

# Formantanalyse mit Hilfe von Linearer Prädiktion

Robin Schmidt

## Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Das Konzept der linearen Prädiktion</b>	<b>2</b>
2.1	Optimalfilterung / Wiener-Filter . . . . .	2
2.2	Die Kostenfunktion . . . . .	3
2.3	Das Minimum der Kostenfunktion . . . . .	6
<b>3</b>	<b>Der Least Mean Square (LMS) Algorithmus</b>	<b>7</b>
3.1	Gradientenabstieg . . . . .	7
3.2	Der stochastische Gradient . . . . .	8
<b>4</b>	<b>Die Levinson/Durbin Rekursion</b>	<b>9</b>
4.1	Zeitreihenvorhersage . . . . .	9
4.2	Rekursive Bestimmung der Filterkoeffizienten . . . . .	10
<b>5</b>	<b>Anwendungen der Linearen Prädiktion</b>	<b>17</b>
5.1	Quelle-Filter Modell . . . . .	17
5.2	Inverse Filterung . . . . .	18
5.3	Spektralanalyse . . . . .	18
5.4	Resynthese und Vocoder-Sounds . . . . .	19
5.5	Formantverläufe und nichtstationäre Signale . . . . .	19
5.6	Sprachcodierung . . . . .	20
<b>6</b>	<b>Literatur</b>	<b>21</b>

Im diesem Text wird die folgende Notation benutzt:

### Variable Bedeutung

$A(z)$	Transferfunktion des Prediction-Error Filters
$a_k$	k-tes Element eines IIR-Filterkoeffizientenvektors
$b_k$	k-tes Element eines FIR-Filterkoeffizientenvektors
$d[n]$	gewünschtes Sample zum diskreten Zeitpunkt n ("desired")
$\hat{d}[n]$	Schätzwert des gewünschten Samples
$e[n]$	Fehlersample zum diskreten Zeitpunkt n
$E\{\dots\}$	Erwartungswert
$H(z)$	Transferfunktion des Synthesefilters im Signalmodell
$J(\mathbf{w})$	Fehler- bzw. Kostenfunktion (abhängig von Gewichtsvektor)
$\mathbf{p}$	Kreuzkorrelationsvektor
$\mathbf{R}$	Autokorrelationsmatrix
$w_k$	k-tes Element eines Gewichtsvektors
$\mathbf{w}$	Gewichtsvektor
$x[n]$	Eingangssample zum diskreten Zeitpunkt n
$\mathbf{x}[n]$	Eingangsvektor zum diskreten Zeitpunkt n
$\hat{x}[n]$	vorhergesagter Wert für das Sample $x[n]$
$y[n]$	Ausgangssignal eines Filters

Im allgemeinen werden skalare Größen durch Kleinbuchstaben dargestellt, Vektoren durch fettgedruckte Kleinbuchstaben und Matrizen durch fettgedruckte Großbuchstaben. Ein Hütchen über einer Variablen deutet an, dass es sich um einen Schätzwert handelt.

# 1 Einleitung

Bei der Analyse von Sprachsignalen spielen Formanten eine sehr wichtige Rolle. Formanten sind Frequenzbereiche, die im Spektrum besonders stark hervortreten. Die Lage dieser Frequenzbereiche ist abhängig von der Größe und geometrischen Form des Vokaltraktes, welches als Filter wirkt. Das Anregungssignal für dieses Filter wird entweder durch die glottalen Pulse (bei Vokalen) und/oder durch geräuschhafte Anteile aufgrund von turbulenten Luftströmungen (bei Konsonanten) erzeugt. Die Lagen der Formantfrequenzen sind jedoch unabhängig von der Grundfrequenz des glottalen Anregungssignals - sie werden nur durch das Vokaltraktfilter bestimmt. Es werden jedoch diejenigen, im Anregungssignal enthaltenen, Frequenzen verstärkt, die in die Formantbereiche hineinfallen. Da die Kenntnis der Formantfrequenzen in bestimmten Anwendungen (wie z.B. der Vokalerkennung) sehr nützlich ist, ist es oft sinnvoll, die Lagen der Formantfrequenzen aus einem Sprachsignal zu extrahieren. Das naheliegendste scheint bei einem solchen Problem (das ja im Spektralbereich angesiedelt ist) eine FFT-Analyse zu sein, bei der anschließend die Peaks im Spektrum gesucht werden. Dies ist aber in diesem Fall problematisch, da die meisten Peaks im FFT-Spektrum nicht die spektralen Eigenschaften unseres (Vokaltrakt)Filters widerspiegeln, sondern in erster Linie mit dem Anregungssignal zusammenhängen - Peaks treten nämlich bei allen ganzzahligen Vielfachen der Grundfrequenz auf. Man könnte jetzt lediglich anhand der Höhe eines Peaks entscheiden, ob dieses Peak in einen Formantbereich fällt oder nicht. Aber selbst dann weiß man immer noch nicht, wo der Formant jetzt tatsächlich liegt, da ja im allgemeinen nicht davon ausgegangen werden kann, dass die Formant-Mittenfrequenz ein ganzzahliges Vielfaches der Grundfrequenz des Anregungssignals ist. Daher wird bei der Formantanalyse oft ein anderer Ansatz verfolgt: mit Hilfe eines adaptiven Filters wird das Sprachsignal in eine Schätzung des Anregungssignals und des (Vokaltrakt)Filters zerlegt. Wenn wir jetzt den Frequenzgang unseres (Modell)Filters betrachten, dann entsprechen die spektralen Peaks tatsächlich den Formantfrequenzen. Bei der Modellierung wird davon ausgegangen, dass das Vokaltraktfilter im wesentlichen durch seine Polstellen bestimmt wird, die ja mit spektralen Peaks einhergehen - die eventuell vorhandenen Nullstellen werden bei der Modellierung nicht berücksichtigt. Wir werden sehen, dass die Modellierung eines solchen Allpol-Filters in engem Zusammenhang mit der linearen Prädiktion steht.

## 2 Das Konzept der linearen Prädiktion

### 2.1 Optimalfilterung / Wiener-Filter

Bei der Optimalfilterung geht es darum, aus einer gegebenen Eingangsfolge  $x[n]$  eine Schätzung/Vorhersage für das aktuelle Sample eines zweiten Signals  $d[n]$  abzugeben. Der Schätzwert für das vorherzusagende Sample wird mit  $\hat{d}$  bezeichnet ( $d$  steht für "desired", und der Hut steht dafür, dass es sich um einen Schätzwert handelt). Dieser Schätzwert wird als gewichtete Summe des aktuellen und vergangener Eingangssamples berechnet:

$$\hat{d}[n] = \sum_{k=0}^{K-1} w_k x[n-k] = \mathbf{w}^T \mathbf{x}[n] \quad (1)$$

wobei

$$\begin{aligned} \mathbf{w} &= (w_0, w_1, w_2, \dots, w_{K-1})^T \\ \mathbf{x}[n] &= (x[n], x[n-1], x[n-2], \dots, x[n-K+1])^T \end{aligned}$$

Dabei ist  $w_k$  ein Gewichtungsfaktor, mit dem das  $[n-k]$ -te Sample der Eingangsfolge in der Summation gewichtet wird. Wie in Gleichung 1 zu sehen ist, lässt sich diese gewichtete Summe auch als ein Skalarprodukt aus einem Eingangssignalvektor und einem Gewichtsvektor auffassen. Diese Vorschrift zur Vorhersage von  $d[n]$  kann durch einfache FIR-Filterung der Eingangsfolge  $x[n]$  implementiert werden. Die Struktur eines solchen FIR-Filters ist in Figure 1 abgebildet - das Ausgangssample  $y[n]$  des Filters entspricht demzufolge unserem Schätzwert  $\hat{d}[n]$ . Dieses Filter wird im Folgenden als Prädiktorfilter oder einfach Prädiktor bezeichnet

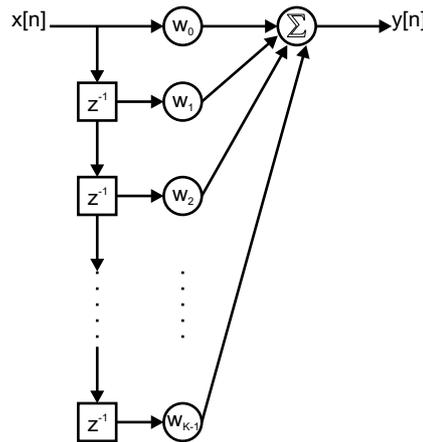


Figure 1: allgemeine Struktur eines FIR-Filters der Ordnung  $K - 1$

werden. Die Transferfunktion dieses Filters wird mit  $P(z)$  bezeichnet. Um eventuell auftretender Verwirrung vorzubeugen: Die Ordnung eines Filters ist definiert als die Anzahl der Unit-Delay Elemente, die in diesem Fall  $K - 1$  beträgt, die Dimensionalität des Gewichtsvektors  $\mathbf{w}$  beträgt jedoch  $K$ , da auch das unverzögerte Sample  $x[n]$  mit einem Faktor  $w_0$  gewichtet wird. Damit ein solche Vorhersage überhaupt möglich ist, muss das Signal  $d[n]$  natürlich in irgendeiner Weise mit dem Signal  $\mathbf{x}[n]$  zusammenhängen. Im allgemeinen wird davon ausgegangen, dass  $x[n]$  und  $d[n]$  zwei stationäre Zufallsfolgen (stationär bedeutet in diesem Zusammenhang, dass sich die statistischen Eigenschaften des Signals wie z.B. Mittelwert und Varianz zeitlich nicht ändern) mit einem Erwartungswert von Null sind, die miteinander korreliert sind - statistisch gesehen bedeutet dies, dass die Zufallsgröße  $d[n]$  sich aus den Zufallsgrößen  $x[n], x[n-1], \dots, x[n-K+1]$  (zumindest zum Teil) erklären lassen muss (wenn  $x[n]$  eine Zufallsgröße ist, dann sind natürlich auch  $x[n-1], \dots, x[n-K+1]$

Zufallsgrößen). Die Abhängigkeit des Signals  $d[n]$  von  $\mathbf{x}[n]$  kann also normalerweise nicht durch einen deterministischen Zusammenhang beschrieben werden sondern ist statistischer Natur.

## 2.2 Die Kostenfunktion

Ziel ist es nun, denjenigen Gewichtsvektor zu finden, mit dessen Hilfe man die Zufallsgröße  $d[n]$  optimal aus den Zufallsgrößen  $x[n], x[n-1], \dots, x[n-K+1]$  unter Verwendung von Gleichung 1 erklären kann (in der Statistik würde man von einem Problem der multiplen linearen Regression sprechen). Um zu definieren, was in diesem Fall optimal heißt, wird zunächst ein Fehlersignal  $e[n]$  eingeführt:

$$e[n] = d[n] - \hat{d}[n]$$

als Blockschaltbild stellt sich der Vorgang folgendermaßen dar:

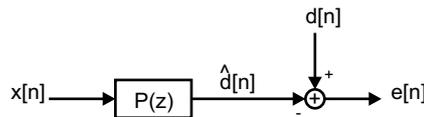


Figure 2: Prädiktor und Fehlersignal

Das Fehlersample zum diskreten Zeitpunkt  $n$  ist also einfach die Differenz zwischen dem wahren Sample  $d[n]$  und unserem Schätzwert  $\hat{d}[n]$ . Als nächstes wird eine Fehler- bzw. Kostenfunktion  $J(\mathbf{w})$  eingeführt:

$$J(\mathbf{w}) = E\{e^2[n]\} \quad (2)$$

wobei  $E$  für den statistischen Erwartungswert steht - schließlich ist  $e[n]$  auch wieder eine Zufallsfolge. Unsere Kostenfunktion stellt also die mittlere quadratische Abweichung des geschätzten Signals  $\hat{d}[n]$  vom wahren Signal  $d[n]$  dar. Man quadriert das Sample  $e[n]$  vor der Mittelwertbildung, damit sich positive und negative Abweichungen nicht gegenseitig aufheben können - man hätte auch den Betrag nehmen können, für die weitere mathematische Betrachtung ist der quadratische Mittelwert jedoch einfacher zu handhaben. Jetzt können wir ein mathematisches Kriterium für die Optimalität unseres Prädiktors formulieren: wir suchen denjenigen Gewichtsvektor  $\mathbf{w}$ , für den die Kostenfunktion  $J(\mathbf{w})$  minimal wird. Die notwendige Bedingung dafür lautet, dass alle  $K$  partiellen Ableitungen  $\partial J / \partial w_k$  gleichzeitig zu Null werden:

$$\frac{\partial J}{\partial w_k} = 0 \quad (k = 0, \dots, K-1)$$

bzw. der Gradient der Funktion  $J(\mathbf{w})$  muss zum Nullvektor werden:

$$\frac{\partial J}{\partial \mathbf{w}} = \mathbf{0}$$

Etwas ausführlicher geschrieben lautet die Kostenfunktion:

$$J(\mathbf{w}) = E\{(d[n] - \mathbf{w}^T \mathbf{x}[n])^2\} \quad (3)$$

dabei ist

$$\begin{aligned} (d[n] - \mathbf{w}^T \mathbf{x}[n])^2 &= (d[n])^2 - 2d[n]\mathbf{w}^T \mathbf{x}[n] + (\mathbf{w}^T \mathbf{x}[n])^2 \\ &= d^2[n] - 2d[n]\mathbf{w}^T \mathbf{x}[n] + \mathbf{w}^T \mathbf{x}[n]\mathbf{w}^T \mathbf{x}[n] \\ &= d^2[n] - 2\mathbf{w}^T d[n]\mathbf{x}[n] + \mathbf{w}^T \mathbf{x}[n]\mathbf{x}^T[n]\mathbf{w} \end{aligned}$$

Damit wird die Kostenfunktion zu:

$$\begin{aligned}
J(\mathbf{w}) &= E\{d^2[n] - 2\mathbf{w}^T d[n]\mathbf{x}[n] + \mathbf{w}^T \mathbf{x}[n]\mathbf{x}^T[n]\mathbf{w}\} \\
&= E\{d^2[n]\} - E\{2\mathbf{w}^T d[n]\mathbf{x}[n]\} + E\{\mathbf{w}^T \mathbf{x}[n]\mathbf{x}^T[n]\mathbf{w}\} \\
&= E\{d^2[n]\} - 2\mathbf{w}^T E\{d[n]\mathbf{x}[n]\} + \mathbf{w}^T E\{\mathbf{x}[n]\mathbf{x}^T[n]\}\mathbf{w}
\end{aligned}$$

wobei die Linearität des Erwartungswert-Operators  $E$  ausgenutzt wurde, um ihn in die Klammer hineinzuziehen. Als nächstes definieren wir für die drei Erwartungswerte in obiger Gleichung folgende Bezeichnungen: Die Varianz  $\sigma_d^2$  des vorherzusagenden Signals  $d[n]$  ist gegeben mit:

$$\sigma_d^2 = E\{d^2[n]\} \quad (4)$$

Der Kreuzkorrelationsvektor  $\mathbf{p}$  des Eingangssignals  $\mathbf{x}[n]$  mit dem vorherzusagenden Signal  $d[n]$  wird definiert als:

$$\mathbf{p} = E\{d[n]\mathbf{x}[n]\} \quad (5)$$

und als Autokorrelationsmatrix  $\mathbf{R}$  des Eingangssignals  $\mathbf{x}[n]$  bezeichnen wir:

$$\mathbf{R} = E\{\mathbf{x}[n]\mathbf{x}^T[n]\} \quad (6)$$

Der Kreuzkorrelationsvektor hat demzufolge folgende Form:

$$\mathbf{p} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{K-1} \end{pmatrix} = E \left\{ \begin{pmatrix} d[n]x[n] \\ d[n]x[n-1] \\ d[n]x[n-2] \\ \vdots \\ d[n]x[n-K+1] \end{pmatrix} \right\} = \begin{pmatrix} E\{d[n]x[n]\} \\ E\{d[n]x[n-1]\} \\ E\{d[n]x[n-2]\} \\ \vdots \\ E\{d[n]x[n-K+1]\} \end{pmatrix}$$

und die Autokorrelationsmatrix nimmt folgende Form an:

$$\begin{aligned}
\mathbf{R} &= E \left\{ \begin{pmatrix} x[n] \\ x[n-1] \\ x[n-2] \\ \vdots \\ x[n-K+1] \end{pmatrix} (x[n], x[n-1], x[n-2], \dots, x[n-K+1]) \right\} \\
&= E \left\{ \begin{pmatrix} x[n]x[n] & x[n]x[n-1] & \dots & x[n]x[n-K+1] \\ x[n-1]x[n] & x[n-1]x[n-1] & \dots & x[n-1]x[n-K+1] \\ \vdots & \vdots & \ddots & \vdots \\ x[n-K+1]x[n] & x[n-K+1]x[n-1] & \dots & x[n-K+1]x[n-K+1] \end{pmatrix} \right\} \\
&= E \left\{ \begin{pmatrix} x[n]x[n] & x[n]x[n-1] & \dots & x[n]x[n-K+1] \\ x[n]x[n-1] & x[n]x[n] & \dots & x[n-1]x[n-K+1] \\ \vdots & \vdots & \ddots & \vdots \\ x[n]x[n-K+1] & x[n-1]x[n-K+1] & \dots & x[n]x[n] \end{pmatrix} \right\} \\
&= \begin{pmatrix} r_0 & r_1 & r_2 & \dots & r_{K-1} \\ r_1 & r_0 & r_1 & \dots & r_{K-2} \\ r_2 & r_1 & r_0 & \dots & r_{K-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{K-1} & r_{K-2} & r_{K-3} & \dots & r_0 \end{pmatrix}
\end{aligned}$$

wobei in der vorletzten Zeile für die Vereinfachung der Diagonalelemente davon Gebrauch gemacht wurde, dass für stationäre Signale gilt :

$$E\{x[n-k]x[n-k]\} = E\{x[n]x[n]\} \quad \forall k$$

was nichts anderes bedeutet, als dass die Varianz des Signals  $x[n]$  sich nicht mit der Zeit ändert. Wie man aus obiger Gleichung schon erkennen kann, ist die Matrix  $\mathbf{R}$  aus den Autokorrelationskoeffizienten  $r_k$  des Eingangssignals  $x[n]$  zusammengesetzt, wobei für jedes  $r_k$  gilt:

$$r_k = E\{x[n]x[n-k]\}$$

Mit diesen Definitionen (Gleichungen 4, 5 und 6) kann man die Kostenfunktion  $J(\mathbf{w})$  als etwas übersichtlichere Matrix-Gleichung schreiben:

$$J(\mathbf{w}) = \sigma_d^2 - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w} \quad (7)$$

Dies ist also die Funktion, die es zu minimieren gilt. Wie schon gesagt, ist die notwendige Bedingung für ein Minimum, dass der Gradient der Funktion zum Nullvektor wird. Derjenige Filterkoeffizientenvektor  $\mathbf{w}$  der diese Bedingung erfüllt, ist der "optimale" Filterkoeffizientenvektor und wird im Folgenden mit  $\mathbf{w}_{opt}$  bezeichnet. Durch Ableitung ergibt sich also folgende Bedingung:

$$\frac{\partial J}{\partial \mathbf{w}} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w} = \mathbf{0} \quad (8)$$

was sich auch schreiben lässt als:

$$\mathbf{R}\mathbf{w} = \mathbf{p} \quad (9)$$

Dies ist die sogenannte Wiener-Hopf Gleichung. Auch wenn sie nicht sonderlich spektakulär aussieht, so ist sie doch aus theoretischer Sicht von fundamentaler Bedeutung für das Gebiet der adaptiven Filterung, da aus ihr eine explizite Berechnungsvorschrift für den optimalen Filterkoeffizientenvektor hervorgeht. Wenn nämlich davon ausgegangen wird, dass die Autokorrelationsmatrix  $\mathbf{R}$  invertierbar ist (was in der Regel der Fall ist), dann ergibt sich unser optimaler Filterkoeffizientenvektor als:

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{p} \quad (10)$$

Um zu zeigen, dass das so gefundene Extremum tatsächlich ein Minimum und kein Maximum ist, müsste man zeigen, dass die Hesse-Matrix (die Matrix der zweiten Ableitungen) der Funktion  $J(\mathbf{w})$  positiv definit ist. Dies wollen wir uns hier jedoch sparen, und stattdessen eine anschauliche Erklärung geben: Die Kostenfunktion  $J(\mathbf{w})$  ist eine quadratische Funktion des Gewichtsvektors  $\mathbf{w}$ . Für  $K = 1$  würde der aktuelle Wert für  $d[n]$  wird nur aus dem aktuellen Wert von  $x[n]$  vorhergesagt:  $\hat{d}[n] = w_0 x[n]$ .  $J(\mathbf{w})$  wäre dann einfach eine Parabel über der  $w_0$ -Achse. Für  $K = 2$  würde auch der vorhergehende Wert von  $x[n]$  einbezogen werden:  $\hat{d}[n] = w_0 x[n] + w_1 x[n-1]$ . In diesem Fall wäre  $J(\mathbf{w})$  ein Paraboloid über der  $(w_0, w_1)$ -Ebene. Für höherdimensionale Gewichtsvektoren würde sich  $J(\mathbf{w})$  als Hyperparaboloid darstellen. In jedem Falle aber - und das ist das Entscheidende - hätte  $J(\mathbf{w})$  genau eine Extremstelle. Diese kann aber deshalb kein Maximum sein, weil es für jeden Filterkoeffizientenvektor möglich sein wird, einen anderen zu finden, der die Kostenfunktion  $J(\mathbf{w})$  noch größer werden lässt (man braucht eigentlich nur einen Koeffizienten gegen unendlich gehen lassen). Gleichung 10 könnte also eigentlich schon die Lösung des Problems der linearen Prädiktion sein - jedoch sind in der Praxis noch ein paar mehr Stolpersteine aus dem Weg zu räumen: zum einen braucht man die Autokorrelationsmatrix  $\mathbf{R}$  des Eingangssignals  $x[n]$  und den Kreuzkorrelationsvektor von Eingangssignal und vorherzusagendem Signal  $d[n]$ , zum anderen ist Matrixinversion ein sehr rechenaufwändiger Prozess (um eine  $N \times N$  Matrix zu invertieren braucht man  $\mathcal{O}(N^3)$  Rechenoperationen). Da sowohl Autokorrelationsmatrix  $\mathbf{R}$  als auch Kreuzkorrelationsvektor  $\mathbf{p}$  als Erwartungswerte definiert sind, können sie aus den vorliegenden Daten nur näherungsweise bestimmt werden - indem nämlich Erwartungswerte durch Kurzzeitmittelwerte ersetzt werden. Bleibt trotzdem noch das Problem der Matrixinversion. In den nächsten Kapiteln werden zwei alternative Wege aufgezeigt werden um die Lösung der Wiener-Hopf Gleichung zu finden und damit den optimalen Filterkoeffizientenvektor zu bestimmen.

### 2.3 Das Minimum der Kostenfunktion

Jetzt kennen wir also denjenigen Gewichtsvektor  $\mathbf{w}_{opt}$ , der die Kostenfunktion minimiert. Was uns als nächstes interessieren könnte, ist der Wert, den die Kostenfunktion an dieser Stelle  $\mathbf{w}_{opt}$  annimmt. Dazu setzen wir einfach den optimalen Gewichtsvektor aus Gleichung 10 in die Kostenfunktion (in der Form von Gleichung 7) ein:

$$\begin{aligned} J_{min} &= J(\mathbf{w}_{opt}) \\ &= \sigma_d^2 - 2\mathbf{w}_{opt}^T \mathbf{p} + \mathbf{w}_{opt}^T \mathbf{R} \mathbf{w}_{opt} \\ &= \sigma_d^2 - 2(\mathbf{R}^{-1} \mathbf{p})^T \mathbf{p} + (\mathbf{R}^{-1} \mathbf{p})^T \mathbf{R} (\mathbf{R}^{-1} \mathbf{p}) \end{aligned}$$

Um diese Gleichung zu vereinfachen machen wir zunächst Gebrauch von der Rechenregel:  $\mathbf{A} \mathbf{B}^T = \mathbf{B}^T \mathbf{A}^T$ , wobei  $\mathbf{A} \hat{=} \mathbf{R}^{-1}$  und  $\mathbf{B} \hat{=} \mathbf{p}$ :

$$J_{min} = \sigma_d^2 - 2\mathbf{p}^T (\mathbf{R}^{-1})^T \mathbf{p} + \mathbf{p}^T (\mathbf{R}^{-1})^T \mathbf{R} \mathbf{R}^{-1} \mathbf{p}$$

Als nächstes stellen wir fest, dass  $\mathbf{R} \mathbf{R}^{-1} = \mathbf{E}$  wobei  $\mathbf{E}$  die Einheitsmatrix ist. Damit fällt der Term  $\mathbf{R} \mathbf{R}^{-1}$  einfach heraus:

$$\begin{aligned} J_{min} &= \sigma_d^2 - 2\mathbf{p}^T (\mathbf{R}^{-1})^T \mathbf{p} + \mathbf{p}^T (\mathbf{R}^{-1})^T \mathbf{p} \\ &= \sigma_d^2 - \mathbf{p}^T (\mathbf{R}^{-1})^T \mathbf{p} \end{aligned}$$

Da  $\mathbf{R}$  eine symmetrische Matrix ist, ist auch  $\mathbf{R}^{-1}$  eine symmetrische Matrix. Für eine symmetrische Matrix  $\mathbf{A}$  gilt:  $\mathbf{A}^T = \mathbf{A}$ . Somit gilt also:  $(\mathbf{R}^{-1})^T = \mathbf{R}^{-1}$ . Damit erhalten wir als Endergebnis für das Minimum der Kostenfunktion:

$$J_{min} = \sigma_d^2 - \mathbf{p}^T \mathbf{R}^{-1} \mathbf{p} = \sigma_d^2 - \mathbf{p}^T \mathbf{w}_{opt} \quad (11)$$

### 3 Der Least Mean Square (LMS) Algorithmus

#### 3.1 Gradientenabstieg

Da die direkte Lösung der Wiener-Hopf Gleichung in Praxis oft nicht möglich ist, soll hier ein iteratives Verfahren zur Bestimmung der optimalen Filterkoeffizientenvektors vorgestellt werden. Iterativ bedeutet in diesem Fall, dass man mit irgendeinem Filterkoeffizientenvektor  $\mathbf{w}[0]$  beginnt und diesen dann Schritt für Schritt verbessert. Die Null in den eckigen Klammern steht auch hier für die diskrete Zeit - da wir den Filterkoeffizientenvektor mit jedem Zeitschritt verbessern, wird dieser selber zu einer zeitabhängigen Größe, wobei  $\mathbf{w}[0]$  die Initialisierung darstellt. In der Regel wird man für  $\mathbf{w}[0]$  den Nullvektor wählen. Wie gesagt entspricht unser optimaler Filterkoeffizientenvektor  $\mathbf{w}_{opt}$  dem Minimum eines (nach oben offenen) Hyperparaboloiden - anschaulicherweise sollte man sich einen normalen Paraboloiden oder auch nur eine Parabel vorstellen. Der Gradient der Kostenfunktion  $\partial J / \partial \mathbf{w}$  gibt diejenige Richtung im  $\mathbf{w}$ -Raum an, in der die Kostenfunktion am steilsten ansteigt. In der Richtung des negativen Gradienten hingegen fällt die Kostenfunktion am schnellsten ab. Wenn wir also den Filterkoeffizientenvektor ein kleines Stückchen in Richtung des negativen Gradienten bewegen, dann können wir erwarten, dass für unseren neuen Filterkoeffizientenvektor der Wert der Kostenfunktion geringer sein wird. Daraus können wir eine Adaptionvorschrift gewinnen:

$$\mathbf{w}_{neu} = \mathbf{w}_{alt} - \eta \cdot \frac{\partial J}{\partial \mathbf{w}_{alt}}$$

bzw.:

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \eta \cdot \frac{\partial J}{\partial \mathbf{w}[n]} \quad (12)$$

wobei  $\eta$  eine kleine Konstante ist, die sinnvoll gewählt werden muss - die sogenannte Adaption- bzw. Lernrate. Offensichtlich benötigen wir zur Umsetzung dieser Vorschrift einen Ausdruck für  $\partial J / \partial \mathbf{w}[n]$  - also für den Gradienten der Kostenfunktion  $J$  an der Stelle  $\mathbf{w}[n]$ . Aus Gleichung 8 wissen wir:

$$\frac{\partial J}{\partial \mathbf{w}} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}$$

...das bringt uns aber hier nicht viel weiter. Stattdessen wollen wir einen alternativen, mathematisch äquivalenten Ausdruck für den Gradienten entwickeln. Dazu rufen wir uns nochmal die Definition der Kostenfunktion  $J(\mathbf{w})$  in Erinnerung:

$$J(\mathbf{w}) = E\{e^2[n]\}$$

Also:

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} E\{e^2[n]\} = E\left\{\frac{\partial}{\partial \mathbf{w}} e^2[n]\right\}$$

wobei die Linearität von Ableitung und Erwartungswert ausgenutzt wurde, um die Ableitung in den Erwartungswert hineinzuziehen. Da  $e[n]$  natürlich auch wieder eine Funktion von  $\mathbf{w}$  ist ergibt sich nach der Kettenregel:

$$\frac{\partial J}{\partial \mathbf{w}} = E\left\{\frac{\partial}{\partial \mathbf{w}} e^2[n]\right\} = E\{2e[n] \frac{\partial}{\partial \mathbf{w}} e[n]\}$$

Schließlich erinnern wir uns, dass:

$$e[n] = d[n] - \hat{d}[n] = d[n] - \mathbf{w}^T \mathbf{x}[n]$$

Somit ist die innere Ableitung gegeben mit:

$$\frac{\partial}{\partial \mathbf{w}} e[n] = \frac{\partial}{\partial \mathbf{w}} (d[n] - \mathbf{w}^T \mathbf{x}[n]) = -\mathbf{x}[n]$$

Damit ergibt sich als Gesamtergebnis:

$$\frac{\partial J}{\partial \mathbf{w}} = E\{-2e[n]\mathbf{x}[n]\} = -2E\{e[n]\mathbf{x}[n]\} \quad (13)$$

Der Gradient ist also einfach proportional zu dem negativen Erwartungswert des Produkts aus Fehlersample  $e[n]$  und Eingabevektor  $\mathbf{x}[n]$ .

