Speech Communication

Autocorrelation and Fundamental Frequency

Fundamental Frequency

The autocorrelation function (ACF) is defined as:

$$r_{ss}(\tau) = E(s(k) \bullet s(k-\tau))$$

and can be estimated for a section of a signal with the length N through the following formula:

$$r_{ss}(\tau) = \frac{1}{N} \sum_{k=1+\tau}^{N} s(k) \bullet s(k-\tau)$$

For a periodic function with the period T the ACF has its maxima at the multiples of the period:

$$k \bullet T \text{ (with } k \in \mathbb{N}_0\text{)}.$$

Dealing with speech signals, these maxima can be interpreted as the period of the fundamental frequency.

E1 Computation of the autocorrelation function

Create the Matlab script prog_homework_01E1.m and load the "speech1.wav" file from the `pa1.zip` file.

Write the function

```
function r_ss = autoCorrelation(signal, maxTimeLag)
```

that computes the ACF according to the formula above and call it from prog_homework_01.m. `maxTimeLag` defines the maximum shift of the signal in samples, in our case please use 50ms (convert it to samples). The ACF should be computed for all time shifts from $\tau = 0$ until `maxTimeLag`. Don't use the build-in matlab function `xcorr()` to calculate the autocorrelation!
Plot `r_ss` and display the time shift τ on the x-axis (use the `linspace` command to create a vector for the x-axis) and the correlation r_{ss} on the y-axis.
Try to figure out which point in your plot indicates the fundamental frequency of the speaker.

E2 Determining the fundamental frequency

As explained, the maxima of the ACF can be interpreted as multiples of the fundamental frequency. Use the result of E1 (vector `r_ss`) to compute the fundamental frequency of the speech file.
What is the value of the mean fundamental frequency and what information can be drawn from it?

Hint: You will notice that the ACF starts with high values (for small shifts the signal correlates strongly with itself). However, we are only interested in the first "real maximum". Therefore, set the values in the ACF before the first zero crossing to zero.

E3 Pitch Contour

In the previous exercises we calculated the zero-crossing-rate. This allows us to assess whether a segment of a speech signal is voiced or unvoiced. From the detected voiced parts, we were able to compute the fundamental frequency of the speaker. This helps us to identify whether the speaker is male or female.
The fundamental frequency however is not constant and can strongly vary over time, depending on the speaking style. In this exercise, you should calculate the contour of the fundamental frequency based on the code from previous exercises.

Download the file `pa1.zip` from the ISIS page which contains a speech signal and some helpful code. Create a new script called `prog_homework_01E3.m`.

1. Then firstly divide the signal into windows with the `makeWin` function that you used in exercise 3 before. Use windows with a length of 25ms and 50% overlap.

2. Create a for loop that cycles through all windows (and thus columns) of the matrix you created and save the current window (e.g. `xWinI = xWin(:,i);`)

3. Now check whether the current window contains voiced or unvoiced speech. To do this use the voiced function that is contained in the zip file.

```
voi = voiced(x, Pth, Zth);
```

   The function returns `voi=1` if the speech window contains voiced speech. If the speech is unvoiced it returns `voi=0`. The function decides whether a window is voiced or not, depending on the power and the zero crossings contained in a window. It is already given in the zip file. However, you still need to add the calculation of the power to make it work. So, open the voiced.m and in line 4 "`Px = ???`" replace the question marks with the calculation of the power. This can be done with following formula:

$$P_x = \frac{1}{N}\sum_{k=1}^{N} x^2(k) \ , \qquad k \in \{1, 2, \dots, N\}$$

   Hint: in Matlab you can use the function `mean` to calculate this without loop.
   To start with, use these threshold values `Pth=0.0001;` and `Zth=0.06;`

4. If the window contains voiced speech, calculate the fundamental frequency for this window as you did in E2 before. Use a `maxTimeLag` of 25ms. In the ACF, ignore values for tau smaller than `tau=0.002`, because we assume that the speaker's pitch is lower than 500Hz.

5. Create a vector `pitch` that saves the pitch frequency for each window. If the window contains unvoiced speech save it as "not a number": `pitch(i) = nan;`

6. Finally plot the pitch together with the signal in a subplot and link the two subplots. This will make it possible to zoom into both subplots at the same time (of course don't forget to label the axis and the title):

```
figure
ax(1) = subplot(2,1,1);
plot(t, x)
ax(2) = subplot(2,1,2);
plot(tWin, pitch)
linkaxes(ax, 'x')
```

7. Now that you are able to plot the results, try around with the thresholds and choose values that work best for you.

8. As an example, your result for “speech1.wav” should look similar to this: