

Sprachverarbeitung: Übung 5

LPC-Analyse und -Synthese von Sprachsignalen

Die lineare Prädiktion wird in der Sprachverarbeitung für viele verschiedene Zwecke eingesetzt. Dem ursprünglichen Anwendungsbereich entsprechend hat sich das Akronym LPC (linear predictive coding) durchgesetzt, wobei die Codierung gewöhnlich als LPC-Analyse und die Decodierung (Signalrekonstruktion) als LPC-Synthese bezeichnet werden. In dieser Übung ist je eine Matlab-Funktion für die LPC-Analyse und die LPC-Synthese zu verwirklichen.

Aufgabe 1: LPC-Analyse

Schreiben Sie eine Funktion `[G,mx,P,A] = LPC_analysis(s,K,Pmin,Pmax,vthr)`, welche für einen Sprachsignalabschnitt `s` die Parameter des LPC-Sprachproduktions-Modells bestimmt. Gemäss Abbildung 4.19 im Buch sind dies der Vektor `A` mit den Koeffizienten a_0, a_1, \dots, a_K des Synthesefilters

$$H(z) = \frac{1}{A(z)} = \frac{1}{a_0 + a_1 z^{-1} + \dots + a_K z^{-K}},$$

die Länge der Signalperiode `P` (Anzahl Abtastintervalle) und der Verstärkungsfaktor `G`. Als weiteres Ausgabeargument ist hier `mx`, das Maximum der normierten Autokorrelationsfunktion vorhanden (siehe weiter unten).

Zur Bestimmung der a_k ist der Signalabschnitt zuerst mit einer Hamming-Fensterfunktion zu multiplizieren und dann die Matlab-Funktion `lpc` anzuwenden.

Die Periodizität des Signals kann aus der normierten Autokorrelationsfunktion (diese Funktion müssen Sie selbst programmieren) bestimmt werden, wobei die Höhe des Maximums (soll als Argument `mx` ausgegeben werden) angibt, wie ausgeprägt die Periodizität des Signalabschnittes ist (vergl. Abbildung 4.12 im Buch). Anhand des Schwellwertes `vthr` kann entschieden werden, ob der Signalabschnitt als stimmhaft (Maximum der normierten AKF ist höher als die Schwelle) oder als stimmlos zu behandeln ist. Der Index des Maximums gibt die Länge der Periode an. Vorteilhafterweise wird das Maximum nur in einem beschränkten Indexbereich der AKF gesucht, nämlich im Bereich `Pmin` bis `Pmax`. Für stimmlose Signalabschnitte (Maximum der normierten AKF ist kleiner als die Schwelle `vthr`) wird die Periode `P` auf null gesetzt.

Sie können die realisierte Funktion mit dem Matlab-Programm `ueb5_1` testen. Dieses Programm lädt ein Sprachsignal in die Variable `sig`, ruft pro Analyseabschnitt `s` die LPC-Analyse-Funktion auf und speichert die berechneten LPC-Parameter in der Matrix `lpc_data`. Zusätzlich werden das Signal, die AKF-Maxima, die Entscheidungsschwelle (punktiert) und der Grundfrequenzverlauf graphisch dargestellt.

Aufgabe 2: LPC-Synthese

Schreiben Sie eine Matlab-Funktion `[so,do,zo] = LPC_synthesis(G,P,di,A,ns,zi)`, welche das “Gegenstück” zur Funktion `LPC_analysis` der Aufgabe 1 darstellt, also aus den LPC-Parametern `G`, `P` und `A` einen `ns` Abtastwerte langen Abschnitt `so` des Signals rekonstruiert. Diese Funktion muss folgendes ausführen:

- Einen stimmhaften oder stimmlosen, `ns` Abtastwerte langen Abschnitt des Eingangssignals für das Synthesefilter generieren (Anregungssignal), je nachdem, ob die Eingangsgrösse `P` (Periodenlänge) grösser oder gleich null ist. Als stimmhaftes Signal wird eine periodische Impulsfolge verwendet, als stimmloses ein weisses Rauschen (kann mit der Funktion `rand` erzeugt werden). Das Anregungssignal soll im Mittel eine Leistung von 1 haben, damit die Multiplikation mit dem Verstärkungsfaktor aus der Analyse wieder dieselbe Signalleistung ergibt, die beim Prädiktionsfehler gemessen worden ist.
- Beachten Sie, dass immer zuerst die letzte Periode des vorausgehenden Syntheseabschnittes beendet sein muss (nur falls dieser überhaupt periodisch ist) bevor im aktuellen Syntheseabschnitt ein neuer Impuls kommen oder das Rauschsignal beginnen darf. Die Eingangsgrösse `di` gibt der Funktion an, um wieviele Abtastwerte der erste Impuls oder das Rauschen verzögert sein muss. Mit der Ausgangsgrösse `do` gibt die Funktion aus, wieviele Abtastwerte der letzten Periode nicht mehr im momentanen Abschnitt Platz gehabt haben und somit am Anfang des nächsten Abschnittes berücksichtigt werden müssen.
- Die Filterung des `ns` Abtastwerte langen Anregungssignals mit dem Synthesefilter (Matlab-Funktion `filter`) ergibt einen gleichlangen Abschnitt des rekonstruierten Sprachsignals.
Achtung: Der Zustandsvektor des Filters muss und darf nur am Anfang initialisiert (d.h. null gesetzt) werden. Dies hat somit im Hauptprogramm (also in `ueb6_1.m`) zu geschehen. Nachher muss der Funktion `filter` jeweils der alte Zustandsvektor wieder übergeben werden. Die Funktion `LPC_synthesis` muss deshalb die Zustandsvektoren `zi` und `zo` einfach weiterreichen.

Die Funktion `LPC_synthesis` können Sie (zusammen mit der Funktion `LPC_analysis`) wiederum mit einem Matlab-Programm testen, nämlich mit `ueb5_2.m`. Dieses LPC-Analyse-Synthese-Programm liest ein Sprachsignal ein, führt abschnittsweise die LPC-Analyse durch und stellt das Signal, die AKF-Maxima, die Stimmhaft/stimmlos-Entscheidungsschwelle (punktiert) und den Grundfrequenzverlauf graphisch dar. Anschliessend wird für jeden Satz von LPC-Parametern die LPC-Synthese-Funktion aufgerufen, also das Signal rekonstruiert, und in einem zweiten Fenster zusammen mit dem Originalsignal graphisch dargestellt. Sie können die beiden Signale auch anhören.

Verändern Sie nun die Stimmhaft/stimmlos-Entscheidungsschwelle (Programmparameter `vthr` in `ueb5_2.m`) so, dass das Signal am besten tönt. Tönt das Signal heiser bis kratzend, dann ist die Schwelle zu hoch, tönt es jedoch an den stimmlosen Stellen schnarrend, dann ist die Schwelle zu tief. Sie können auch den Einfluss anderer Programmparameter auf die Sprachqualität testen, beispielsweise den Einfluss der Prädiktorordnung oder der Länge des Analysefensters `WinSize`.

Verifizieren Sie auch, dass bei der Rekonstruktion des Signals die Dauer und die Grundfrequenz über die Programmparameter `DurModFac` und `FoModFac` voneinander unabhängig verändert werden können.