### 3.2 Der stochastische Gradient

Der auf diese Weise hergeleitete Gradient hat immer noch einen Haken: er enthält einen Erwartungswert. Wir könnten jetzt wieder versuchen diesen Erwartungswert zu approximieren, indem wir die Größe  $e[n]\mathbf{x}[n]$  eine Zeit lang beobachten und dann einen Kurzzeitmittelwert bilden. Beim sogenannten LMS-Algorithmus wird stattdessen einfach der Erwartungswert durch den aktuellen Wert ersetzt (man könnte dies natürlich auch als Kurzzeitmittelwert über ein Element ansehen). Es wird also folgende Approximation benutzt:

$$E\{e[n]\mathbf{x}[n]\} \approx e[n]\mathbf{x}[n]$$

Dies ist endlich mal eine Größe, die wir aus unseren vorliegenden Signalen direkt berechnen können. Damit können wir den Gradienten aus Gleichung 13 folgendermaßen approximieren:

$$\frac{\partial J}{\partial \mathbf{w}} = -2E\{e[n]\mathbf{x}[n]\} \approx -2e[n]\mathbf{x}[n]$$

Dies wird als stochastischer Gradient bezeichnet - er entspricht einer Größe, die im zeitlichen Mittel in Richtung des tatsächlichen Gradienten der Kostenfunktion zeigt, jedoch von Zeitschritt zu Zeitschritt mehr oder weniger starke zufällige Abweichungen von der Richtung des tatsächlichen Gradienten aufweist. Wenn wir diesen Ausdruck in Gleichung 12 an die Stelle des tatsächlichen Gradienten setzen ergibt sich als Adaptionvorschrift:

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \eta \cdot -2e[n]\mathbf{x}[n] = \mathbf{w}[n] + 2\eta e[n]\mathbf{x}[n]$$

da  $\eta$  eine Konstante ist, die vom Benutzer gewählt werden muss, können wir den Faktor 2 auch in der Adaptionrate  $\eta$  absorbieren, indem wir  $\eta$  einfach verdoppeln. Daraus ergibt sich die Adaptionvorschrift des LMS-Algorithmus:

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \eta e[n]\mathbf{x}[n] \quad (14)$$

Da alle drei Größen  $\mathbf{w}[n]$ ,  $\mathbf{x}[n]$  und  $e[n]$  auf der rechten Seite von Gleichung 14 zum Zeitpunkt  $n$  bekannt sind, können wir mit dieser Vorschrift tatsächlich für den nächsten Zeitschritt  $\mathbf{w}[n+1]$  einen verbesserten Filterkoeffizientenvektor berechnen. Der LMS-Algorithmus arbeitet also samplebasiert - soll heißen, dass er für jeden Zeitschritt einen neuen Filterkoeffizientenvektor berechnet. Die im nächsten Kapitel vorgestellte Levinson-Durbin Rekursion dagegen arbeitet blockbasiert, d.h. dass aus dem Signal zeitliche Blöcke herausgeschnitten werden und für jeden einzelnen Block der optimale Filterkoeffizientenvektor berechnet wird. Es folgt eine Zusammenfassung des LMS-Algorithmus:

1. initialisiere Zeitindex mit Null:

$$n = 0$$

initialisiere Filterkoeffizientenvektor mit dem Nullvektor:

$$\mathbf{w}[n] = \mathbf{w}[0] = \mathbf{0}$$

2. berechne Schätzwert  $\hat{d}[n]$  für das gewünschte Signal  $d[n]$ :

$$\hat{d}[n] = \mathbf{w}^T[n]\mathbf{x}[n]$$

mit

$$\mathbf{x}[n] = (x[n], x[n-1], x[n-2], \dots, x[n-K+1])^T$$

3. berechne Fehlersignal  $e[n]$ :

$$e[n] = d[n] - \hat{d}[n]$$

4. verbessere Filterkoeffizientenvektor  $\mathbf{w}$  gemäß LMS-Regel:

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \eta e[n]\mathbf{x}[n]$$

5. inkrementiere Zeitindex  $n$ :

$$n = n + 1$$

und gehe zurück zu 2

## 4 Die Levinson/Durbin Rekursion

### 4.1 Zeitreihenvorhersage

Ein wichtiger Spezialfall für die Anwendung von Wiener-Filtern ist die Zeitreihenvorhersage. Dabei soll ein zukünftiges Sample  $x[n+1]$  einer Zeitreihe (also eines Signals) aus dem aktuellen und vergangenen Samples derselben Zeitreihe vorhergesagt werden. Dies bedeutet, dass unser gewünschtes Signal  $d[n]$  dem Signal  $x[n+1]$  entspricht. Um die Filterkoeffizienten zu bestimmen (z.B. mit LMS), benötigt man natürlich zum Zeitpunkt  $n$  das tatsächliche Sample  $x[n+1]$  damit man das Fehlersignal  $e[n]$  berechnen kann. Dies steht aber zum Zeitpunkt  $n$  noch nicht zur Verfügung. Daher verzögert man in der Praxis alle Signale um ein Sample. Dies bedeutet, dass  $d[n]$  jetzt zu  $x[n]$  wird, und der Eingangssignalvektor  $\mathbf{x}[n]$  aus dem ein Schätzwert für  $x[n]$  vorhergesagt werden soll ergibt sich nur aus vergangenen Samples des Signals  $x[n]$ . Die Berechnungsvorschrift für den Schätzwert für  $x[n]$  ergibt sich also als:

$$\hat{x}[n] = \sum_{k=1}^K w_k x[n-k] = \mathbf{w}^T \mathbf{x}[n] \quad (15)$$

wobei

$$\begin{aligned} \mathbf{w} &= (w_1, w_2, \dots, w_K)^T \\ \mathbf{x}[n] &= (x[n-1], x[n-2], \dots, x[n-K])^T \end{aligned}$$

Man beachte, dass sich die Notation hier etwas geändert hat: der Index  $k$  läuft jetzt von 1 bis  $K$  statt von 0 bis  $K-1$  um der Tatsache Rechnung zu tragen, dass wir es jetzt nur vergangene Samples zur Vorhersage benutzen - vorher hatten wir das aktuelle Sample  $x[n]$  mit in die Vorhersagevorschrift einbezogen, da wir ja ein zweites (korreliertes) Signal  $d[n]$  (was uns zusätzlich zur Verfügung stand) vorhersagen wollten. Jetzt allerdings wollen wir ja das Sample  $x[n]$  zum Zeitpunkt  $n$  aus den vergangenen Werten desselben Signals vorhersagen. Als Blockschaltbild ergibt sich:

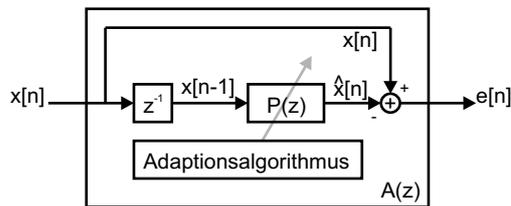


Figure 3: Prädiktorfilter  $P(z)$  und FPE-Filter  $A(z)$

Das Differenzsignal  $e[n]$  aus tatsächlichem Sample  $x[n]$  und vorhergesagtem Sample  $\hat{x}[n]$  kann auch wieder als gefilterte Version des Signal  $x[n]$  angesehen werden. Daher wird der gesamte Kasten in der Literatur auch oft als "Forward Prediction Error Filter" (FPE-Filter) bezeichnet. Diese Bezeichnung scheint etwas redundant zu sein, da ja Prediction (Vorhersage) schon das Forward (Vorwärts) impliziert. Da aber in der Literatur auch gelegentlich die "Rückwärtsvorhersage" auftaucht (die Schätzung eines Samples, das weiter hinten liegt als die zur Verfügung stehenden), wird das Forward meist mitgeschrieben. Die Transferfunktion dieses Filters wird meist mit  $A(z)$  bezeichnet. Im Zeitbereich ergibt sich als Ausgangssignal des FPE-Filters folgendes Signal:

$$\begin{aligned} e[n] &= x[n] - \hat{x}[n] \\ &= x[n] - \sum_{k=1}^K w_k x[n-k] \end{aligned} \quad (16)$$

Dies ist auch wieder exakt die Form eines FIR-Filters. Zur Verdeutlichung sei hier nochmal allgemein das Ausgangssignal eines FIR Filters angegeben:

$$y[n] = \sum_{k=0}^K b_k x[n-k]$$

Wenn wir  $b_0 = 1$  und  $b_k = -w_k \quad k = 1, \dots, K$  setzen, ist das Signal  $e[n]$  gleich dem Signal  $y[n]$ .

## 4.2 Rekursive Bestimmung der Filterkoeffizienten

Die Levinson-Durbin Rekursion macht sich eine spezielle Eigenschaft der Autokorrelationsmatrix  $\mathbf{R}$  zunutze um die Wiener-Hopf Gleichung effizient zu lösen. Dabei wird die rechenaufwändige Matrixinversion umgangen. Stattdessen wird wie folgt vorgegangen: Zuerst wird der optimale Gewichtsvektor erster Ordnung berechnet, welcher nur aus einem Element besteht (der Vektor reduziert sich also zu einem Skalar). Damit reduziert sich auch die entsprechende Autokorrelationsmatrix auf eine  $1 \times 1$  Matrix - also einen Skalar. Die Lösung Wiener-Hopf Gleichung für den optimalen ein-elementigen Gewichtsvektor enthält somit statt einer inversen Matrix eine einfache Division. Danach wird ein optimaler Filterkoeffizientenvektor der Länge  $K + 1$  aus einem als bekannt angenommenen Filterkoeffizientenvektor der Länge  $K$  berechnet - und zwar solange bis die endgültige gewünschte Ordnung der Vorhersage erreicht ist. Die Berechnungsvorschrift ist also rekursiv in dem Sinne, dass sich Filterkoeffizientenvektoren höherer Ordnung immer aus dem jeweiligen Vektor niedrigerer Ordnung berechnen lassen. Die Eigenschaft der Autokorrelationsmatrix  $\mathbf{R}$ , die ein solches rekursives Vorgehen ermöglicht, ist die, dass alle Elemente entlang einer beliebigen Diagonalen (von links oben nach rechts unten) gleich sind. Eine Matrix mit dieser Eigenschaft wird auch als Töplitz-Matrix bezeichnet. Die Autokorrelationsmatrix der Größe  $K \times K$  ist gegeben mit:

$$\mathbf{R}_K = \begin{pmatrix} r_0 & r_1 & r_2 & r_3 & \dots & r_{K-1} \\ r_1 & r_0 & r_1 & r_2 & \dots & r_{K-2} \\ r_2 & r_1 & r_0 & r_1 & \dots & r_{K-3} \\ r_3 & r_2 & r_1 & r_0 & \dots & r_{K-4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{K-1} & r_{K-2} & r_{K-3} & r_{K-4} & \dots & r_0 \end{pmatrix}$$

wobei ein neuer Index eingeführt wurde, um die Größe der Matrix zu charakterisieren. Die einzelnen Autokorrelationskoeffizienten  $r_k$  sind definiert als:

$$r_k = E\{x[n]x[n-k]\}$$

Da diese in der Regel unbekannt sind, müssen sie aus den Daten geschätzt werden. Dabei wird der Erwartungswert  $E$  durch den Mittelwert ersetzt. Die Berechnungsvorschrift lautet:

$$r_k = \frac{1}{N} \sum_{n=0}^{N-k-1} x[n]x[n-k] \quad (17)$$

wobei  $N$  die Anzahl der für die Schätzung benutzten Samples (also die Länge des Zeitfensters) ist. Man beachte, dass diese Formel nur für  $k = 0$  einen echten Kurzzeitmittelwert berechnet, da nur in diesem Fall die Anzahl der summierten Elemente gleich  $N$  ist. Für alle anderen  $k$  ist die Anzahl der summierten Elemente kleiner als  $N$ , nämlich gleich  $N - k$ . Das bedeutet, dass die höheren Autokorrelationskoeffizienten systematisch zu klein (vom Betrag her) geschätzt werden, da ja in jedem Fall durch  $N$  dividiert wird. Aus Gründen, auf die hier nicht näher eingegangen werden soll, benutzt man diese Formel jedoch oft trotzdem. Wenn aber davon ausgegangen wird, dass das größte  $k$  immer noch sehr viel kleiner als  $N$  ist, fällt dieser Unterschied sowieso nicht weiter ins Gewicht.

