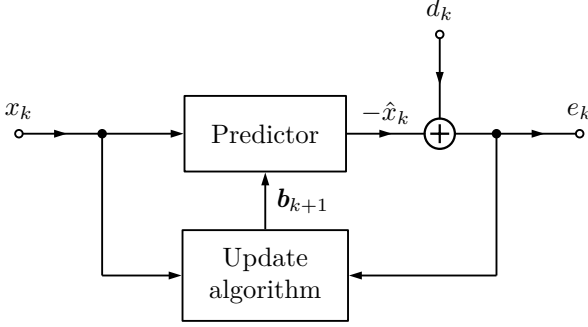


Contents

1	Aptive Filter	3
1.1	Filter structure	3
1.2	Predictor	3
1.3	Update Algorithm	3
1.4	Echo and Noise cancellation	6
2	Prädiktion	7
2.1	Linearer Prädiktor	7
2.2	Bestimmung des Schätzfehlers	8
2.3	Optimale Filterkoeffizienten	9
2.4	Levinson-Durbin-Algorithmus	9
2.5	Least-Mean-Square-Algorithmus	11
2.6	Effiziente Varianz Berechnung	13
2.7	Störunterdrückung	14
2.8	Dekorrelation	16
2.8.1	Dekorrelation mittels Prädiktion	16
2.8.2	Quantisierte Prädiktion	17
2.8.3	Signalrückgewinnung	18
2.8.4	Prädiktionsgewinn	18

1 Adaptive Filter

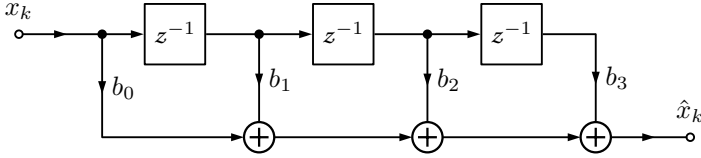
1.1 Filter structure



1.2 Predictor

$$\hat{x}_k = \sum_{i=0}^N b_i x_{k-i}$$

$$H(z) = \frac{\hat{X}(z)}{X(z)} = \sum_{i=0}^N b_i z^{-i} = b_0 + b_1 z^{-1} + \dots + b_N + z^{-N}$$



$$e_k = d_k - \hat{x}_k \quad (1.1)$$

1.3 Update Algorithm

Least Mean Square algorithm

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \beta \frac{\partial E(e_k^2)}{\partial \mathbf{b}_k}$$

$$E(e_k^2) = \frac{1}{N} \sum_{k=0}^{N-1} (x_k - \mathbf{b}_k \mathbf{X}_k)^2 = E\left((x_k - \mathbf{b}_k \mathbf{X}_k)^2\right)$$

$$E(e_k^2) = E(x_k^2) - 2\mathbf{b}_k^T E(\mathbf{X}_k) + \mathbf{b}_k^T E(\mathbf{X}_k^2) \mathbf{b}_k$$

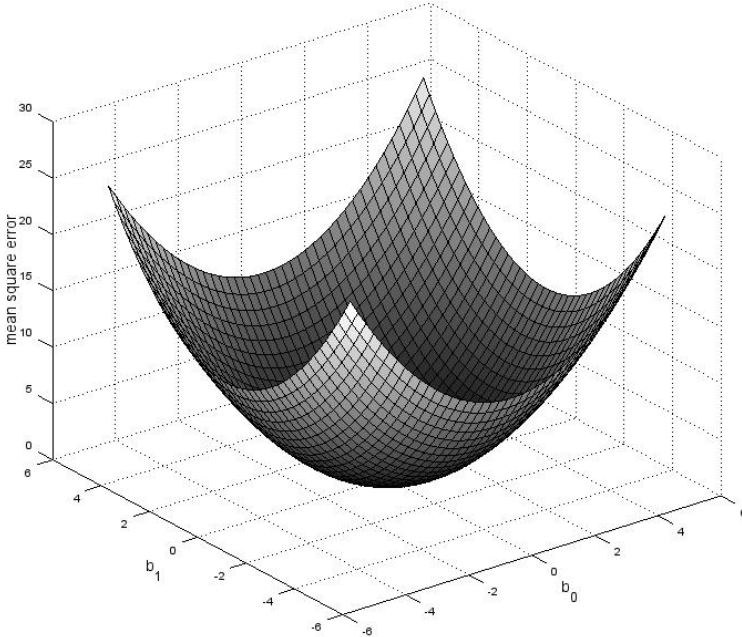
$$\frac{\partial E(e_k^2)}{\partial \mathbf{b}_k} = -2E(\mathbf{X}_k) + 2\mathbf{b}_k^T E(\mathbf{X}_k^2)$$

