ELEC-E5521 Speech and language processing methods L (5 cr)

- Idea: To familiarize the student with fundamental tools of speech processing using practical assignments based on real speech data.
- Three topics:
  - I: Linear prediction (Professor Paavo Alku)
  - II: Cepstrum, III: Language modelling (Professor Mikko Kurimo)
- Assignments are done in groups (2-3 students per group) using, e.g., MATLAB or other tools.
- Assignments and speech data needed in the assignments are available in MyCourses
- Returning the assignments: Each group books a session for each assignment. In this session, the team's returned assignment is discussed between the students and professor. Note: groups should not book the session before the assignment is properly done => If you have questions, contact the teacher by mail to get advice to finish the assignment before booking the session.
- Material: lecture slides

- No exam, but ALL assignments must be returned in spring 2022 (see the schedule below)
- Course prerequisite: fundamentals of speech processing and/or signal processing highly recommended
- **Schedule:**

24.1.2022:	1st lecture (topic: linear prediction, by Paavo Alku)
7.2.2022:	2nd lecture (topics: cepstrum, language modelling, by Mikko Kurimo)
25.1.2022 – 22.4.2022:	Working on the assignments, return by - 25.3 (1st assignment) - 22.4 (2nd and 3rd assignment)

Linear prediction (LP)

1. Introduction

- Linear prediction (LP) (also called Linear Predictive Coding, LPC), is a parametric spectral estimation method that is among the most widely used tools in speech processing [2,3].
- Examples of speech technology areas where LP can be used:
  - speech coding (to compress speech)
  - speech and speaker recognition (in feature extraction)
  - speech synthesis (to express speech waveforms in a parametric form and to synthesise speech from these parameters)
  - speech analysis (in formant extraction, pitch estimation etc.)
- As a spectral estimation methods, LP belongs to autoregressive (AR) modelling
- Basic idea: Prediction of a speech sample as a linear combination of previous samples (see Fig. 1).

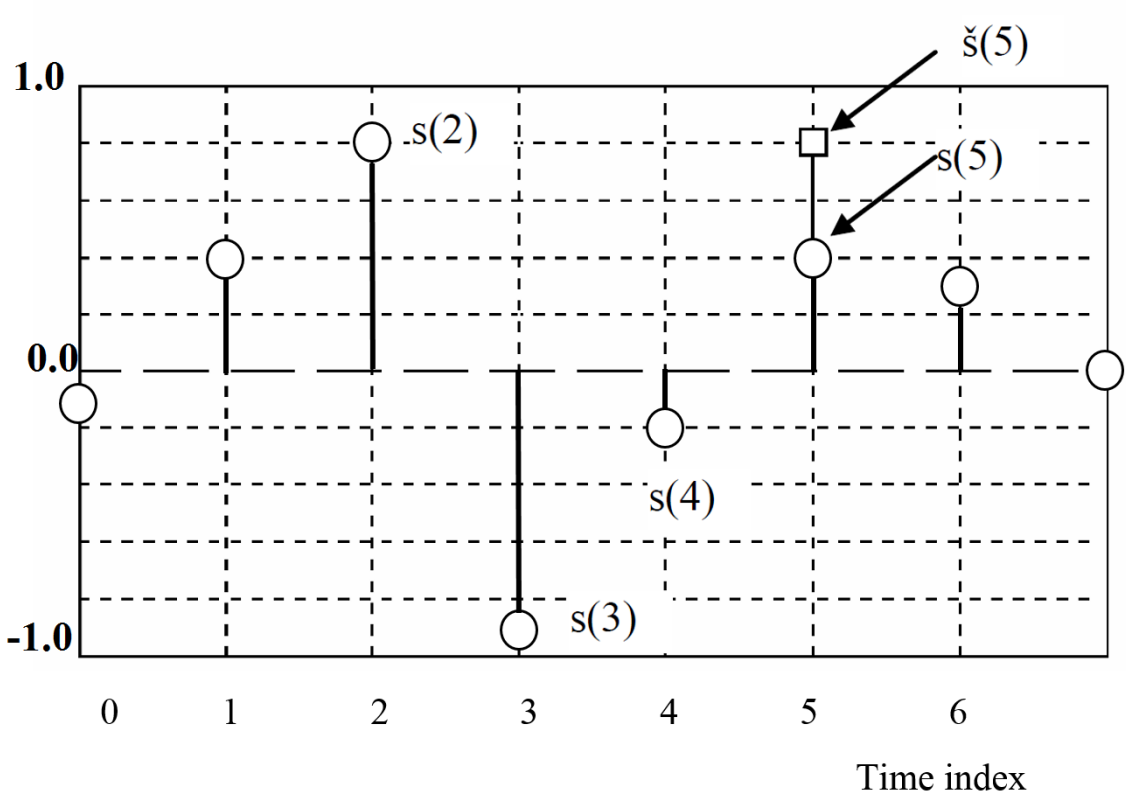


Fig. 1. Speech sample at instant n=5, s(5), is predicted from p=3 previous samples by multiplying these with coefficients a(1), a(2) and a(3). Prediction denoted by š(5).

$$\check{s}(5) = a(1)s(4) + a(2)s(3) + a(3)s(2)$$

- Prediction is repeated for several consecutive samples (i.e. for a frame of samples) by using the same set of prediction coefficients a(k), $1 \leq k \leq p$.
- The best set of coefficients is the one that yields the **minimum energy** of the prediction error.
- Energy-based optimization (i.e. mean squares error criterion) results in an optimization task that is mathematically easy to solve.
- Note: Prediction in LP takes advantage of three simple operations: addition, multiplication and delay, in other words, it uses FIR as a digital filter structure.

2. Filter optimization

- By using *p*th order LP, the signal prediction can be formulated as:

$$\check{s}(n) = \sum_{k=1}^{p} a(k)\, s(n-k)$$

- Prediction error, the residual, can be expressed:

$$e(n) = s(n) - \check{s}(n) = s(n) - \sum_{k=1}^{p} a(k)\, s(n-k)$$

- The energy of the residual can be expressed as:

$$E = \sum_n e^2(n) = \sum_n \left[s(n) - \sum_{k=1}^{p} a(k)s(n-k) \right]^2$$

$$= \sum_n \left\{ s^2(n) - 2s(n) \sum_{k=1}^{p} a(k)s(n-k) + \left[\sum_{k=1}^{p} a(k)s(n-k) \right]^2 \right\}$$

- In order to minimize the residual energy, let us set the partial derivatives of E to zero:

$$\frac{\partial E}{\partial a(i)} = 0, \qquad 1 \le i \le p$$

=>

$$\sum_n \left\{ -2s(n)s(n-i) + 2\sum_{k=1}^{p} a(k)s(n-k)s(n-i) \right\} = 0, \qquad 1 \le i \le p$$

=>

$$\sum_n s(n)s(n-i) = \sum_{k=1}^{p} a(k)s(n-k)s(n-i), \qquad 1 \le i \le p$$

$$\sum_{k=1}^{p} a(k)\phi(i,k) = \phi(i,0), \qquad \phi(i,k) = \sum_n s(n-i)s(n-k) \qquad 1 \le i \le p$$

- Equations above are known as the normal equations. They are a set of *p* equations with *p* unknowns (i.e. filter coefficients *a(k), 1≤k≤p)*.

- In equations above, the time span over which optimization is computed is left undefined.
- There are two LP variants which use different time spans for the optimization:
  - LP with the **autocorrelation criterion**: error minimization is computed in principle over $-\infty < n < \infty$
  - LP with the covariance criterion: minimization is computed over $0 \leq n \leq N-1$where N denotes the frame size
- Autocorrelation criterion is by far the most widely used variant of LP because (1) it results in normal equations that can be solved with fast algorithms and (2) the resulting optimal FIR (LP inverse filter) is guaranteed to be **minimum phase** (i.e. its roots are always inside the unit circle in the z-domain which implies that the inverse of the FIR is a stable all-pole filter). In the following, we will only discuss LP with the autocorrelation criterion.

- Infinite time span cannot be treated in real-life applications. Therefore, infinite time duration is changed to finite by assuming that signal is zero outside the frame (i.e. speech waveform is **windowed** into frames of finite length and the waveform is assumed to be zero outside the frame).

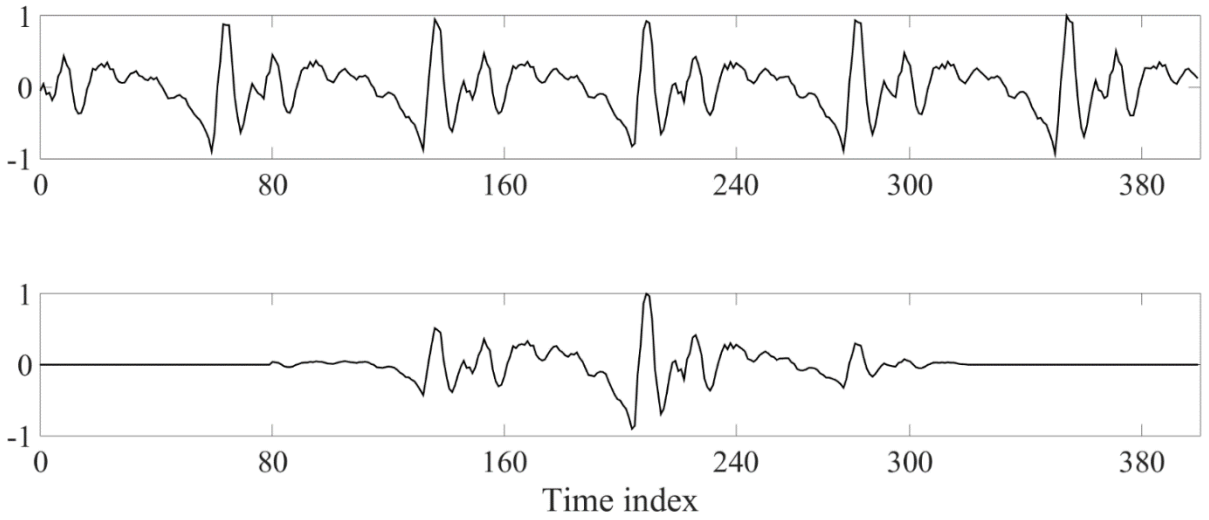


Fig. 2. Signal frame (bottom) windowed from continuous speech (top). Hamming window, frame position $80 \leq n \leq 320-1$.

- If frame length is *N* samples, the residual of a *p*th order LP is non-zero over $0 \le n \le N-1+p$. Therefore, we can write the normal equations as follows in matrix notation:

$$\phi(i,k) = R(i-k) = R(k-i) = \sum_{j=0}^{N-1-(i-k)} s(j)s(j+i-k)$$

$$\mathbf{R} \cdot \mathbf{A} = \mathbf{R}' \quad \Rightarrow \quad \mathbf{A} = \mathbf{R}^{-1} \cdot \mathbf{R}'$$

$$\mathbf{R} = \begin{bmatrix} R(0) & R(1) & \dots & R(p-1) \\ R(1) & R(0) & \dots & R(p-2) \\ . & . & \dots & . \\ R(p-1) & R(p-2) & \dots & R(0) \end{bmatrix}$$

$\mathbf{A} = (a(1), a(2). . . a(p))^T$

$\mathbf{R}' = (R(1), R(2). . . R(p))^T$

Summary of the main parts of LP analysis based on the autocorrelation criterion:

1. Select the order of prediction (p). As rule of thumb, p should be selected as the sampling frequency in kHz added by a small integer [3]. For example, if signal is sampled with 8 kHz, a proper prediction order is p=10 or p=12.
2. Select a value for the frame length (N) and the window type. A good choice for N is a value that guarantees that there are a few fundamental cycles of voiced speech within the frame, yet not too many in order to keep the signal stationary. Hence, a typical window length is 20-30 ms which corresponds to 160-240 samples when fs=8 kHz. Windowing is recommended to be done with windows such as Hamming or Hann (i.e. not rectangular).
3. Window the signal to get N windowed samples into the frame.
4. Compute autocorrelation terms R(0)-R(p).
5. Solve matrix **A**. Due to the fact that the autocorrelation matrix in **R** is symmetric (= columns correspond to rows) and Toepliz (= same element in diagonals), matrix inversion can be computed with the Levison-Durbin recursion which is computationally efficient.
6. Form a digital FIR filter from the solved coefficients:

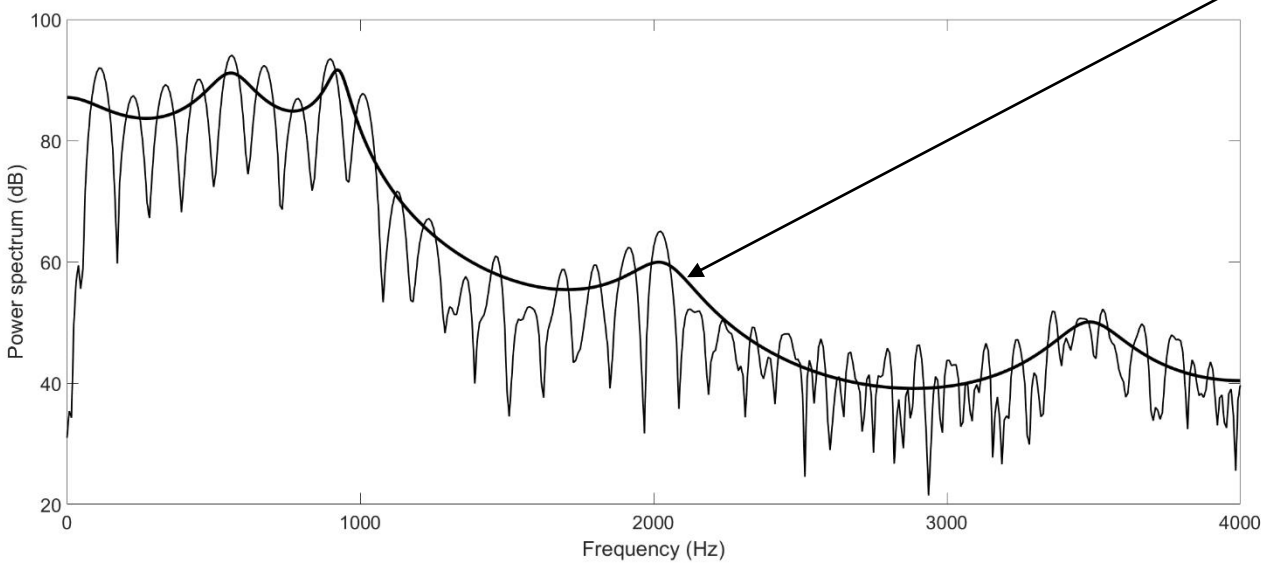
$$A(z) = 1 - \sum_{k=1}^{p} a(k) z^{-k}$$

7. Filter the original speech frame with A(z) to obtain the residual

3. LP filtering

- LP can be considered to consist of two parts:
  (1) LP-analysis: separation of a speech frame into a filter (FIR, LP-inverse filter) and time-domain residual
  (2) LP-synthesis: synthesis of the speech signal from the residual and the LP-filter (IIR, all-pole)
- Since residual is noise-like and spectrally flat, LP-filter forms an all-pole match to the speech spectrum

	LP-analysis:	LP-synthesis:
Time domain:	$e(n) = s(n) - \sum_{k=1}^{p} a(k)s(n-k)$	$s(n) = e(n) + \sum_{k=1}^{p} a(k)s(n-k)$
z-domain:	$E(z) = S(z) \cdot A(z)$	$S(z) = E(z) \cdot \frac{1}{A(z)}$



$$\frac{\sigma}{A(z)} = \frac{\sigma}{1 - \sum_{k=1}^{p} a(k) z^{-k}}$$

Fig. 3. Spectra computed from a vowel sound ([a]):
FFT-spectrum (thin) and LP-spectrum (thick) (p=10).