Aufgrund der Tatsache, dass jetzt das Prädiktor-Eingangssignal und das gewünschte Signal ein und dasselbe Signal sind (nur dass das Prädiktor-Eingangssignal um ein Sample verzögert ist) stehen im Kreuzkorrelationsvektor  $\mathbf{p}$  jetzt auch die Autokorrelationskoeffizienten des Signals  $x[n]$ . Der Kreuzkorrelationsvektor wird also zu einer Art Autokorrelationsvektor und ist gegeben mit:

$$\mathbf{p}_K = E\{x[n]\mathbf{x}[n]\} = E \left\{ \begin{pmatrix} x[n]x[n-1] \\ x[n]x[n-2] \\ x[n]x[n-3] \\ \vdots \\ x[n]x[n-K] \end{pmatrix} \right\} = \begin{pmatrix} E\{x[n]x[n-1]\} \\ E\{x[n]x[n-2]\} \\ E\{x[n]x[n-3]\} \\ \vdots \\ E\{x[n]x[n-K]\} \end{pmatrix}$$

oder, anders geschrieben:

$$\mathbf{p}_K = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_K \end{pmatrix} \quad (18)$$

wobei auch hier der Index für die Länge des Vektors steht (unter Länge soll hier die Anzahl der Elemente verstanden werden). Somit können wir bereits eine Formel für den optimalen einelementigen Gewichtsvektor angeben: die Lösung der Wiener-Hopf Gleichung  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$  reduziert sich zu:

$$\mathbf{w}_K = r_1/r_0 \quad K = 1 \quad (19)$$

wobei K die Ordnung des Prädiktors angibt. Wie bereits erwähnt, stehen  $\mathbf{R}$  und  $\mathbf{p}$  in einem engen Zusammenhang. Wenn wir nämlich mit  $\tilde{\mathbf{p}}_K$  einen Vektor definieren, der die Elemente von  $\mathbf{p}$  in umgekehrter Reihenfolge enthält, also:

$$\tilde{\mathbf{p}}_K = \begin{pmatrix} r_K \\ r_{K-1} \\ r_{K-2} \\ \vdots \\ r_1 \end{pmatrix}$$

läßt sich die nächstgrößere Autokorrelationsmatrix  $\mathbf{R}_{K+1}$  in Blockmatrizenform folgendermaßen darstellen:

$$\begin{aligned} \mathbf{R}_{K+1} &= \begin{pmatrix} r_0 & r_1 & r_2 & r_3 & \dots & r_K \\ r_1 & r_0 & r_1 & r_2 & \dots & r_{K-1} \\ r_2 & r_1 & r_0 & r_1 & \dots & r_{K-2} \\ r_3 & r_2 & r_1 & r_0 & \dots & r_{K-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r_K & r_{K-1} & r_{K-2} & r_{K-3} & \dots & r_0 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{R}_K & \tilde{\mathbf{p}}_K \\ \tilde{\mathbf{p}}_K^T & r_0 \end{pmatrix} \end{aligned}$$

Analog dazu definieren wir einen Vektor  $\tilde{\mathbf{w}}_K$ , der die Elemente des optimalen Gewichtsvektors  $\mathbf{w}_K$  in umgekehrter Reihenfolge enthält:

$$\tilde{\mathbf{w}}_K = \begin{pmatrix} w_K \\ w_{K-1} \\ w_{K-2} \\ \vdots \\ w_1 \end{pmatrix}$$

Für den optimalen Gewichtsvektor  $\mathbf{w}_K$  der Länge  $K$  muss die Wiener-Hopf Gleichung gelten (diesmal mit Indizes):

$$\mathbf{R}_K \mathbf{w}_K = \mathbf{p}_K$$

Aufgrund der Töplitz-Struktur der Matrix  $\mathbf{R}_K$  ist diese Aussage äquivalent zu:

$$\mathbf{R}_K \tilde{\mathbf{w}}_K = \tilde{\mathbf{p}}_K \quad (20)$$

Nach wie vor ist es unser Ziel, eine Vorschrift herzuleiten, mit der wir aus dem als bekannt vorausgesetzten optimalen Gewichtsvektor der Länge  $K$  einen optimalen Gewichtsvektor der Länge  $K + 1$  berechnen können. Natürlich muss auch für den Gewichtsvektor der Ordnung  $K + 1$  die Wiener-Hopf Gleichung gelten. Wir verlangen also:

$$\mathbf{R}_{K+1} \mathbf{w}_{K+1} = \mathbf{p}_{K+1} \quad (21)$$

Für die weitere Betrachtung ist es nötig, für den optimalen Gewichtsvektor neben dem Längenindex einen zweiten Index für die Ordnung der Lösung einzuführen. Dieser zweite Index wird hochgestellt notiert. Der optimale Gewichtsvektor der Ordnung  $K$  mit der Länge  $K$  ist also gegeben mit:  $\mathbf{w}_K^K$ . Die ersten  $K$  Elemente des optimalen Gewichtsvektors der Ordnung  $K + 1$  werden notiert als:  $\mathbf{w}_K^{K+1}$ . Diese Notation ist möglicherweise am Anfang etwas verwirrend, wir werden aber (hoffentlich) gleich sehen, dass sie nützlich ist. Mit dieser Definition können wir Gleichung 21 in Blockmatrizenform schreiben:

$$\begin{pmatrix} \mathbf{R}_K & \tilde{\mathbf{p}}_K \\ \tilde{\mathbf{p}}_K^T & r_0 \end{pmatrix} \begin{pmatrix} \mathbf{w}_K^{K+1} \\ \kappa_{K+1} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_K \\ r_{K+1} \end{pmatrix} = \mathbf{p}_{K+1} \quad (22)$$

wobei  $\kappa_{K+1}$  ein noch zu bestimmender Koeffizient ist. Nochmal zu Verdeutlichung:  $\mathbf{w}_K^{K+1}$  sind die ersten  $K$  Elemente des Lösungsvektors der Ordnung  $K + 1$  und  $\kappa_{K+1}$  ist das  $(K + 1)$ te Element dieses Vektors. Also:

$$\mathbf{w}_{K+1}^{K+1} = \begin{pmatrix} \mathbf{w}_K^{K+1} \\ \kappa_{K+1} \end{pmatrix} \quad (23)$$

Damit haben wir  $K + 1$  Gleichungen für die  $K + 1$  Unbekannten (nämlich die  $K$  Elemente von  $\mathbf{w}_K^{K+1}$  und  $\kappa_{K+1}$ ). Als nächstes ziehen wir aus Gleichung 22 eine Vektorgleichung und eine Skalargleichung heraus indem wir einfach nach den Gesetzen der Matrixmultiplikation ("Zeile mal Spalte") vorgehen. Die Vektorgleichung lautet:

$$\mathbf{R}_K \mathbf{w}_K^{K+1} + \tilde{\mathbf{p}}_K \cdot \kappa_{K+1} = \mathbf{p}_K \quad (24)$$

und die Skalargleichung lautet:

$$\tilde{\mathbf{p}}_K^T \mathbf{w}_K^{K+1} + r_0 \kappa_{K+1} = r_{K+1} \quad (25)$$

Als nächsten Schritt multiplizieren wir beide Seiten von 24 mit  $\mathbf{R}_K^{-1}$  und erhalten:

$$\mathbf{R}_K^{-1} \mathbf{R}_K \mathbf{w}_K^{K+1} + \mathbf{R}_K^{-1} \tilde{\mathbf{p}}_K \cdot \kappa_{K+1} = \mathbf{R}_K^{-1} \mathbf{p}_K$$

wobei der Term  $\mathbf{R}_K^{-1} \mathbf{R}_K$  wieder die Einheitsmatrix ergibt und somit rausfällt:

$$\mathbf{w}_K^{K+1} + \mathbf{R}_K^{-1} \tilde{\mathbf{p}}_K \cdot \kappa_{K+1} = \mathbf{R}_K^{-1} \mathbf{p}_K$$

Jetzt stellen wir fest, dass der Term  $\mathbf{R}_K^{-1} \mathbf{p}_K$  nach der Wiener-Hopf Gleichung genau dem optimalen Gewichtsvektor  $\mathbf{w}_K^K$  der Ordnung  $K$  entspricht. Analog dazu gilt:  $\mathbf{R}_K^{-1} \tilde{\mathbf{p}}_K = \tilde{\mathbf{w}}_K^K$  (siehe Gleichung 20). Damit vereinfacht sich die Gleichung weiter:

$$\mathbf{w}_K^{K+1} + \tilde{\mathbf{w}}_K^K \cdot \kappa_{K+1} = \mathbf{w}_K^K$$

bzw.:

$$\mathbf{w}_K^{K+1} = \mathbf{w}_K^K - \kappa_{K+1} \cdot \tilde{\mathbf{w}}_K^K \quad (26)$$

Jetzt ist etwas entscheidendes passiert: Wir sehen, dass die ersten  $K$  Elemente des optimalen Gewichtsvektors der Ordnung  $K + 1$  (notiert als  $\mathbf{w}_K^{K+1}$ ) sich als Linearkombination aus  $\mathbf{w}_K^K$  und  $\tilde{\mathbf{w}}_K^K$  berechnen lassen. Die

einzigste Größe die es jetzt noch zu bestimmen gilt ist der Koeffizient  $\kappa_{K+1}$  welcher auch als Reflektionsfaktor bezeichnet wird. Dazu setzen wir das Ergebnis für  $\mathbf{w}_K^{K+1}$  aus Gleichung 26 in Gleichung 25 ein und erhalten:

$$\tilde{\mathbf{p}}_K^T (\mathbf{w}_K^K - \kappa_{K+1} \cdot \tilde{\mathbf{w}}_K^K) + r_0 \kappa_{K+1} = r_{K+1}$$

Ausmultiplizieren ergibt:

$$\tilde{\mathbf{p}}_K^T \mathbf{w}_K^K - \kappa_{K+1} \cdot \tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K + r_0 \kappa_{K+1} = r_{K+1}$$

Als nächstes wir der erste Term nach rechts gebracht:

$$-\kappa_{K+1} \cdot \tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K + r_0 \kappa_{K+1} = r_{K+1} - \tilde{\mathbf{p}}_K^T \mathbf{w}_K^K$$

bzw.:

$$r_0 \kappa_{K+1} - \kappa_{K+1} \cdot \tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K = r_{K+1} - \tilde{\mathbf{p}}_K^T \mathbf{w}_K^K$$

