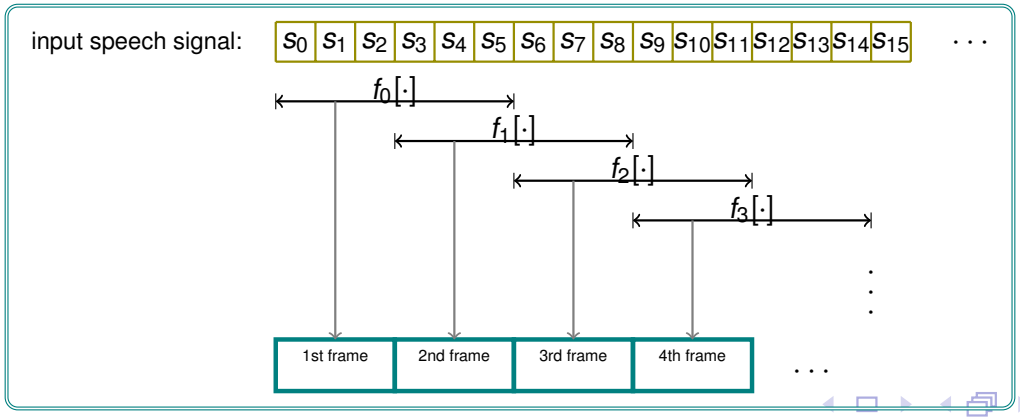
- - - Speech Processing - - -

- The previous procedures for *pre-processing* are realized according to one of the possibilities: *real-time* or *off-line*. For the former, a number of signal samples, corresponding to a certain time-interval of about 20 or 30ms, is stored and then processed in such a way that the process persists continuously. For the latter, it is possible to process either small signal frames of the same length or the whole signal, depending on the application and strategy chosen. In case a frame-by-frame processing is chosen, then a 50% overlap between consecutive frames, also called *windows* in this case, is usually adopted, as shown below.



- - - Speech Processing - - -

- For *feature extraction* and *classification*, the strategies to be adopted highly depend on the application. Thus, we will discuss those steps based on a connection with the intended task. Hereafter, we discuss **handcrafted feature extraction**.

- There is an enormous number of possibilities for features in the field of speech processing, being impossible for us to discuss all of them. Thus, we will comment on the most common ones:
  - **energy**: $E = \sum_{i=0}^{M-1} (s_i)^2$, where $s[n]$ is the M-sample long speech signal under analysis. E, which may be computed over the entire signal or over specific frames, refers to the capacity to perform work. It can be used to speech recognition, speaker (voice) recognition, emotion detection, pathology detection, and so on. Interesting ways to compute energy and use it can be found in paper "GUIDO, R.C. A Tutorial on Signal Energy and its Applications. *Neurocomputing*, v. 179, pp.264-282, (2016)".