$$\frac{\partial E(e_k^2)}{\partial \mathbf{b}_k} = -2 \underbrace{E(\mathbf{X}_k)}_{\text{mean value free}} + 2\mathbf{b}_k^T \underbrace{E(\mathbf{X}_k^2)}_{\mathbf{R}_{xx,k}}$$

$$\frac{\partial E(e_k^2)}{\partial \mathbf{b}_k} = 2\mathbf{R}_{xx,k} \mathbf{b}_k$$

$$\frac{\partial E(e_k^2)}{\partial \mathbf{b}_k} \approx \frac{\Delta E(e_k^2)}{\Delta \mathbf{b}_k} = 2\mathbf{R}_{xx,k} (\mathbf{b}_k - \mathbf{b}_{k-1})$$

$$\mathbf{b}_{k+1} = \mathbf{b}_k - 2\beta \mathbf{R}_{xx,k} (\mathbf{b}_k - \mathbf{b}_{k-1})$$



Optimum reached by $\mathbf{b}_{k+1} = \mathbf{b}_{k-1}$

$$\beta = \frac{1}{2} I \mathbf{R}_{xx,k}^{-1} \xrightarrow{n \times n} \beta = \frac{1}{2} \mathbf{R}_{xx,k}^{-1}$$

$$0 < \beta < \mathbf{R}_{xx,k}^{-1}$$

Convergence at sum of $\mathbf{R}_{xx,k}$ intrinsic values

$$(N+1) R_{xx}(0) = (N+1) \sigma_{xx}^2$$

$$0 < \beta < \frac{1}{(N+1) \sigma_{xx}^2}$$

$$\frac{\partial E(e_k^2)}{\partial \mathbf{b}_k} = E \left(\frac{\partial e_k^2}{\partial \mathbf{b}_k} \right)$$

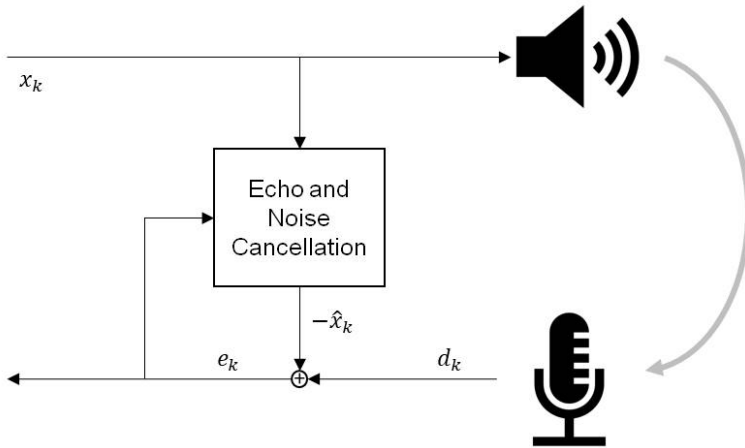
$$\frac{\partial e_k^2}{\partial \mathbf{b}_k} = 2e_k \frac{de_k}{d\mathbf{b}_k}$$

$$e_k = x_k - \mathbf{b}_k \mathbf{X}_k$$

$$\frac{\partial e_k^2}{\partial \mathbf{b}_k} = 2e_k(-\mathbf{X}_k)$$

$$\frac{\partial E(e_k^2)}{\partial \mathbf{b}_k} = -2E(e_k \mathbf{X}_k) = -2e_k \frac{1}{N} \sum_{i=0}^{N-1} x_{k-i} \xrightarrow{N=1} -2e_k x_k$$