Jetzt kann  $\kappa_{K+1}$  ausgeklammert werden:

$$\kappa_{K+1} (r_0 - \tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K) = r_{K+1} - \tilde{\mathbf{p}}_K^T \mathbf{w}_K^K$$

und wir erhalten:

$$\kappa_{K+1} = \frac{r_{K+1} - \tilde{\mathbf{p}}_K^T \mathbf{w}_K^K}{r_0 - \tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K}$$

Da es beim Skalarprodukt egal ist, welcher von beiden Vektoren in umgekehrter Reihenfolge steht gilt:  $\tilde{\mathbf{p}}_K^T \mathbf{w}_K^K = \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K$ . Für den Fall, dass beide Vektoren in umgekehrter Reihenfolge stehen, kann man sich beide Umkehrungen auch sparen denn:  $\tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K = \mathbf{p}_K^T \mathbf{w}_K^K$ . Also können wir auch schreiben:

$$\kappa_{K+1} = \frac{r_{K+1} - \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K}{r_0 - \mathbf{p}_K^T \mathbf{w}_K^K} \quad (27)$$

Mit den Gleichungen 27, 26 und 23 haben wir also alles was wir brauchen, um aus dem optimalen Gewichtsvektor der Ordnung  $K$  den optimalen Gewichtsvektor der nächsthöheren Ordnung  $K + 1$  zu berechnen. Zuerst berechnet man  $\kappa_{K+1}$  mit 27 und dann  $\mathbf{w}_K^{K+1}$  mit 26 und erhält dann nach 23 den Gewichtsvektor  $\mathbf{w}_{K+1}^{K+1}$  nächsthöherer Ordnung. Man beachte, dass der Reflektionskoeffizient  $\kappa_{K+1}$  eine Doppelfunktion hat: erstens skaliert er  $\tilde{\mathbf{w}}_K^K$  in Gleichung 26, zweitens ist er das letzte Element in unserem optimalen Gewichtsvektor der Ordnung  $K + 1$ . Wir können jedoch die Rekursion mit Hilfe von Gleichung 11 noch weiter vereinfachen. Der Übersichtlichkeit halber sei sie hier nochmal angegeben:

$$J_{min} = \sigma_d^2 - \mathbf{p}^T \mathbf{w}_{opt}$$

Wir stellen zunächst fest, dass die Varianz des vorherzusagenden Signals  $\sigma_d^2 = E\{d^2[n]\}$  in unserem speziellen Fall der Varianz des Eingangssignals  $E\{x^2[n]\}$  entspricht - und diese wiederum ist ja genau der nullte Autokorrelationskoeffizient  $r_0$ . Der Term  $\mathbf{p}^T \mathbf{w}_{opt}$  hatte zwar im Kapitel 2 noch keine Indizes für die Ordnung der Prädiktion, von der Bedeutung her ist er jedoch identisch mit  $\mathbf{p}_K^T \mathbf{w}_K^K$ . Für das Minimum der Kostenfunktion bei gegebener Filterordnung  $K$  gilt also:

$$J_{min}^K = r_0 - \mathbf{p}_K^T \mathbf{w}_K^K \quad (28)$$

was wir als den Nenner des Bruchs in Gleichung 27 identifizieren. Also können wir Gleichung 27 auch schreiben als:

$$\kappa_{K+1} = \frac{r_{K+1} - \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K}{J_{min}^K} \quad (29)$$

Als nächstes soll eine Rekursion für  $J_{min}^{K+1}$  aus gegebenem  $J_{min}^K$  hergeleitet werden. Dieses Ergebnis kann dann in der  $(K + 2)$ ten Iteration des Algorithmus anstatt des Nenners von Gleichung 27 benutzt werden. Allgemein gilt für das Minimum der Kosten bei gegebener Filterordnung  $K + 1$ :

$$J_{min}^{K+1} = r_0 - \mathbf{p}_{K+1}^T \mathbf{w}_{K+1}^{K+1}$$

Wir spalten das Skalarprodukt  $\mathbf{p}_{K+1}^T \mathbf{w}_{K+1}^{K+1}$  auf (indem wir den letzten Summanden herausziehen) und erhalten:

$$J_{min}^{K+1} = r_0 - \mathbf{p}_K^T \mathbf{w}_K^{K+1} - \kappa_{K+1} r_{K+1}$$

Den Ausdruck  $\mathbf{w}_K^{K+1}$  ersetzen wir nach Gleichung 26:

$$J_{min}^{K+1} = r_0 - \mathbf{p}_K^T (\mathbf{w}_K^K - \kappa_{K+1} \tilde{\mathbf{w}}_K^K) - \kappa_{K+1} r_{K+1}$$

und multiplizieren die Klammer aus:

$$J_{min}^{K+1} = r_0 - \mathbf{p}_K^T \mathbf{w}_K^K + \kappa_{K+1} \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K - \kappa_{K+1} r_{K+1}$$

Jetzt stellen wir fest, dass wir die ersten zwei Terme nach Gleichung 28 ersetzen können:

$$J_{min}^{K+1} = J_{min}^K + \kappa_{K+1} \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K - \kappa_{K+1} r_{K+1}$$

Schließlich klammern wird noch  $\kappa_{k+1}$  aus:

$$J_{min}^{K+1} = J_{min}^K + \kappa_{K+1} (\mathbf{p}_K^T \tilde{\mathbf{w}}_K^K - r_{K+1})$$

und kehren das Vorzeichen vor und in der Klammer um:

$$J_{min}^{K+1} = J_{min}^K - \kappa_{K+1} (r_{K+1} - \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K) \quad (30)$$

Gleichung 30 stellt ein Zwischenergebnis dar. Jetzt müssen wir uns nur noch um den Term in Klammern  $r_{K+1} - \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K$  kümmern. Dazu sei hier noch einmal an Gleichung 25 erinnert:

$$\tilde{\mathbf{p}}_K^T \mathbf{w}_K^{K+1} + r_0 \kappa_{K+1} = r_{K+1}$$

Wir setzen also  $r_{K+1}$  ein:

$$J_{min}^{K+1} = J_{min}^K - \kappa_{K+1} (\tilde{\mathbf{p}}_K^T \mathbf{w}_K^{K+1} + r_0 \kappa_{K+1} - \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K)$$

Wieder ersetzen wir  $\mathbf{w}_K^{K+1}$  nach Gleichung 26:

$$J_{min}^{K+1} = J_{min}^K - \kappa_{K+1} (\tilde{\mathbf{p}}_K^T (\mathbf{w}_K^K - \kappa_{K+1} \tilde{\mathbf{w}}_K^K) + r_0 \kappa_{K+1} - \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K)$$

und multiplizieren die innere Klammer aus:

$$J_{min}^{K+1} = J_{min}^K - \kappa_{K+1} (\tilde{\mathbf{p}}_K^T \mathbf{w}_K^K - \kappa_{K+1} \tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K + r_0 \kappa_{K+1} - \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K)$$

Da  $\tilde{\mathbf{p}}_K^T \mathbf{w}_K^K = \mathbf{p}_K^T \tilde{\mathbf{w}}_K^K$  heben sich der erste und der letzte Term in der Klammer auf:

$$\begin{aligned} J_{min}^{K+1} &= J_{min}^K - \kappa_{K+1} (-\kappa_{K+1} \tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K + r_0 \kappa_{K+1}) \\ &= J_{min}^K - \kappa_{K+1} (r_0 \kappa_{K+1} - \kappa_{K+1} \tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K) \end{aligned}$$

jetzt können wir noch ein  $\kappa_{K+1}$  ausklammern, außerdem benutzen wir, dass  $\tilde{\mathbf{p}}_K^T \tilde{\mathbf{w}}_K^K = \mathbf{p}_K^T \mathbf{w}_K^K$ :

$$J_{min}^{K+1} = J_{min}^K - \kappa_{K+1}^2 (r_0 - \mathbf{p}_K^T \mathbf{w}_K^K)$$

und erstaunt stellen wir fest, dass der Ausdruck in Klammern nach Gleichung 28 wieder  $J_{min}^K$  ist. Damit erhalten wir also (nach zugegeben etwas langwieriger Rechnung):

$$J_{min}^{K+1} = J_{min}^K - \kappa_{K+1}^2 \cdot J_{min}^K$$

oder auch:

$$J_{min}^{K+1} = (1 - \kappa_{K+1}^2) J_{min}^K \quad (31)$$

Dabei ist der Startwert  $J_{min}^0 = r_0$ , denn ein Prädiktor ohne Koeffizienten hat auch kein Ausgangssignal. Demzufolge ist das Prediction-Error Signal  $e[n]$  gleich dem Signal  $x[n]$ . Da die Kostenfunktion ja als Varianz des Fehlersignals definiert war, ist sie in diesem Falle gleich der Varianz des Eingangssignals - und diese wiederum ist ja der nullte Korrelationskoeffizient. Es folgt eine Zusammenfassung der Levinson/Durbin Rekursion, wobei jedoch der Übersicht halber nicht  $\tilde{\mathbf{w}}_{K+1}^{K+1}$  aus  $\tilde{\mathbf{w}}_K^K$  berechnet wird, sondern stattdessen die äquivalente rückwärtsgerichtete Formulierung benutzt wird, so dass  $\tilde{\mathbf{w}}_K^K$  aus  $\tilde{\mathbf{w}}_{K-1}^{K-1}$  berechnet wird:

### Vorbereitung:

Berechne die Autokorrelationskoeffizienten  $r_k$  gemäß Gleichung 17:

$$r_k = \frac{1}{N} \sum_{n=0}^{N-k-1} x[n]x[n-k]$$

wobei  $k = 0, \dots, K_{max}$  mit  $K_{max}$ : gewünschte Ordnung des Prädiktors.