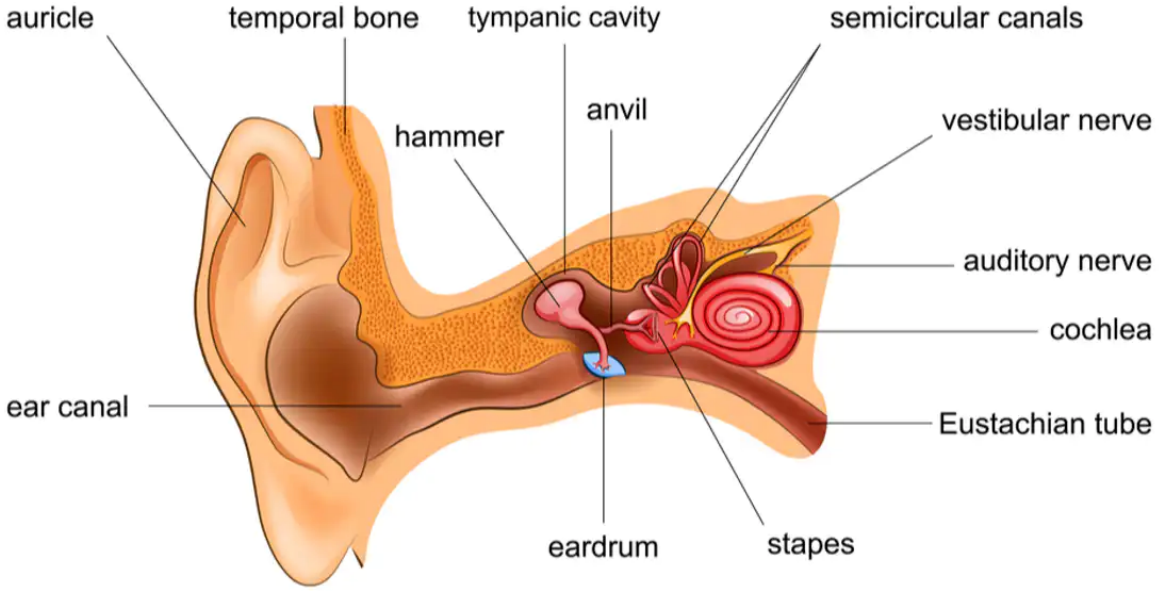
- - - Speech Processing - - -

- Example: Let $s[n] = \{1, 2, 3, 4, -4, -2, 2, 5\}$ be the input speech signal under analysis. Calculate its total energy. Calculate the energy of each rectangular window of length 4, assuming that each subsequent window overlaps the previous one in 50%.
- **zero-crossing rate (ZCR)**: It is defined as $\text{ZCR} = \frac{1}{2}\sum_{j=0}^{M-2} |sign(s_j) - sign(s_{j+1})|$, being $ZCR \geqslant 0$ and $sign(x) = \begin{cases} 1 & \text{if } x \geqslant 0; \\ -1 & \text{otherwise} \end{cases}$. It refers to the number of times a signal crosses the amplitude zero over time, somewhat reflecting the low spectral signal content. Interesting ways to compute ZCR and use it in speech analysis are detailed in paper "GUIDO, R.C. ZCR-aided Neurocomputing: a study with applications. *Knowledge-based Systems*, v. 105, pp. 248-269, (2016)".
- **voiced/unvoiced/noise decision**: to check whether a speech frame is voiced or not, the most simple approach consists of measuring its energy and its ZCR, as follows:

$$\begin{cases} \text{high } E \text{ and low ZCR } (\sim 1400 \text{ crossing per second}) & \rightarrow \text{the signal frame is voiced;} \\ \text{high } E \text{ and high ZCR } (\sim 4900 \text{ crossing per second}) & \rightarrow \text{the signal frame is unvoiced;} \\ \text{low } E \text{ and high ZCR} & \rightarrow \text{noise only} \end{cases}$$

- - - Speech Processing - - -

- **energy as a function of Bark scale**: human auditory system perceives frequencies by means of an important organ inside the ears: the cochlea, as shown below. Each cochlea's semicircular segment is responsible to react to one frequency band. Thus, features matching those frequency bands, simulating the way cochlea works, might be convenient. Interestingly, cochlea is subjected to masking!



Bark	(Hz)
0	0 - 100
1	100 - 200
2	200 - 300
3	300 - 400
4	400 - 510
5	510 - 630
6	630 - 770
7	770 - 920
8	920 - 1080
9	1080 - 1270
10	1270 - 1480
11	1480 - 1720
12	1720 - 2000
13	2000 - 2320
14	2320 - 2700
15	2700 - 3150
16	3150 - 3700
17	3700 - 4400
18	4400 - 5300
19	5300 - 6400
20	6400 - 7700
21	7700 - 9500
22	9500 - 12000
23	12000 - 15500
24	15500 - 22050

- - - Speech Processing - - -

- Procedure to use the Bark scale:
  - design 25 FIR filters, each one matching a Bark band;
  - pre-process the input signal under analysis: remove mean, normalize, pre-emphasis, Hamming/Hanning window;
  - filter the input signal by using, separately, each one of the 25 filters. This produces 25 filtered signals;
  - calculate the energy of each filtered signal, obtaining 25 scalars;
  - normalize the 25 scalars, creating the feature vector;
- **Today's Short Test (ST10)**: Consider the hypothetical speech signal segment $s[n] = \{1, 2, -3, 3, -2, 1, -1, -1, 4, 5, -5, 4\}$, sampled at 8000 samples per second. Assume that a sliding rectangular window $w[n]$ traverses it in order to extract features for inclusion in the feature vector $f[n]$, covering 0.25ms at each placement, with 50% overlap between consecutive windows. What is the length of $f[n]$? What are the values in $f[n]$, considering the raw energy as being the feature used? What if ZCR is considered instead of energy?

- - - Speech Processing - - -

- **Autocorrelation**: used in practice to find F_0, the autocorrelation is the correlation of a signal with itself. Given two signals, $a[n]$ and $b[n]$, their correlation is defined as $R_{a,b}[n] = a[n] \circ b[n] = \sum_k a_k b_{n+k}$.

  Practically speaking, the correlation between $a[n]$ and $b[n]$ is mathematically equivalent to the convolution of $a[n]$ with $b[-n]$.

  Example: assuming that $a[n] = \{1, 2, 3\}$ and $b[n] = \{4, 5\}$, find $R_{a,b}[n] = a[n] \circ b[n]$.