Observe that in Fig. 3 the levels of the two spectra match nicely because the energy of the LP filter has been adjusted to be equal to the energy of the (windowed) speech frame. It can be shown [3] that this is achieved, for example, by defining the gain (σ) of the LP filter as follows

$$H(z) = \frac{\sigma}{A(z)}$$, where σ^2 is the residual energy

- By increasing the prediction order, the spectral modelling performance of LP improves
- A simple method to quantify LP performance is to compare the energy ratio, called prediction gain, between the residual and the original speech signal

$$M = \frac{E_{res}}{E_{sig}} = \frac{\sum_n e^2(n)}{\sum_n s^2(n)}$$

4. LP information

4.1 General

- Most speech technology applications of LP take advantage of the LP filter (e.g. compression, feature extraction, formant trajectory modelling etc.).
- LP filter (i.e. *p*th-order all-pole type of IIR filter) contains the essence of LP analysis and the data needed to define this filter are called LP information.
- In filter optimization, LP information is expressed in the form of *p* filter coefficients (a(k), $1 \leq k \leq p$) of a direct form II filter structure.
- In many applications, LP information needs to be processed (i.e. quantized, predicted using, e.g., DNNs etc.) which changes the computed coefficients.

  =>

  LP synthesis filter, which is guaranteed to be stable in its **original form** when computed with the autocorrelation criterion, might become **unstable.**

- In addition to direct form II filter taps, there are alternative ways to express the LP information:

  - reflection coefficients (PARCOR, Partial Correlation)
  - logarithmic area ratios (LAR)
  - line spectrum pair (LSP)

4.2 Reflection coefficient

- Solving the normal equations (section 2) with the Levinson-Durbin recursion can be expressed:

$$E^{(0)} = R(0)$$

$$k_i = \left\{ R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(|i-j|) \right\} \Big/ E^{(i-1)}$$

$$\alpha_i^{(i)} = k_i$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \qquad 1 \le j \le i-1$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)}$$

- The set of equations above (except the first line) are repeated recursively with i = 1, 2,....*p* resulting in LP coefficients:

$$a(i) = \alpha_i^{(p)}, 1 \le i \le p$$

- As by-product, the recursion yields reflection coefficients

  $k_i, 1 \leq i \leq p$

- Reflection coefficients enable presenting the LP inverse filter in lattice form [4]. Fig. 2 shows two 3rd order LP inverse filters, one in direct form structure (upper) and the other one (lower) in lattice structure.

- Using reflection coefficients is motivated by the following fact:

  A(z) is minimum phase

  <=>

  all *p* reflection coefficients are between -1.0 and 1.0

- The property above helps in checking stability of 1/A(z) when LP information is quantized.

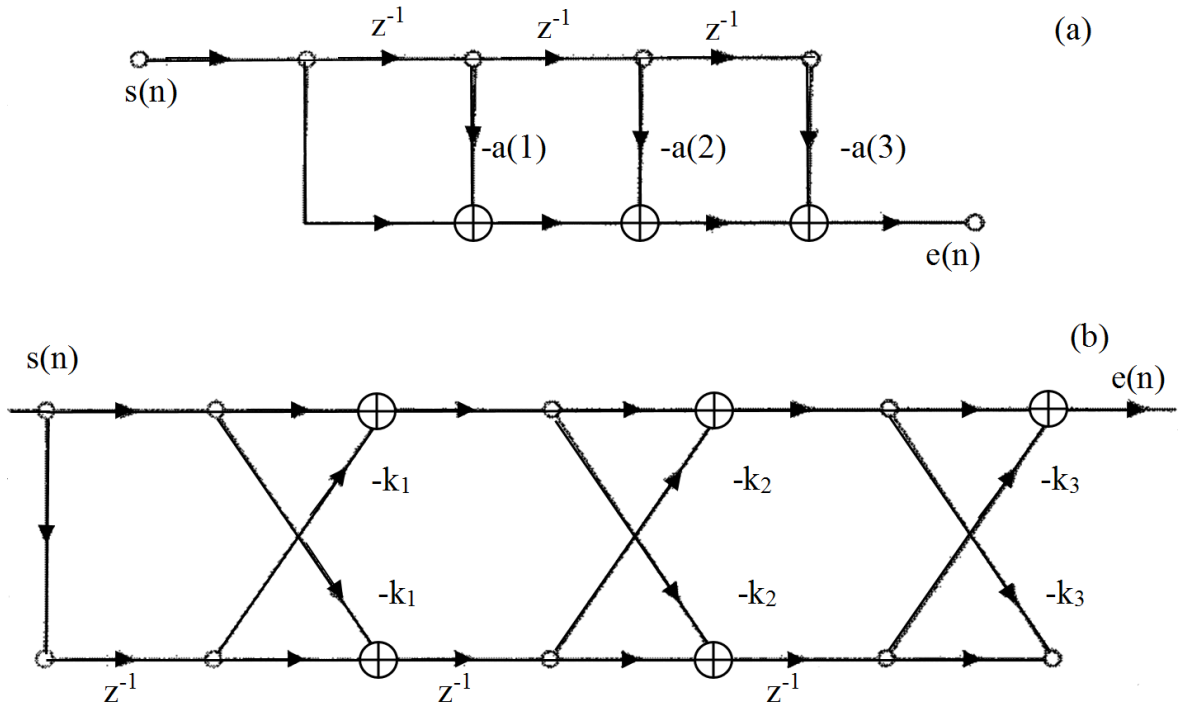


Fig. 4. Third order (p=3) LP inverse filter in (a) direct form II structure and in (b) lattice structure.

4.3 LSP decomposition

- LSP decomposition (Line Spectral Pair, also called Line Spectral Frequency, LSF) transforms a *p*th-order LP inverse filter (polynomial A(z)) into two polynomials P(z) and Q(z) [5]:

$$P(z) = A(z) + z^{-(p+1)} A(z^{-1})$$

$$Q(z) = A(z) - z^{-(p+1)} A(z^{-1})$$

- P(z) and Q(z) are LSP polynomials with the following properties:
  - order of P(z) and Q(z) equals p+1
  - P(z) is symmetric (even), Q(z) is antisymmetric (odd)
- The original LP inverse filter can be obtained by summing P(z) and Q(z):

$$A(z) = \frac{1}{2}(P(z) + Q(z))$$

- P(z) and Q(z) have fixed roots in the z-domain, known as trivial roots, irrespectively of the values of the LP coefficients.

- Trivial roots appear as follows:
  (1) even values of *p*: trivial root of P(z) at z=-1, trivial root of Q(z) at z=1
  (2) odd values of *p*: no trivial roots in P(z), two trivial roots of Q(z)
  at z=1 and z=-1

- In addition to the properties above, LSP has the following features which make its use in, for example, quantization of LP information highly justified [1,6]:

  (1) Roots of P(z):n and Q(z):n are always on the unit circle in the z-domain. These roots are called LSFs.

  (2) Roots of P(z) and Q(z) interlace on the unit circle
  <=>
  A(z) is minimum phase

- Property 2) implies that when LSFs change from their original positions due to, for example, quantization, the stability of the corresponding LP synthesis filter can be guaranteed by ensuring that the processed LSFs interlace on the unit circle.

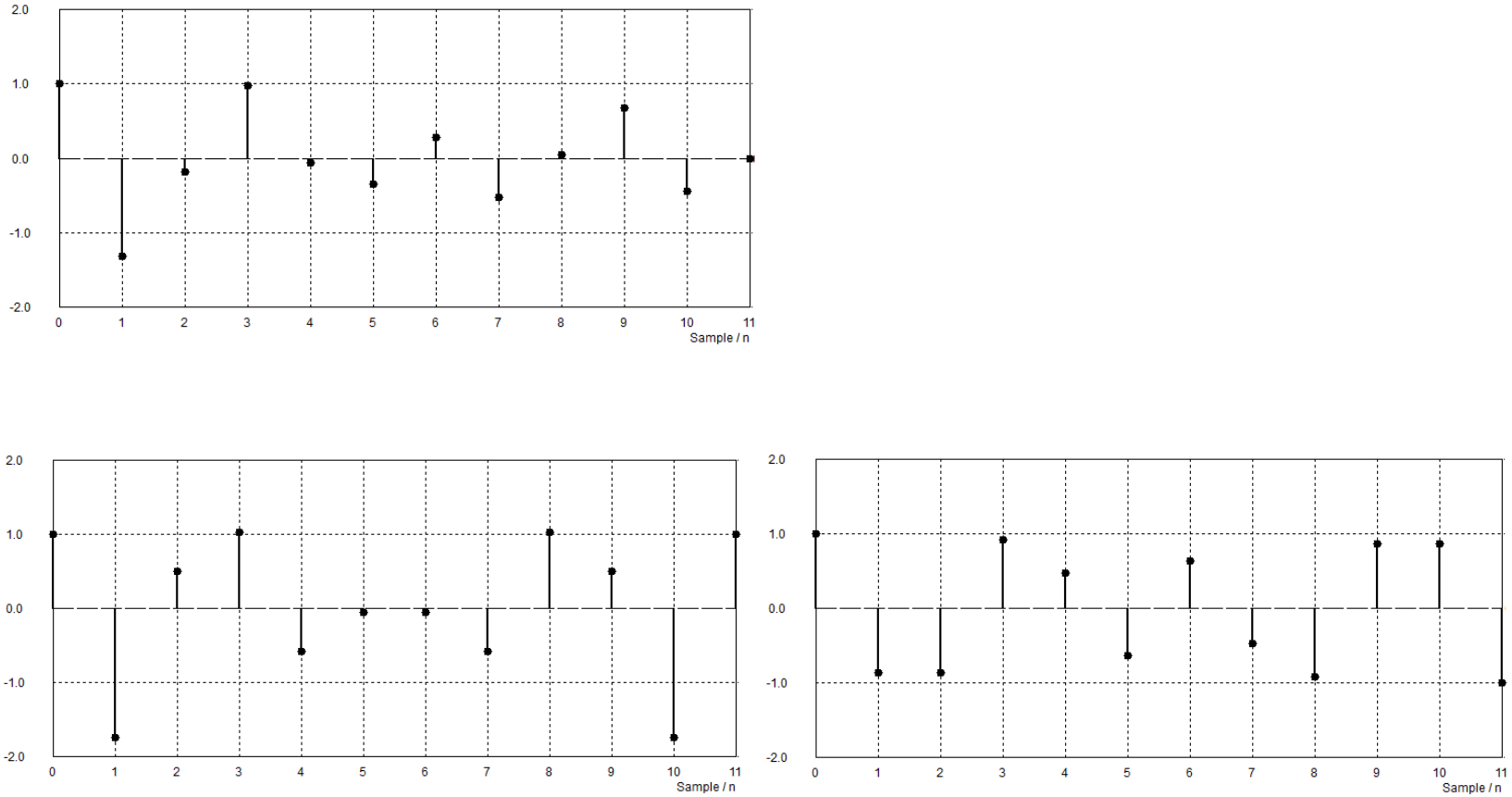


Fig. 5. Examples of impulse responses. Top: impulse response of A(z) (p=10), bottom left: impulse response of P(z), bottom right: impulse response of Q(z).

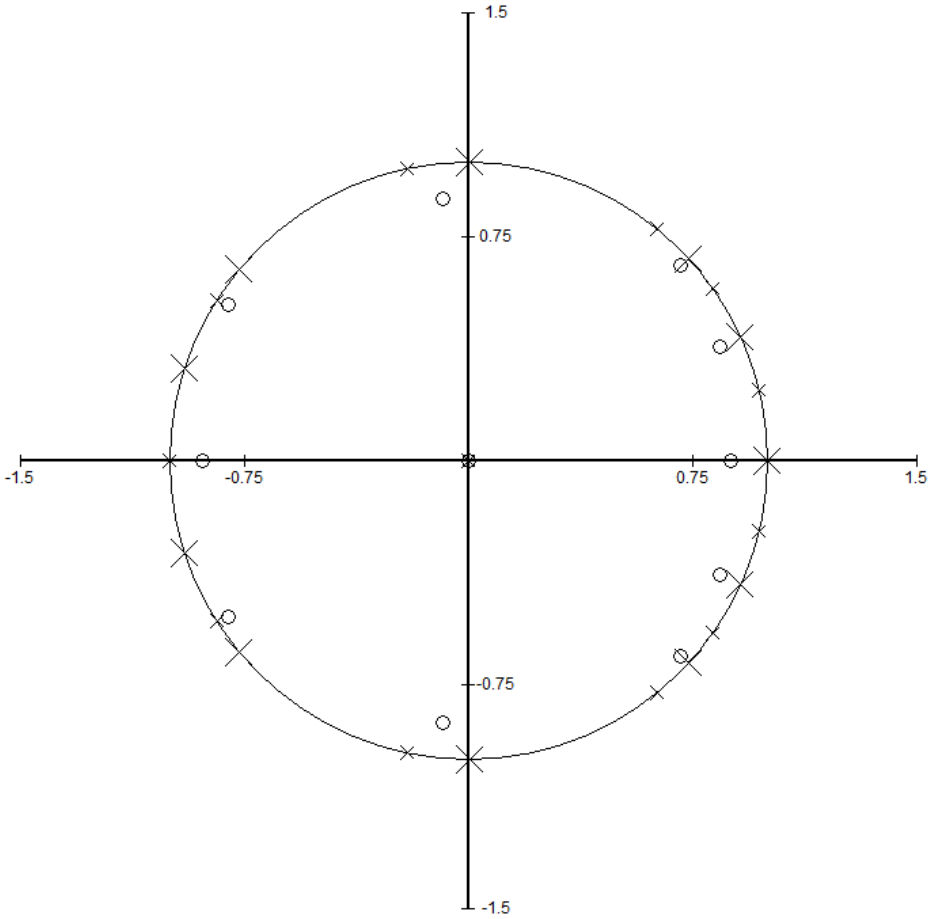


Fig. 6. LSP in the z-domain (p=10). Roots of A(z): circles, roots of P(z): small crosses, roots of Q(z): large crosses.

References:

[1] Kleijn, W., Paliwal, K. (Eds.), Speech Coding and Synthesis, Elsevier, 1995.

[2] Makhoul, J., Linear Prediction: A tutorial review, Proc. IEEE, Vol. 63, pp. 561-580, 1975.

[3] Markel, J.D., Gray, A.H., Jr., Linear Prediction of Speech, Springer-Verlag, 1976.

[4] Oppenheim, A.V., Schafer, R.W., Discrete-time Signal Processing, Prentice-Hall, 1989.

[5] Soong, F.K., Juang, B-H., Line spectrum pair (LSP) and speech data compression, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc., paper 1.10, pp. 1-4, 1984.

[6] Viswanathan, R., Makhoul, J., Quantization properties of transmission parameters in linear predictive systems, IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. 23, No. 3, pp. 309-321, 1975.