### Initialisierung:

Initialisiere minimale Kosten ( $\hat{=}$  Kosten ohne Prädiktor):

$$J_{min}^K = J_{min}^0 = r_0$$

Berechne ersten Reflektionskoeffizient  $\kappa_1$  nach Gleichung 29:

$$\begin{aligned} \kappa_K = \kappa_1 &= \frac{r_K - \mathbf{p}_{K-1}^T \tilde{\mathbf{w}}_{K-1}^{K-1}}{J_{min}^{K-1}} \\ &= \frac{r_1 - \mathbf{p}_0^T \tilde{\mathbf{w}}_0^0}{J_{min}^0} \\ &= \frac{r_1}{r_0} \end{aligned}$$

wobei der "leere" Term  $\mathbf{p}_0^T \tilde{\mathbf{w}}_0^0$  als Null angenommen wird ( $\mathbf{w}_0^0$  ist sozusagen der "Prädiktor" nullter Ordnung - also ein Vektor ohne Elemente)

Berechne optimalen Gewichtsvektor der Ordnung 1 nach Gleichung 19:

$$\mathbf{w}_1^1 = \frac{r_1}{r_0}$$

Berechne minimale Kosten für Prädiktor der Ordnung 1 mit Gleichung 31:

$$\begin{aligned} J_{min}^K = J_{min}^1 &= (1 - \kappa_K^2) J_{min}^{K-1} \\ &= (1 - \kappa_1^2) J_{min}^0 \end{aligned}$$

### Rekursion: für $K = 2$ bis $K = K_{max}$ :

Berechne Reflektionskoeffizient  $\kappa_K$  nach Gleichung 29:

$$\kappa_K = \frac{r_K - \mathbf{p}_{K-1}^T \tilde{\mathbf{w}}_{K-1}^{K-1}}{J_{min}^{K-1}}$$

Berechne die ersten  $K - 1$  Elemente des neuen Gewichtsvektors der Ordnung  $K$  nach Gleichung 26:

$$\mathbf{w}_{K-1}^K = \mathbf{w}_{K-1}^{K-1} - \kappa_K \cdot \tilde{\mathbf{w}}_{K-1}^{K-1}$$

Formiere den neuen optimalen Gewichtsvektor der Ordnung  $K$  durch Anhängen von  $\kappa_K$  gemäß Gleichung 23:

$$\mathbf{w}_K^K = \begin{pmatrix} \mathbf{w}_{K-1}^K \\ \kappa_K \end{pmatrix}$$

Berechne die neuen minimalen Kosten für den Prädiktor der Ordnung  $K$  nach 31:

$$J_{min}^K = (1 - \kappa_K^2) J_{min}^{K-1}$$

Oft findet man auch eine äquivalente Form dieser Rekursion ohne die Vektorschreibweise. Diese sei hier der Vollständigkeit halber ebenfalls angegeben (in etwas weniger ausführlicher Weise). Dabei soll mit  $w_j^K$  das  $j$ -te Element eines optimalen Gewichtsvektors der Ordnung  $K$  gemeint sein:

**Vorbereitung:**

wie oben

**Initialisierung:**

Initialisiere minimale Kosten ( $\hat{=}$  Kosten ohne Prädiktor):

$$J_{min}^0 = r_0$$

Berechne ersten Reflektionskoeffizient  $\kappa_1$ :

$$\kappa_1 = \frac{r_1}{r_0}$$

Berechne optimalen ein-elementigen Prädiktor:

$$w_1^1 = \frac{r_1}{r_0}$$

Berechne minimale Kosten für Prädiktor der Ordnung 1:

$$J_{min}^1 = (1 - \kappa_1^2) J_{min}^0$$

**Rekursion:** für  $K = 2$  bis  $K = K_{max}$ :

Berechne Reflektionskoeffizient  $\kappa_K$ :

$$\kappa_K = \frac{1}{J_{min}^{K-1}} \left( r_K - \sum_{j=1}^{K-1} w_j^{K-1} r_{K-j} \right)$$

Berechne die ersten  $K - 1$  Elemente des neuen optimalen Gewichtsvektors der Ordnung  $K$  mit:

$$w_j^K = w_j^{K-1} - \kappa_K w_{K-j}^{K-1} \quad j = 1, \dots, K - 1$$

Setze  $K$ -tes Element des neuen optimalen Gewichtsvektors:

$$w_K^K = \kappa_K$$

Berechne die neuen minimalen Kosten für den Prädiktor der Ordnung  $K$ :

$$J_{min}^K = (1 - \kappa_K^2) J_{min}^{K-1}$$

## 5 Anwendungen der Linearen Prädiktion

### 5.1 Quelle-Filter Modell

In diesem Kapitel soll geklärt werden, warum lineare Prädiktion sich zur Spektralanalyse im allgemeinen und speziell zur Extraktion von Formanten eignet. Dazu gehen wir von einem bestimmten Modell der Entstehung unseres Signal aus - wir nehmen nämlich an, dass das Signal von irgendeiner Quelle erzeugt wurde, die ein weißes Spektrum aufweist (ein Spektrum also, in dem alle Frequenzen mit gleicher Amplitude enthalten sind). Dieses Quellsignal - so unsere Modellannahme - wurde dann von einem Filter im Frequenzgang geformt. Dieses Signalmodell würde (auf digitaler Ebene) folgendermaßen aussehen:

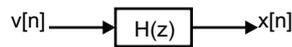


Figure 4: Signalmodell

Dabei ist  $v[n]$  das Anregungssignal und  $H(z)$  ein Filter. Wenn wir für dieses Filter annehmen, dass es ein rein rekursives IIR-Filter ist ("rein rekursiv" soll hier bedeuten, dass alle feedforward-Koeffizienten Null sind), dann kann unser Signal durch folgende Differenzgleichung beschrieben werden:

$$x[n] = v[n] + \sum_{k=1}^K a_k x[n-k] \quad (32)$$

$v[n]$  ist also das Anregungssignal, die Summe entspricht der IIR-Filterung und  $x[n]$  ist unser gefiltertes Signal. Als Blockschaltbild würde diese Differenzgleichung folgendermaßen aussehen:

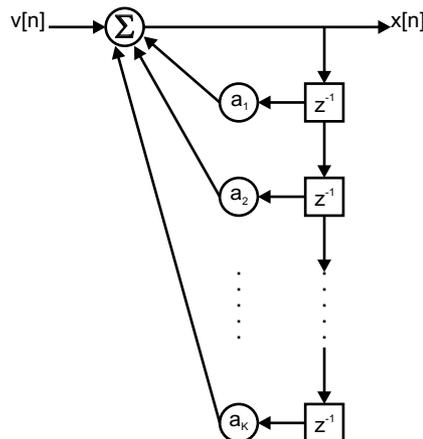


Figure 5: allgemeine Struktur eines rein rekursiven IIR-Filters der Ordnung  $K$

Ein solches Signalmodell wird auch als autoregressives (AR) Modell bezeichnet, da das Filter ausschließlich seine eigenen vergangenen Ausgangswerte zur Filterung benutzt (im Gegensatz zu Filtern, die auch vergangene Eingangswerte benutzen). Ein solches IIR-Filter wird in der  $z$ -Ebene ausschließlich durch seine Polstellen bestimmt - man spricht deshalb auch von einem Allpol-Filter. Da Polstellen in der  $z$ -Ebene mit spektralen Peaks einhergehen, sollte ein solches Filter gut geeignet sein, um Formanten zu erzeugen. Daher können wir für Signale, die sich hauptsächlich durch ihre Formanten auszeichnen, annehmen, dass die rein rekursive Modellierung von  $H(z)$  adäquat ist. Wenn unser Gesamtsignal  $x[n]$  tatsächlich durch rein rekursive Filterung eines Anregungssignals entstanden ist, dann besteht es aus zwei Anteilen: dem Anregungssignal

(welches nicht von vergangenen Werten von  $x[n]$  abhängt) und dem rückgekoppelten Ausgangssignal (welches ausschließlich von seinen eigenen vergangenen Werten abhängt). Wir können den rückgekoppelten Anteil demzufolge als vorhersagbar aus vergangenen Werten betrachten. Alles was wir für die Vorhersage des vorhersagbaren Anteils brauchen, sind die Koeffizienten  $a_k$ . Und genau diese Koeffizienten werden mit Hilfe der linearen Prädiktion geschätzt - die Minimierung der Varianz des Prediction-Error Signals bewirkt nämlich, dass der vorhersagbare Teil des Signals  $x[n]$  wieder abgezogen wird. Dies funktioniert natürlich nur, wenn die Prädiktorkoeffizienten  $w_k$  gleich den Filterkoeffizienten  $a_k$  sind. Man mache sich diese Subtraktion des vorhersagbaren Anteils anhand von Figure 6 klar.

## 5.2 Inverse Filterung

Aus Gleichung 32 ist ersichtlich, dass sich das Anregungssignal  $v[n]$  als Differenz zwischen Gesamtsignal  $x[n]$  und vorhersagbarem Anteil darstellen lässt:

$$v[n] = x[n] - \sum_{k=1}^K a_k x[n-k]$$

Man vergleiche dies mit Gleichung 16 für das Prediction-Error Signal, die hier nochmal angegeben wird:

$$e[n] = x[n] - \sum_{k=1}^K w_k x[n-k]$$

Wenn die  $w_k$  aus Gleichung 16 genau den  $a_k$  entsprechen, dann ist das Prediction-Error Signal  $e[n]$  genau das Anregungssignal  $v[n]$  in unserem Modell. Die Minimierung der Varianz des Prediction-Error Signals (also die Suche nach dem "optimalen" Gewichtsvektor) bewirkt - wie gesagt - tatsächlich, dass die Koeffizienten  $w_k$  die Werte der  $a_k$  annehmen - natürlich immer noch unter der Voraussetzung, dass das Signalmodell richtig ist und das Eingangssignal  $v[n]$  ein weisses Spektrum aufweist. Wenn das Anregungssignal z.B. ein weisses Rauschen wäre, wäre diese Bedingung erfüllt. Aber auch ein idealer Impulskamm hat ein weisses Spektrum (es treten zwar nur Harmonische auf, diese haben aber alle die gleiche Amplitude). Bei Sprachsignalen und vielen Instrumentenklängen kann das Anregungssignal tatsächlich näherungsweise entweder als Impulskamm oder weisses Rauschen angesehen werden - oder zumindest kann durch geeignete Vorfilterung (Pre-Emphase) ein Signal erzeugt werden, welches die spektralen Abweichungen des Anregungssignals vom idealen Impulskamm bzw. idealen weißen Rauschen kompensiert. Damit ist das FPE-Filter  $A(z)$  genau das inverse Filter zu dem Filter  $H(z)$ , welches unser Signal geformt hat. Da das Ausgangssignal eines solchen FPE-Filters wieder ein weißes Spektrum hat (es entspricht ja dem geschätzten Anregungssignal), wird das FPE-Filter auch als Whitening-Filter bezeichnet.

