repetitive driving of the vocal tract by using what we call some epoch functions. An epoch function represents the envelope of a puff of air that goes into the vocal tract to excite one of its resonant modes (i.e., a formant frequency) and produces a component of the speech waveform at the lips. Figure 8 shows examples of the epoch function, taking the form of $p(t) = (at + c)e^{-bt},\ 0 \le t$:

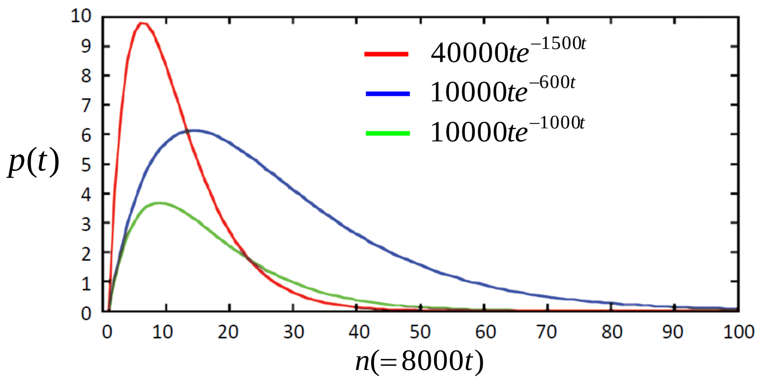


Figure 8 Three examples of the speech epoch function

An important consideration for implementing the epoch function is the proper choice of the values for the triplet (a, b, c). In this lab, we can assume $c = 0$. A general rule of thumb is that when applied to higher formants, constant a should be smaller and b should be larger as the dissipation of energy in the vocal tract is in general more severe for higher frequencies. A typical epoch function may be implemented as:

```
tt = 0:1/fs:0.015;            % an epoch rarely exceeds 0.015 ms
epf = aa*tt.*exp(-bb*tt);    % assuming cc = 0; define your own aa and bb
```

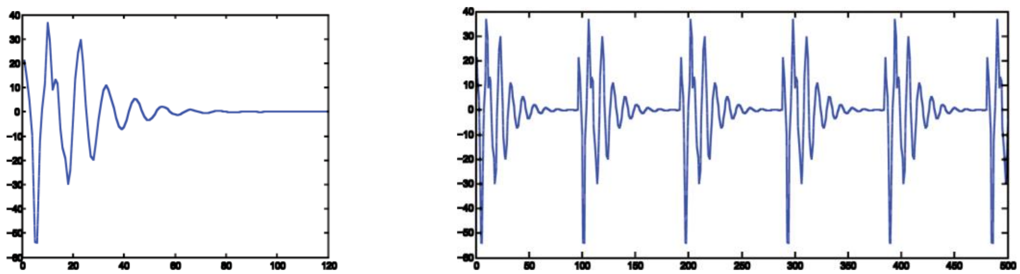
The examples in Figure 8 gave some indirect suggestions about the values of **aa** and **cc**. The sampling rate **fs** for speech is typically 8000 samples per second and the length of **epf** in the above example is thus 120 samples. Since we are going to synthesize the vowels based on the three formant frequencies in Table I, we need three such epoch functions, ready to be applied to the three sinusoids, respectively.

To synthesis one epoch of a vowel sound, you need to carry out the following steps:

- Look up the formant frequencies for the intended vowel sound from Table 1;

- Generate separately three sinusoids of the formant frequencies;
- Generate the three corresponding epoch functions and multiply the three pairs of functions separately (make sure you specify the correct length for the multiplying functions to avoid the problem of unmatched vector dimensions);
- Add the three resultant waveforms together to form one epoch of the intended vowel sound.

Figure 9(a) shows a single epoch of a vowel sound produced by adding three sinusoids together, each modulated by a corresponding epoch function. One epoch of vowel is too short for listening. Now, based on the synthesized epoch of waveform, we need to repeat it to sustain the sound. The rate of such a repetition determines the pitch of the voice sound that you hear. Figure 9(b) shows a synthetic vowel waveform produced by repeating the single epoch in Fig. 9(a) at a rate of 83.33 repetitions per second. So, this synthetic vowel is indeed a periodic signal with a fundamental frequency of 83.33 Hz, which is slightly above the note E2. Note that a single epoch may be longer than the fundamental period; in the example of Figure 9, the epoch length is 120 samples and the fundamental period is 96 samples (=8000/83.33). Therefore, when repeating the epochs, there is an overlap of 24 samples between successive epochs; their values have to be added together to produce the final result (Fig. 9(b)).



9(a) One pitch epoch of a vowel; 9(b) A sustained vowel sound with a fundamental frequency of 83Hz.

Figure 9 An example of a synthetic vowel.

