

Lesson 5: Exercises

Learning goals

Note. This lesson includes four sections and lots of little tasks, so feel free to focus on just a few topics according to your own interests and schedule. However, try to perform each step at least superficially. You don't have to report everything in the learning diary, but explain at least where you succeeded! If you run into problems, post a message in the discussion forum as usual.

A. Manipulating the fundamental frequency contour and speech rate of an existing sound with Praat

B. "Concatenative speech synthesis":

You try to find out if natural-sounding speech can be achieved by cutting out chunks of speech, pasting them back together in different order, and perhaps by applying the manipulation editor in Praat for the final touches. (Many of the current speech synthesis systems are actually based on pretty much the same type of technology!)

C. Try using Praat for setting up a so-called *Multiple Forced Choice* perception test (*ExperimentMFC* object type).

D. You develop your "literacy" in Praat scripts

... and you understand (at least vaguely) what a **variable** is and how variables can be used in scripts.

Exercises

A. Manipulating voice

1. Open an audio file in the Praat object. (Preferably, choose one where the speaker does not creak or whisper very much, for example, so that the pitch analysis can be more successful.)
2. Then press the **Manipulate - To Manipulation...** button in the dynamic menu. If necessary, provide more appropriate parameters for the pitch analysis. Press OK to create a new Manipulation object in the object list. You can now manipulate the sound by pressing Edit.
3. The familiar sound waveform is displayed at the top of the manipulation editor, and the blue vertical lines show the "pulses" (periods) detected by the pitch analysis. The middle pane shows the fundamental frequency curve calculated from the sound. At the bottom of the editor window you can see the usual playback bars for listening to the sound, but here the sound to be played is *resynthesized* by the *PSOLA* method by applying the pitch curve (shown in green), which can be edited:
4. The green dots can be moved by dragging them with the mouse (this can be a little tricky, but it does work). The curve formed between the dots represents the

"intonation contour" that you want to obtain for the resynthesized sound.

5. At first, there are too many dots and it would be too difficult to try and move every single one of them individually. To reduce the number of dots, you can stylize the curve: select **Stylize pitch...** from the **Pitch** menu and press OK. Then, only those adjacent dots whose pitch difference is more than 2 semitones will remain, and you will be able to create an audible change to the sound by moving just one of the dots.
6. Try dragging the dots up and down and/or slightly to the left and right and play the sound in between! If necessary, you can add more dots by selecting **Add pitch point at cursor** under the **Pitch** menu, if you want to modify the curve more precisely.
7. When you drag the dots around, the original fundamental frequency curve appears in gray in the background. Note that the smaller the changes you make in relation to the original pitch curve, the more natural the sound of the synthesis will sound. If the original pitch analysis is not successful in some portion of the sound, i.e. if no pulses were found even if the sound is audibly voiced, the sound will not be resynthesized at these points, and so the manipulation will not work for those regions.
8. At the bottom of the window, there is another white pane for manipulating duration, i.e., speech rate. At first, the duration pane does not contain any dots that could be modified in order to change the speech rate. Duration dots can be created by selecting **Add duration point at cursor** under the **Dur** menu. The light blue dashed line indicates the original rate of the sound. Where the curve formed by the duration dots is above the dashed line, speech will be slowed down in the resynthesis; and the parts of the curve below the dashed line will cause the speech to be accelerated. Give it a try! You can even manipulate durations and pitch simultaneously by changing both dotted curves with respect to the original.
9. If you want to save the sound you manipulated, select **Publish resynthesis** from the **File** menu first, and the resynthesized sound will appear in the Praat object list. Then, you can save the sound object in the normal way by selecting **Write to WAV file...** in the Object window.

Think about how such a manipulation tool could be used for research purposes.

B. "Concatenative speech synthesis"

1. Remember how the sound editor can be used to cut and paste snippets of a **Sound** object (**Cut**, **Copy**, and **Paste** under **Edit**).
2. By copying and pasting small clips from speech samples together, try to create a spoken utterance such as "Merry Christmas!", "Happy New Year!" or some other nice greeting that contains at least two words. You can use all the audio files in the course materials that you are already familiar with, or any other speech samples you have access to. (Please don't record the greetings directly yourself - that would be boring!)
3. As the starting point, you can take an existing sound object of suitable duration. First, turn the sound into silence (find out how to do this by using Praat's sound editor

commands). Then, you can start collecting pieces from other sound samples – you can view them simultaneously in other editor windows.