1.4 Echo and Noise cancellation



2 Prädiktion

Viele Aufgaben in der digitalen Signalverarbeitung werden mit der Prädiktion realisiert. Dazu zählen Anwendungen in der Audio- und Videocodierung, Vorhersagen von Aktienkursen, Störunterdrückung und überall dort wo Schätzungen von Signalwerten durchgeführt werden müssen.

2.1 Linearer Prädiktor

Die Prädiktion schätzt mittels Linearkombination aus einem vorangegangenen Signal neue Werte. Die Linearkombination kann durch folgende Differenzengleichung ausgedrückt werden:

$$\hat{x}_k = \sum_{i=0}^N b_i x_{k-i} \quad (2.1)$$

Die zugehörige zeitdiskrete Übertragungsfunktion lautet nach der z-Transformation:

$$H(z) = \frac{\hat{X}(z)}{X(z)} = \sum_{i=0}^N b_i z^{-i} = b_0 + b_1 z^{-1} + \dots + b_N z^{-N}$$

Diese Übertragungsfunktion lässt sich auch als FIR-Filter mit der zugehörigen Struktur hier als Beispiel mit $N=3$ darstellen:

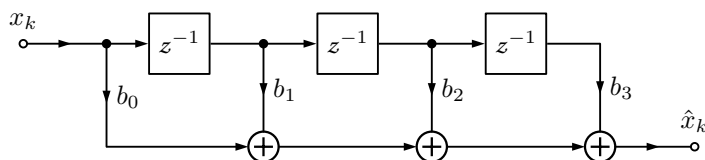


Figure 2.1: FIR-Filterstruktur für $N=3$

Um das Prädiktionsfilter auf dessen Genauigkeit beurteilen zu können wird der Originalwert x_k mit dem geschätzten Wert \hat{x}_k verglichen. In Abbildung 2.2 wird die FIR-Filterstruktur als Übertragungsfunktion $H(z)$ dargestellt.

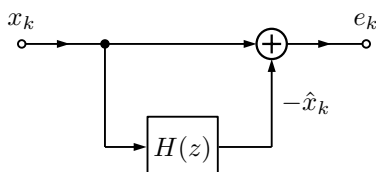


Figure 2.2: Struktur eines linearen Prädiktors

Im Idealfall ist die Differenz aus bekanntem Wert und dem geschätzten Wert gleich Null. In Realität jedoch erhält man einen Fehlerwert

$$e_k = x_k - \hat{x}_k \quad (2.2)$$

2.2 Bestimmung des Schätzfehlers

An zentraler Stelle steht zur Bestimmung von Schätzwerten die Berechnung des Erwartungswertes. Der Erwartungswert beschreibt allgemein den Mittelwert einer Wertefolge \mathbf{X} :

$$E(\mathbf{X}) = \sum_{i=0}^{N-1} p_i x_i$$

Wobei p_i die Wahrscheinlichkeit eines Wertes der Wertefolge darstellt. Für Erwartungswerte eines Produktes aus 2 stochastik unabhängigen Wertefolgen \mathbf{X}_i und \mathbf{X}_j gilt der Zusammenhang:

$$E(\mathbf{X}_i \mathbf{X}_j) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} p_i p_j x_i x_j \quad (2.3)$$

Handelt es sich bei einer Wertefolge \mathbf{X} bestehend aus x -Elementen um einen stationären Prozess, so kann der Erwartungswert aus dem Produkt der Werte von x mit Hilfe der Korrelationsfunktion $R_{xx}(\tau)$ berechnet werden.