3 Lab Exercise

3.1 Revisiting addition of sinusoids and playing with `spectrogram`

In Lab 3, you have come across an exercise that requires you to produce a signal that is a sum of several sinusoids, each of which has its own frequency, amplitude, and phase. If you still have the function **`add_sines`**, retrieve it and modify it to suit our need here. If not, it is just as easy to write a new one (now that you are more experienced after several weeks of labs). What we need is a function (**`threecos.m`**) that generates three sinusoids of different frequencies and phases without adding them. If we need to add them, we do it outside the function. A header of the function may look like:

```
function xcos = threecos( amps, freqs, phs, fs, dur )
%     generate three cosines
%     amps: vector of amplitude
%     freqs: vector of frequency in Hz
%     phs:  vector of phase
%     fs:   sampling rate; say 8000 samples per second
%     dur:  duration of the signal in seconds
%     xcos: output matrix xx(1:fs*dur,1:length(amps))
% complete the rest of the code yourself <<<<🡐====
```

Now, use a sampling rate (**`fs`**) of 8000 samples/s to generate a 2-second long signal that is the sum of three sinusoids, with frequency at 320, 800 and 2400Hz, respectively. Let's arbitrarily assume that their corresponding complex amplitudes are $2\angle 0.1\pi, 1\angle -0.3\pi,$ and $0.8\angle 0.2\pi$. In other words, use **`threecos`** to generate three separate sinusoids and then add them with a code line that looks like:

```
xx = xcos(:,1) + xcos(:,2) + xcos(:,3);
```

Find the fundamental frequency, the fundamental period in number of samples (**N0**), and plot exactly 4 periods of **xx**. (Learn to use the command **figure** to keep the plot from being overwritten by subsequent plots.)

Then, use **spectrogram** to analyze **xx**. You already have some experience with the command in previous labs. Here, we want to focus on the signal and use a particular arrangement with regard to the so-called data window. Here is the code with some variables left for you to experiment with:

```
win = ones(wlen,1);
% wlen is the length of the rectangular window
% you may want to try a few values of wlen, say between 100 (12.5ms)
% and 400 (50ms), and observe the difference in the results
spectrogram(xx,win,novlp,nfft,fs,'yaxis');
% novlp is the amount of overlap between successive data windows
% nfft is the size of the discrete Fourier Transform
```

For this exercise, we can keep it simple by setting **novlp=wlen/2**, i.e., maintaining a 50% overlap between successive data windows. Try several sets of values for **wlen** and **nfft**; pick one set, jot down what you have observed on the lab sheet and show your plots (both the 4-period waveform plot and the spectrogram) to the instructor for verification. (Note: What you learn in this exercise will enable you to answer the question Q7.1 in the Lab Exercise sheet. Pay attention to the possible relationship among **N0**, **wlen**, and **nfft**.)

Instructor's Verification (Lab Sheet)

3.2 The epoch function

In the Pre-Lab section, epoch functions were introduced as a way to simulate a puff of air turbulence to be modulated by the vocal tract to produce a component in an epoch of a vowel sound. An example of MATLAB codes to create a single epoch function of 120 samples long (i.e., 15 ms @ 8000 samples/s sampling rate) was given in Pre-Lab. Now, write a function (**threeepo.m**) to generate three separate epoch functions. The header may look like:

```
function epo = threeepos( as, bs, fs, dur )
%      generate three separate epoch functions
%      as: vector of the a parameter
%      bs: vector of the b parameter
%      fs:   sampling rate; say 8000 samples per second
%      dur:  duration of the signal in seconds (use 0.015 here)
%      epo: output matrix epo(1:fs*dur,1:length(as))
% complete the rest of the code yourself <<<<🡸====
```

You can try your own parameters for *a* and *b*, but here is a suggestion: for the first epoch function **epo(:,1)**, use (4000, 1000), i.e., *a*=4000, *b*=1000, the second, **epo(:,2)**, (2000, 800), and the third, **epo(:,3)**, (8000, 1200). Plot the epoch functions in one figure (learn to use **hold**) and show it to the instructor for verification.

Instructor's Verification (Lab Sheet)