Alternatively, you can create a "blank" sound with the **Create Sound from formula...** command in the **New** menu. Just enter the number 0 in Formula, and you can obtain some complete silence! In this case, the sampling frequency (*Sampling frequency (Hz)*) must also be set to 22050 Hz so as to be compatible with the other audio files used in this course, because you cannot combine audio samples with different sample rates.

4. When cutting out pieces of the sound, be sure to select speech sounds or combinations of sounds whose beginning and end are as "clear" as possible, i.e., be careful not to include portions of adjacent sounds.
5. If the beginning or end of the area to be cut is at a point where the sound waveform curve is not close to zero (the horizontal light blue dashed line in the waveform image), you may end up hearing a clicking sound at the beginning or end of the clip, which can ruin your result. So try to select the portion to be cut so that the beginning and end are at zero. You can do this automatically in Praat: first select a region in the sound editor and then **Select: Move start / end of selection to the nearest zero crossing**. When you zoom in on the sound so that the selected region and its boundaries are accurately displayed, you will see that the boundaries move to the exact locations where the waveform intersects with the horizontal zero line. (Sometimes, however, the nearest zero crossing may be too far from the desired location. If that happens, you just need to try and strike a balance...)
6. Save the new audio file you created in WAV format.
7. The sound you composed from smaller pieces will probably sound a bit discontinuous and unnatural. You can try using the Praat Manipulation Editor to "improve" the sound and, if necessary, the durations / speech rate. You can make your wish as funny or fancy as you like! However, try to create a comprehensible result.
8. Be sure to transfer the audio you manipulated from the editor to a Praat object (**File: Publish resynthesis** in the manipulation editor) and finish by saving the audio file in WAV format.

During an online course, you can even post the audio file you created to the discussion forum in the Moodle course area, to amuse your fellow students!

Now for a bit more "serious" content:

C. Performing a perception experiment

1. Try to get an idea of what sort of experimental setups have been used to study issues related to speech perception. (You could do this by googling for "speech perception experiment", "categorical perception", "speech and hearing", "auditory perception", etc.) Note. Before taking the next "experiment", you should make sure that the volume level of the computer is appropriate (and that you can hear the sounds in

Praat).

2. Download the zip package [questiontest.zip](http://www.fon.hum.uva.nl/praat/questiontest.zip) to your computer. Unzip it to a suitable location on your computer. The question test directory contains the test.Emfc file as well as the subdirectory "sounds", which contains different, manipulated versions of the word "sika" ('pig' in Finnish). First, you can open the test.Emfc file in a text editor or for example Word, to examine its contents. The file contains an ExperimentMFC object that can be read by Praat. It describes a complete perception experiment in a specific way. Try to get some idea of the content of the file and the meaning of the different lines, for example with the help of Praat's internal manual (the corresponding manual page can also be found at <http://www.fon.hum.uva.nl/praat/manual/ExperimentMFC.html>).
3. Then open the *test.Emfc* file in Praat (you should probably close the Word document first). Run the test with the **Run** button. The idea is for you to decide for each sound stimulus whether the Finnish word for "pig" sounded like a question (i.e., like an interrogative) or not, and in each case you click on the appropriate box on the screen. From the upper left corner of the window, you can follow the progress of the experiment (there will be a total of 28 stimuli).
4. Once you have listened to all the stimuli and the experiment is over, you can close the *ExperimentMFC* window. Now press the **Extract results** button in the object window - *note that during real perception experiments, this must be done before the new subject starts performing the experiment!* A new *Results* object is created in the object list. You may want to convert this to a table with the **Collect to table** button, and you can save the new *Table* object to a text file with the **Save as tab-separated file...** (or **Save as comma-separated file...**) command on the Save menu. Since there would always be many subjects in real experiments, it is a good idea to name the Table file so that you remember who the subject was (you can use, for example, a unique code for each subject, etc.). The files can then be opened in Excel, for example, or with a statistical program that can be used to study the distribution of responses for different stimuli.