$$E(x_k^2) = E(x_k) E(x_k) = R_{xx}(0) \quad (2.4)$$

$$E(x_k x_{k-i}) = E(x_k) E(x_{k-i}) = R_{xx}(i) \quad (2.5)$$

$$E(x_i x_j) = E(x_i) E(x_j) = R_{xx}(i-j) \quad (2.6)$$

Quadiert man den Schätzfehler und berechnet man daraus den Erwartungswert erhält man den mittleren quadratischen Fehler $E(e_k^2)$.

$$E(e_k^2) = E((x_k - \hat{x}_k)^2) = E(x_k^2 - 2x_k \hat{x}_k + \hat{x}_k^2)$$

Aus der linearen Eigenschaft der Erwartungswerte kann die Gleichung umgeformt werden zu:

$$E(e_k^2) = E(x_k^2 - 2x_k \hat{x}_k + \hat{x}_k^2) = E(x_k^2) - 2E(x_k \hat{x}_k) + E(\hat{x}_k^2) \quad (2.7)$$

Jetzt kann man die Gleichung mit den Beziehungen (2.1) und (2.4) in der Abhängigkeit von den Faktoren b_k und den Signalwerten x_k ausschreiben.

$$E(e_k^2) = R_{xx}(0) - 2E\left(\sum_{i=0}^N b_i x_{k-i} x_k\right) + E\left(\left(\sum_{i=0}^N b_i x_{k-i}\right)^2\right)$$

Mit der Beziehung aus (2.3) entfällt der quadratische Teil.

$$E(e_k^2) = R_{xx}(0) - 2E\left(\sum_{i=0}^N b_i x_{k-i} x_k\right) + E\left(\sum_{i=0}^N \sum_{j=0}^N b_i b_j x_{k-i} x_{k-j}\right)$$

Wie bei (2.7) mittels der linearen Eigenschaft der Erwartungswerte und der Korrelationsbeziehungen aus (2.5) und (2.6) entfällt die Abhängigkeit der Signalwerte x_k .

$$E(e_k^2) = R_{xx}(0) - 2 \sum_{i=0}^N b_i E(x_{k-i} x_k) + \sum_{i=0}^N \sum_{j=0}^N b_i b_j E(x_{k-i} x_{k-j})$$

$$E(e_k^2) = R_{xx}(0) - 2 \sum_{i=0}^N b_i R_{xx}(i) + \sum_{i=0}^N \sum_{j=0}^N b_i b_j R_{xx}(j-i)$$

In Matrix schreibweise erhält man für den mittleren quadratischen Fehler nun:

$$E(e_k^2) = R_{xx}(0) - 2 \mathbf{b}^T \mathbf{r}_{xx} + \mathbf{b}^T \mathbf{R}_{xx} \mathbf{b} \quad (2.8)$$

2.3 Optimale Filterkoeffizienten

Um den kleinsten mittleren quadratischen Fehler (engl. least mean square = LMS) zu ermitteln wendet man die Extremwertbetrachtung über die erste Ableitung an. Bei der obigen Gleichung handelt es sich um eine Gleichung zweiter Ordnung, somit wird sich nach der ersten Ableitung eine lineare Gleichung ergeben.

$$\frac{dE(e_k^2)}{d\mathbf{b}} = -2\mathbf{r}_{xx} + 2\mathbf{R}_{xx}\mathbf{b} \quad (2.9)$$

Der Schnittpunkt mit der x-Achse zeigt den Extremwert.

$$\frac{dE(e_k^2)}{d\mathbf{b}} = 0$$

Die Koeffizientenmatrix \mathbf{b} erhält man demnach mit (2.10). Die Koeffizienten sind an dieser Stelle optimal. Der Rechenaufwand ist enorm, da jeweils für N-Werte die Korrelationsmatrizen \mathbf{R} berechnet werden müssen.