3.3 Generating one epoch of vowel

In this section, we'll produce one epoch of a vowel sound. Pick one vowel to synthesize, say the /i/ sound in Table I. It has three formant frequencies at 270, 2290, and 3010Hz. Use **threecos** to generate these three sinusoids with **fs=**8000 and duration **dur=**0.015ms, same as those parameters that you have used for generating the epoch functions in 3.2. Individual phase can be arbitrary. Let's call these sinusoids, **ss1** (for the 1^{st} formant), **ss2** (for the 2^{nd} formant), and **ss3** (for the 3^{rd} formant), respectively. Then, multiply each sinusoid with the corresponding epoch function, respectively, and finally add the three results together to form an epoch of the /i/ vowel. Plot the synthesized vowel epoch and show it to the instructor for verification. For ease of reuse, you may want to put the above calls into a wrapper or a function that may look like:

```
function vepo = mkvepo( amps, freqs, phs, as, bs, fs, dur )
%      synthesize one single epoch of a vowel sound
%      amps: vector of amplitude
%      freqs: vector of frequency in Hz, the formants
%      phs:  vector of phase
%      as:   vector of a parameter
%      bs:   vector of b parameter
%      fs:   sampling rate; say 8000 samples per second
%      dur:  duration of the signal in seconds
%      vepo: output vector containing one epoch of the vowel sound
%
nformants = length(amps);     % should be 3 here
xcos = threecos( amps, freqs, phs, fs, dur );
epo = threeepos( as, bs, fs, dur );
epocos = xcos.*epo;
vepo = sum(epocos');  % make sure you are summing over "time"
end
```

Instructor's Verification (Lab Sheet)

3.4 Adding epochs of waveforms at prescribed intervals

At this point you have an individual epoch of a vowel sound (which is too short to hear). As explained, one single epoch of waveform is considered as the result of exciting the vocal tract with just one short release of air controlled by the vocal cords. Now, we will simulate results of sustained excitation. We assume that the vocal cords vibrate at 83.333 times per second; i.e., F_0=83.333. It means that in one second there will be 83.333 of these epochs of waveform coming out of the lips; in this context, we can also call this the epoch rate. So, the waveform epochs need to be stacked together in time. Figure 10 illustrates the "stacking" process. At 8000 samples/s, an epoch rate of 83.333 translates to the fact that the epochs are 96 samples ($= f_s / F_0$) apart.

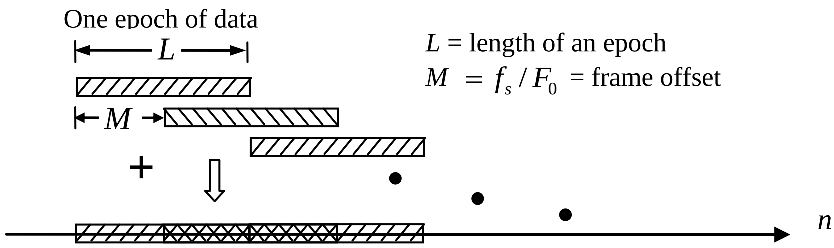


Figure 10   Overlapping and adding epochs together to produce the output

Following the illustration, write a MATLAB code to produce the desired output at the specified epoch rate of 96 (i.e., the fundamental frequency). Use the code to generate a 2s long output waveform. Play the result and show the spectrogram to the instructor. You may want to experiment with different fundamental frequencies to hear how the fundamental frequency changes the pitch of the speech sound.

Instructor's Verification (Lab Sheet)

3.5 Playing vowels together (Time Permits)

Follow the same procedure in 3.3 and 3.4 above and synthesize the five vowel sounds /a/, /ɛ/, /i/, /ɔ/, and /u/. This should be easy if you have included all the above functions in one code such that all you need is supply the formant frequencies and possibly the corresponding amplitudes. You can reuse the epoch functions if you'd like; or you can try to tweak the parameters for the epoch functions to achieve better vowel quality, however slight that may appear. Call these synthetic vowel sounds **ssa**, **sse**, **ssi**, **sso**, and **ssu**. Play all the vowels together by using

```
soundsc([ssa,sse,ssi,sso,ssu])
```

Are they recognizable to your ears?

Instructor Verification (Lab Sheet)

4 Optional Project: Singing the Scale and the Star Spangled Banner

With the results in the Exercise section, you can now try to synthesize a vocalized song. But, let's first do some practices: singing scales.

4.1 Singing the Scale (Vocal Warm-up)

Vocalists sing musical scales to warm up their voice and to keep accurate calibration of the notes with their voices. Do our voice synthesizers need to sing scales to warm up? Maybe. There are many musical scales that vocalists commonly practice on. For example, the basic "Major Scale" is made up of 7 basic notes, which are: Doh Re Mi Fa Sol La Ti, with another Doh (one octave higher) added at the end to help release the remnant pressure in the lung. Another example is the "Major Arpeggio scale," which is made of the notes: Doh, Mi, Sol, (high) Doh.

In this "vocal warm-up" section, we'll synthesize singing voices in several scales:

- Major scale: Doh Re Mi Fa Sol La Ti Doh (or equivalently, C D E F G A B C)
- Scale A: Doh Mi Sol Mi Doh (or equivalently, C E G E C)
- Scale B: Doh Mi Re Fa Sol Ti La Doh (or equivalently, C E D F G B A C)

The fundamental frequency of the synthesized vowel voice must match the frequency of the corresponding note. For example, C2 has a frequency of 65.41Hz which translates to approximately an offset of 122 samples (i.e., 8000/65.41) each time a vowel epoch is repeated (see Section 3.4 above). You need to find the corresponding F0 and the offset for each note in the scale. (Review Lab 4 and 5 to determine the note-frequency relationship, or you can just look over the note-frequency chart that can be downloaded from t-square.)

If you choose to do the optional project and manage to produce some results, send them to the TA who will be more than happy to share some comments with you.

4.2 Singing the Vowel Portion of the Star Spangled Banner

With your experience in translating a sequence of musical notes into a sequence of frequencies and durations (you have been able to use sinusoids to generate the song Chpsticks if you had carried out the song synthesis section there), you are ready to attempt at making a machine sing. Here, we'll try to work on the first ten words of the National Anthem "Oh, say can you see by the dawn's early light." We'll use the machine to sing the (approximate) vowel version of these ten words:

[AO, AO, EH, AA, UW, IY, AA, EH, AO, EH, IY, AA].

(In advanced courses, you'll learn also to synthesize the consonant parts. Here, we have to settle with the vocalized "vowel" portions.) The attached partial sheet music in Figure 11 contains the corresponding note designation. Equivalently, we can write the sequence of notes as:

G4(3/16)-E4(1/16)-C4(1/4)-E4(1/4)-G4(1/4)-C5(1/2)-E5(3/16)-D5(1/16)-C5(1/4)-E4(1/4)-F#4(1/4)-G4(3/8)

In this expression, each note is coupled with a duration specification in the parentheses. For example, C4(1/4) represents a quarter length C4 note. There are twelve notes corresponding to the eleven syllables. This should make it easier for those who are not familiar with sheet music and the notation.

For each note, you can find the frequency from previous labs or use the chart in **`NoteFreq.pdf`** which you can download from t-square. This frequency will serve as the fundamental frequency or the epoch rate for the particular portion of the synthesized output. You should review Section 3.3 to see how this is going to impact on the synthesis output. You should also experiment with note sequences that are one (or even two) octave lower or higher to experience the sensation of bass, tenor or soprano sounds (or mail vs. female). One immediate issue arises though – most keys do not correspond to an integer number of samples as the fundamental period and the frame offset f_s / F_0 is not an integer where the sampling rate is 8000 samples per second. At this point, we only have experience in adding frames of data together when the frames have an integer number of samples as offset (see Figure 10). Here we shall just round of the offset to the nearest whole number. (This rounding will result in off-key singing. Can you calculate the frequency deviation for each key?)

Write a MATLAB code to sing the (approximate) vowel version of first ten syllables of the Star Spangled Banner.

Figure 11 The first line of the Star Spangled Banner

A few worthy notes:

- As an extra, you can try to add the singing formant (an additional sinusoidal component) at around 3000 Hz and hear how the results change;
- It is a good idea to define the absolute duration that is corresponding to a unitary note, say a one-sixteenth note and write your code based on multiples of this unitary duration. In this way, you can easily change the pace (and duration) of the song.

Your final song(s) **should be sent to the grading TA in a `wav` file** together with a short report. (When you write the synthesis result to a **`wav`** file, make sure you normalize the result first to avoid clipping.) Add detailed comments as to how you accomplish the intended results as well as any additional insights you might have developed during exploration of the mentioned extra topics.

Lab #7

ECE-2026 Fall 2013

Lab Sheet

For each verification, be prepared to explain your answer and respond to other related questions that the lab TA's or professors might ask. Turn this page in at the beginning of your next lab.

Name: ___________________________________ Date of Lab:________________

Part 3.1 Answer the following questions for the signal **`xx`**.
What is the fundamental frequency? _______Hz. What is the fundamental period in terms of number of samples in **`xx`**? ______ samples.
Show the two plots (4 periods of **`xx`** and the overall spectrogram) to the instructor.

Verified: ___________________________ Date/Time:_________________

Part 3.2 Show the plot of epoch functions.

Verified: ___________________________ Date/Time:_________________

Part 3.3 Show the plot of one vowel epoch. Provide identity of the vowel and show the values of parameters used.

Verified: ___________________________ Date/Time:_________________

Part 3.4 Play the synthesized vowel sound and show the corresponding spectrogram.

Verified: ___________________________ Date/Time:_________________

Questions

Question 7.1 In Section 3.1, you generated a signal **`xx`** by summing three sinusoids together. You also have produced and showed a spectrogram to the instructor using a particular (rectangular) window to successively extract a block of data for analysis:

```
win = ones(wlen,1);
spectrogram(xx,win,novlp,nfft,fs,'yaxis');
```

By using proper values of **`wlen`** and **`nfft`** that match the properties of the signal, it is possible to obtain a spectrogram for this signal that looks "pure", with three horizontal lines, like the following:

This is probably very different from what you have been able to produce. Explain why this is possible and given the same code as above, what you need to do to generate this spectrogram (e.g., choosing the right window, setting the proper values of the calling arguments).