Handout 9 – Linear predictive coding (LPC)

Exercise 9.1 – Computing LPC coefficients

a. Load the speech signal `zee.wav`.

```matlab
>> [x,fs] = wavread('zee.wav')
```

b. Plot the signal and listen to it.

```matlab
>> figure, plot(x), soundsc(x,fs)
```

c. Put short sections of the signal in columns of a matrix. This operation is called ‘buffering’. The first parameter is the sequence to buffer, the second parameter is the length of the short-term sections, and the third parameter is the overlap between sections. To avoid empty columns in the beginning of the buffer, specify the ‘nodelay’ flag.

```matlab
>> X = buffer(x,200,100,'nodelay')
```

d. Plot a few columns of the buffer `X`.

```matlab
>> figure, n=1:200, r=[0 400];
>> subplot(3,1,1),plot(n,X(:,21)),xlim(r),title('column 21')
>> subplot(3,1,2),plot(n+100,X(:,22)),xlim(r),title('column 22')
>> subplot(3,1,3),plot(n+200,X(:,23)),xlim(r),title('column 23')
```

e. Compute the 14th order LPC coefficients for the short-term signals in the buffer.

```matlab
>> ar = lpc(X,14);
```

The 14th order LPC coefficients of each short-term signal are stored in the rows of the matrix `ar`.

f. Display the LPC spectrum and the speech spectrum.

```matlab
>> figure, spec(X(:,21),fs,1024,'db')
>> hold on, spec(impz(1,ar(21,:),1024),fs,'db','r');
```

Exercise 9.2 – Computing the prediction error

Matlab’s `lpc` command does not directly return the prediction errors, but the predication errors can be found by filtering the short-term sections with the LPC coefficients in the feedforward mode.

a. To filter the short-term sections in the buffer with the LPC coefficients, we need to use a for-loop. The `for` command allows us to filter each column of the buffer with the corresponding LPC coefficients.

```matlab
>> e = zeros(size(X)); % pre-allocate the matrix e for efficiency
>> for i=1:size(X,2), % i=1,2,3,...,N
>>   e(:,i) = filter(ar(i,:),1,X(:,i));
>> end;
```
The output of this operation, the matrix $e$, contains the LPC prediction error for each short-term segment in its columns.

b. Plot the prediction error of one column of the buffer, with its spectrum.

```matlab
>> figure
>> subplot(1,2,1), plot(e(:,21))
>> subplot(1,2,2), spec(e(:,21),fs,1024,'db');
```

**Exercise 9.3 – Reconstructing speech**

We can reconstruct the speech signal from the LPC coefficients and the prediction error, or excitation signal, by filtering the prediction error with the LPC coefficients in the feedback mode.

a. To filter the excitation signals with the LPC coefficients, we need another for-loop, which is very similar to the previous for-loop.

```matlab
>> Y = zeros(size(e)); % pre-allocate the matrix Y for efficiency
>> for i=1:size(e,2), % i=1,2,3,...,N
   >> Y(:,i) = filter(1,ar(i,:),e(:,i));
>> end;
```

The output of this operation, the matrix $Y$, contains the reconstructed short-term signals in its columns.

b. Plot a few columns of the output to verify that they contain the reconstructed speech signals.

```matlab
>> figure, n=1:200, r=[0 400];
>> subplot(3,1,1), plot(n,Y(:,21)),xlim(r),title('column 21')
>> subplot(3,1,2), plot(n+100,Y(:,22)),xlim(r),title('column 22')
>> subplot(3,1,3), plot(n+200,Y(:,23)),xlim(r),title('column 23')
```

c. Now all that is left is that we need to undo the buffer operation to obtain a single signal. That is easiest with the function unbuffer (written by the instructor, not part of regular Matlab)

```matlab
>> y = unbuffer(Y,200,100); % undo the effect of buffering
```

d. Plot and listen to the reconstructed speech signal.

```matlab
>> figure, plot(y), soundsc(y,fs)
```

**Closing notes**

- The use of windows in the computation of the LPC coefficients of the short-term signals was omitted for clarity
- The ability to reconstruct a speech signal from its LPC coefficients and excitation signal opens the door to a number of possibilities:
  - Modify the speech signal by modifying the LPC coefficients
  - Compactly represent the speech signal by its LPC coefficients and a parametric description of the excitation signal (voiced/unvoiced, fundamental frequency, etc.)

Have a nice weekend!