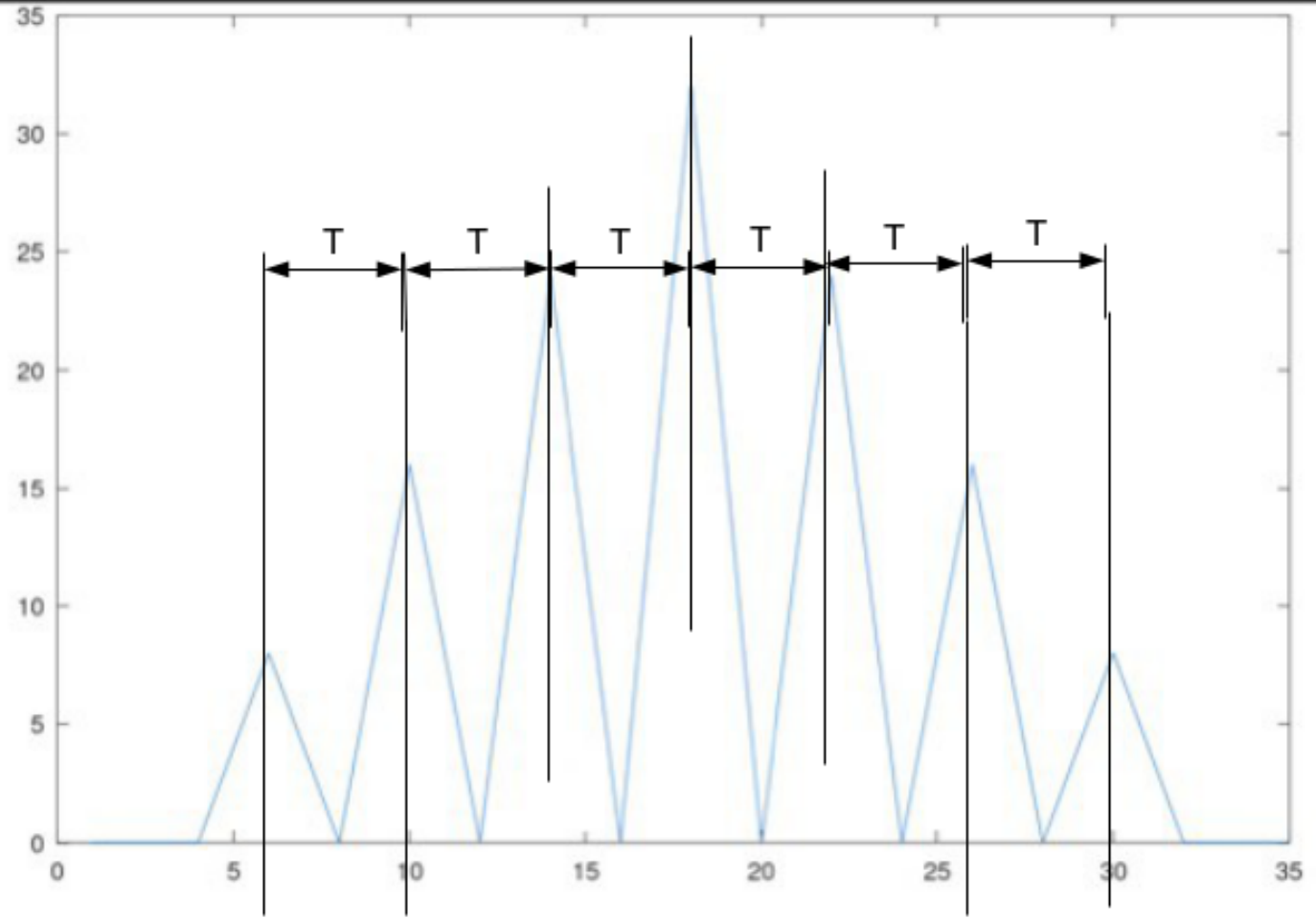
  Example: assuming that $a[n] = \{1, 2, 3\}$, find its autocorrelation signal, i.e., $R_{a,a}[n] = a[n] \circ a[n]$. Note that $R_{a,a}[n]$ is **symmetric**!

  The autocorrelation has maxima for the points where the input signal is periodic, with period T. Thus, the time interval between consecutive periods is the period of pitch, i.e., $F_0 = \frac{1}{T}$.

  Example: assuming that $a[n] = \{0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0\}$, find its autocorrelation, i.e., $R_{a,a}$, observing the periodicity found.

- - - Speech Processing - - -

```
plot(conv([0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0],[0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2 0 0]))
```



- - - Speech Processing - - -

- **Average Magnitude Difference Function (AMDF)**: also used in practice to find F_0, AMDF works similarly to the correlation, however, instead of multiplications we use differences, as follows: $AMDF[n] = a[n] \diamond b[n] = \sum_k |a_k - b_{n+k}|$.

  When $a[n] = b[n]$, AMDF has minima for values of k near multiples of F_0, instead of peaks in case of autocorrelation.

  Example: assuming that $a[n] = \{1, 2, 3\}$ and $b[n] = \{4, 5\}$, find $a[n] \diamond b[n]$.

  Example: assuming that $a[n] = \{1, 2, 3\}$, find its auto AMDF signal, i.e., $a[n] \diamond a[n]$.

  Example: assuming that $a[n] = \{0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0\}$, find its auto AMDF signal, i.e., $a[n] \diamond a[n]$, observing the periodicity found.

- - - Speech Processing - - -

```
plot([0 0 4 8 4 0 8 16 8 0 12 24 12 0 16 32 16 0 16 32 16 0 12 24 12 0 8 16 8 0 4 8 4 0 0 ])
```