## 5.3 Spektralanalyse

Da  $A(z)$  das inverse Filter zu  $H(z)$  ist, erhalten wir, wenn wir wiederum  $A(z)$  invertieren, schließlich eine Schätzung für  $H(z)$  (welche mit  $\hat{H}(z)$  bezeichnet werden soll). Wie gut diese Schätzung tatsächlich ist, hängt stark davon ab, wie gut unser Signalmodell mit der tatsächlichen Entstehung des analysierten Signals übereinstimmt. Um den Frequenzgang des Filters  $\hat{H}(z)$  zu bestimmen haben wir mehrere Möglichkeiten. Betrachten wir zunächst die die  $z$ -transformierte des FPE-Filters (man kann z.B. sie aus dem Blockschaltbild von  $A(z)$  in Figure 6 ablesen):

$$A(z) = 1 - \sum_{k=1}^K w_k z^{-k}$$

Die  $z$ -transformierte des inversen FPE-Filters ist daher gegeben mit:

$$\begin{aligned} \hat{H}(z) &= A^{-1}(z) \\ &= \frac{1}{1 - \sum_{k=1}^K w_k z^{-k}} \end{aligned}$$

Um den Wert der Übertragungsfunktion bei einer bestimmten normierten Kreisfrequenz  $\Omega$  (mit  $\Omega = 2\pi f/f_s$ ) zu erhalten ersetzen wir einfach  $z$  in obiger Gleichung mit  $e^{j\Omega}$  und erhalten so  $\hat{H}(\Omega)$ . Dabei kann  $\Omega$  eine beliebige Kreisfrequenz zwischen 0 und  $\pi$  sein ( $\pi$  entspricht von der Frequenz her der halben Samplerate). Da wir also (zumindest soweit das in den, durch das Sampling-Theorem vorgegebenen, Grenzen sinnvoll ist) beliebige Frequenzen einsetzen können, erhalten wir ein kontinuierliches Spektrum, d.h. das Spektrum ist nicht auf ganz bestimmte Spektrallinien ("bins") begrenzt, wie es z.B. bei einem FFT-Spektrum der Fall wäre. Als zweite Möglichkeit könnten wir auch das Filter  $A^{-1}(z)$  direkt im Zeitbereich implementieren. Aus der  $z$ -transformierten gewinnen wir folgende Vorschrift zur Berechnung des Filterausgangssignals:

$$y[n] = x[n] + \sum_{k=0}^K w_k y[n-k]$$

Man beachte den Vorzeichenwechsel der  $w_k$  gegenüber der Darstellung in der  $z$ -Transformierten. Ein digitales Filter zu invertieren ist also ziemlich einfach: einfach feedback-Koeffizienten gegen feedforward-Koeffizienten vertauschen und die Vorzeichen der Koeffizienten umkehren. Da  $A(z)$  ein FIR-Filter ist, erhalten wir mit  $\hat{H}(z) = A^{-1}(z)$  wieder ein rein rekursives IIR-Filter. Wenn wir nun die Impulsantwort unseres Modellfilters  $\hat{H}(z)$  nehmen (indem wir einfach einen Impuls hineinschicken und das Ausgangssignal aufzeichnen) und diese mit Hilfe der FFT im Spektralbereich betrachten, dann sehen wir den (geschätzten) Frequenzgang des Filters, welches unser Signal geformt hat. Dieses Filter können wir jetzt auch zu Resynthese benutzen. Wenn man nur an der Spektralanalyse und nicht an der Resynthese interessiert ist, dann braucht man das FPE-Filter nicht explizit zu invertieren und dessen Impulsantwort zu messen - man kann stattdessen auch direkt die Impulsantwort des FPE-Filters nehmen (also einfach die FPE-Koeffizienten, da das FPE-Filter ja ein FIR-Filter ist), diese Fourier-transformieren und dann das Spektrum invertieren. In jedem der drei Fälle erhalten wir also den Frequenzgang unseres Modellfilters - und dieser Frequenzgang ist unabhängig vom Anregungssignal  $v[n]$ .

## 5.4 Resynthese und Vocoder-Sounds

Mit Hilfe der linearen Prädiktion kann also ein Signal in eine Schätzung des Anregungssignals  $v[n]$  und eine Schätzung eines nachgeschalteten (Allpol)Filters  $H(z)$  zerlegt werden. Dabei ist die Schätzung des Anregungssignals gleich dem Prediction-Error Signal:  $\hat{v}[n] = e[n]$  und die Schätzung des Filters ist das inverse Prediction-Error Filter (FPE-Filter):  $\hat{H}(z) = A^{-1}(z)$ . Um das Signal  $x[n]$  zurückzugewinnen (Resynthese) müssen wir einfach nur  $\hat{v}[n]$  mit  $\hat{H}(z)$  filtern. Ein Blockschaltbild für einen solchen Analyse/Resynthese-Vorgang ist in Figure 6 gegeben. Unter der Voraussetzung, dass die  $a_k$  mit den  $w_k$  übereinstimmen, subtrahiert das FPE-Filter  $A(z)$  genau diejenigen verzögerten und gewichteten Werte von  $x[n]$ , die das Synthesefilter  $H(z)$  hinzugefügt hat. Damit ist es tatsächlich das inverse Filter zu  $H(z)$ . Dann ist natürlich auch  $e[n] = v[n]$ . Schließlich kann das Signal  $x[n]$  exakt rekonstruiert werden mit Hilfe von  $\hat{H}(z) = A^{-1}(z)$ . Die exakte Resynthese von  $x[n]$  funktioniert übrigens immer - egal ob das Signalmodell stimmt oder nicht. Denn schließlich ist ja  $\hat{H}(z) = A^{-1}(z)$  immer das inverse Filter von  $A(z)$  - und zwar unabhängig davon, wie gut  $\hat{H}(z)$  das Originalfilter  $H(z)$  repräsentiert oder auch nicht. Wir können jetzt jedoch auch andere Anregungssignale durch unser Modellfilter  $\hat{H}(z)$  schicken. Auf diese Weise würde das Anregungssignal einen neuen Frequenzgang aufgeprägt bekommen und zwar denjenigen unseres Analysesignals. Damit haben wir einen Vocoder auf Basis der Linearen Prädiktion.

## 5.5 Formantverläufe und nichtstationäre Signale

Ein Punkt, der in der bisherigen Darstellung noch nicht berücksichtigt wurde, ist, dass die gesamte Herleitung der Bestimmung der optimalen Prädiktionskoeffizienten auf der Annahme beruht, dass das zu analysierende Signal ein stationärer stochastischer Prozess ist, dessen Autokorrelationsmatrix bekannt, bzw. messbar ist. Nun ist z.B. Sprache aber ein hochgradig nichtstationärer Prozess - die Formanten (und damit die Autokorrelationsmatrix) ändern sich ständig, sind also Funktionen der Zeit. Jedoch ändern sich die Parameter des "Prozesses" Sprache aufgrund der Trägheit der Zunge nicht beliebig schnell. Das heißt, man kann Sprache

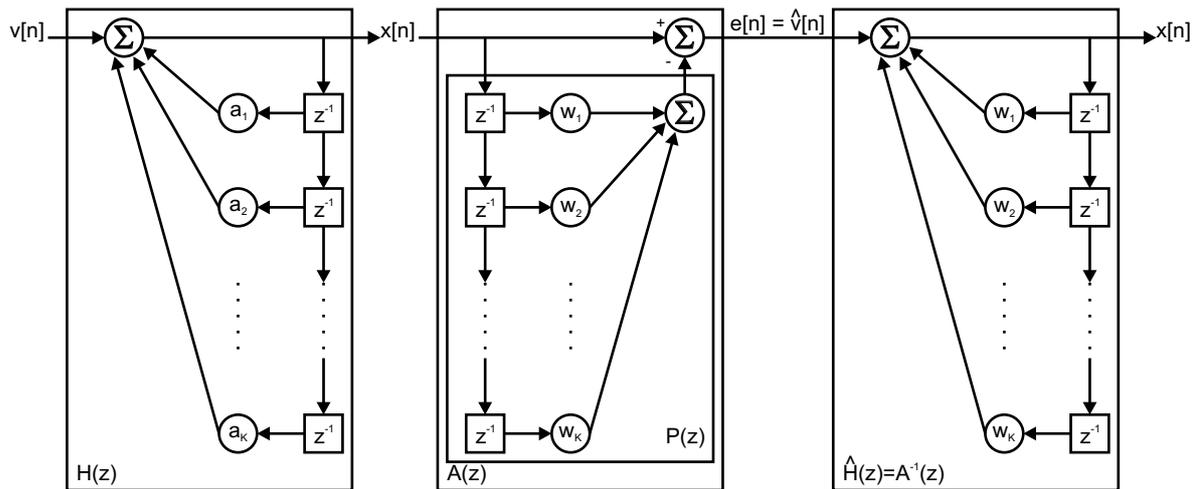


Figure 6: Analyse und Resynthese (von  $x[n]$ ) mit Hilfe der linearen Prädiktion - man beachte, wie im FPE-Filter  $A(z)$  genau derjenige Teil von  $x[n]$  subtrahiert wird, der durch  $H(z)$  addiert wurde (wenn  $w_k = a_k \quad \forall k$ )

(und auch viele Instrumentenklänge) in einem hinreichend kleinen Zeitfenster als quasi-stationär annehmen. Bei der Sprachanalyse sollte das Zeitfenster im Bereich von 10-20 ms liegen. Für jedes einzelne Zeitfenster werden dann die Autokorrelationskoeffizienten  $r_k$  ermittelt und dann die Levinson/Durbin Rekursion gestartet. Anders sieht es aus, wenn man mit dem LMS-Algorithmus arbeitet: da hier sowieso ständig ein samplebasierter Adaptionsprozess mitläuft, ist dieser Algorithmus in der Lage, den Instationaritäten des Signals automatisch zu folgen - wie schnell er den Änderungen des Signals folgen kann, hängt von der Adaptionsrate  $\eta$  ab. LMS hat also den Vorteil, dass die etwas unnatürlich erscheinende Segmentierung des Signals wegfällt - das ist besonders beim Vocoding sehr angenehm, wenn man im Ausgangssignal keine stufenförmigen Filterhüllkurven haben will. Bei der Spektralanalyse ist jedoch erfahrungsgemäß Levinson/Durbin genauer.

## 5.6 Sprachcodierung

Mit Hilfe der Linearen Prädiktion kann man Sprache auch sehr effizient codieren (man spricht dann von **Linear Prediction Coding**, kurz LPC). Dies kann z.B. geschehen, indem man das Sprachsignal in kleine Zeitabschnitte zerlegt (10-20 ms) und für diese Zeitabschnitte anstatt der tatsächlichen Signal-Samples nur einen Satz von Filterkoeffizienten plus einem sehr grob quantisierten Prediction-Error Signal übermittelt. Es kann statt dem Error-Signal aber auch einfach eine Information über Grundfrequenz, Stimmhaftigkeit und Signalamplitude übermittelt werden, aus der dann von einem entsprechenden Synthesizer auf der Empfangsseite wieder ein passendes Anregungssignal generiert wird. Die meisten digitalen (Mobil)Telefone benutzen eine solche Codierung um Sprache zu komprimieren - da die Hauptinformation bei Sprache in den Formanten steckt, welche durch die LPC-Koeffizienten recht gut repräsentiert werden, bleibt die Sprachverständlichkeit auch bei sehr hohen Komprimierungsfaktoren noch erhalten. Auch die verlustlose Audiokomprimierung im kommenden MPEG-4 Standard wird auf der Linearen Prädiktion basieren. Die Komprimierung wird dabei durch eine geschickte Quantisierung und Codierung des Prediction-Error Signals erreicht.

## 6 Literatur

- Alexander, S. Thomas: Adaptive Signal Processing - Theory And Applications
- Haykin, Simon: Adaptive Filter Theory