In fact, you could also perform various statistical analyzes directly from the Results object in Praat, but we will not practice those in this context. Of course you can try it yourself if you like.

5. Do you think you might be able to use such a tool for running experiments? If you think you'll be building your own perception experiments in the future, you might want to read the *ExperimentMFC* section of Praat's internal manual more carefully. The previous test was an identification test (identification test), but Praat can also perform discrimination tests (discrimination test) in which the subject is presented with two stimuli at a time and he/she is asked to tell whether the stimuli were similar or different.

D. Developing your skills for "reading" scripts

1. Open a new script window in Praat (select **Praat: New Praat script**). Type or copy the following lines into it:

```
form Calculator
  comment Give two numbers:
  real number_1
  real number_2
endform

summed = number_1 + number_2
appendInfoLine: "The sum of the numbers 'number_1' and 'number_2'
is 'summed'."
```

Run the above script, just to test it. The script uses three variables: `number_1`, `number_2`, and `summed`. Variables are sort of "lockers" that can be used for storing information during the execution of a script. The variables in this script are numeric variables that can only contain numeric values. The names of the variables can be chosen arbitrarily, but the name must not contain punctuation, special characters, or spaces, it must not begin with a capital letter, and it cannot be identical to any of the commands that are used in the scripting language.

Try to change the script in such a way that it calculates the difference of the given numbers (subtraction)! How can you make it perform multiplication or division? Test and check that you actually got the script to perform the calculation you wanted.

If you like, you can save the script from the **File** menu (**Save as...**) for later use.

2. Open another script window in Praat again (**New Praat script**) or clear the previous script window. Type or copy the following lines there:

```
# First, let's ask the user what day it is today:
form Weekday test
  sentence Weekday Tuesday
endform

# and then, a conditional expression:
if weekday$ = "Friday"
  appendInfoLine: "Have a good weekend!"
elseif weekday$ = "Sunday"
  appendInfoLine: "Enjoy today, because tomorrow you will need to
work again!"
else
  appendInfoLine: "It is 'weekday$' today."
endif
```

Run the script several times and see what the script does if you type different days of the week on the form dialog. This script uses only one variable called `weekday$`. It is of the variable type string, meaning that it can contain numbers as well as text. The names of string variables must always end in the dollar sign \$.

Try to figure out why variable names are sometimes typed inside 'single quotes' in this and the previous script!

3. Open a new script window in Praat or clear the previous one. Type or copy the following lines:

```
form Repetitions
  integer How_many_times 5
endform

echo Now we start repeating.

for rep to how_many_times
  appendInfoLine: "This is the rep number 'rep'!"
endfor

appendInfoLine: "The end!"
```

Run the script above several times and see what the script does. What is the significance of the *for - endfor* construction? This script utilizes two number variables. Can you figure out what they are?

4. Test and see what the following script does (when you select the subdirectory *questiontest/sounds* where the test sounds for the previous perception experiment are):

```
polku$ = chooseDirectory$ ("Locate a directory that contains sound
files with the extension .wav.")

Erase all

appendInfoLine: path$
Create Strings as file list: "filelist", "'path$'/*.wav"
files = Get number of strings

appendInfoLine: "Analyzing 'files' files."

for file to files
  select Strings filelist
  filename$ = Get string... 'file'
  Read from file... 'path$/'/filename$'
  To Pitch: 0, 75, 600
  # Draw a picture
  Draw: 0, 0, 75, 400, "no"
endfor
```

5. If you are particularly interested in Praat scripts, check out the *Scripting tutorial* pages in Praat's internal manual. The fastest way to access them is the **Help** menu in the Object window.
6. Finally, consider the things you have learned during the course. Would you like to learn more about something? Did you get new ideas for your own research or for further studies?

In the sixth lesson, we will test the ELAN program a bit and we will see how annotation techniques can affect the ways in which speech corpora can be analyzed.