$$\mathbf{b}_{opt} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{xx} \quad (2.10)$$

2.4 Levinson-Durbin-Algorithmus

Der Levinson-Durbin-Algorithmus löst das lineare Gleichungssystem (2.10). Nach \mathbf{r}_{xx} umgestellt

$$\mathbf{R}_{xx} \mathbf{b}_{opt} = \mathbf{r}_{xx}$$

bekommt man das Gleichungssystem

$$\begin{bmatrix} R_{xx}[0] & R_{xx}[1] & \cdots & R_{xx}[N] \\ R_{xx}[-1] & R_{xx}[0] & \cdots & R_{xx}[N-1] \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}[-N] & \cdots & \cdots & R_{xx}[0] \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_N \end{bmatrix} = \begin{bmatrix} -R_{xx}[1] \\ -R_{xx}[2] \\ \vdots \\ -R_{xx}[N+1] \end{bmatrix}$$

Im Levinson-Durbin-Algorithmus werden die Koeffizienten in rekursiver Weise mit zunehmender Ordnung aus (2.11) berechnet. Für die Koeffizienten eines Prädiktors p -ter Ordnung gilt folgender Zusammenhang:

$$\alpha_i^{(p)} = -b_{i-1}^{(p)} \quad \text{für } i = 1, \dots, p \quad (2.11)$$

Mit dem mittleren quadratischen Fehler bei optimalen Filterkoeffizienten aus (2.8) und (2.10)

$$E(e_k^2)_{opt} = R_{xx}[0] - \mathbf{b}_{opt}^T \mathbf{r}_{xx} \quad (2.12)$$

und obigem Zusammenhang (2.11) erhält man das Gleichungssystem

$$\begin{bmatrix} R_{xx}[0] & R_{xx}[1] & R_{xx}[2] \\ R_{xx}[1] & R_{xx}[0] & R_{xx}[1] \\ R_{xx}[2] & R_{xx}[1] & R_{xx}[0] \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} E(e_k^2)_{opt} \\ 0 \\ 0 \end{bmatrix}$$

exemplarisch für einen Prädiktor 2. Ordnung. Wobei hierbei der einfache Zusammenhang nach (2.11) gilt. Zur Lösung des Gleichungssystems und zur Ermittlung der Koeffizienten α muss eine Konstante k eingeführt werden.

$$\begin{bmatrix} 1 \\ \alpha_1^2 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1^1 + k_2 \alpha_1^1 \\ k_2 \end{bmatrix}$$

Verallgemeinert man die Linearkombination der Lösung kann man den Zusammenhang so

$$\begin{aligned} \alpha_0^{(p)} &= 1 \\ \alpha_m^{(p)} &= \alpha_m^{(p-1)} + k_p \cdot \alpha_{p-m}^{(p-1)} \quad \text{für } m = 1 : p-1 \text{ und } p \geq 2 \\ \alpha_p^{(p)} &= k_p \end{aligned}$$

aufstellen. Setzt man obige Linearkombination in das Gleichungssystem eines Prädiktors 2. Ordnung ein, erhält man:

$$\begin{bmatrix} R_{xx}[0] + \alpha_1 R_{xx}[1] + k_2 (\alpha_1 R_{xx}[1] + R_{xx}[2]) \\ R_{xx}[1] + \alpha_1 R_{xx}[0] + k_2 (\alpha_1 R_{xx}[0] + R_{xx}[1]) \\ R_{xx}[2] + \alpha_1 R_{xx}[1] + k_2 (\alpha_1 R_{xx}[1] + R_{xx}[0]) \end{bmatrix} = \begin{bmatrix} E(e_k^2)_{opt} \\ 0 \\ 0 \end{bmatrix} \quad (2.13)$$

Mit der Einführung einer weiteren Zwischengröße q

$$q = R_{xx}[2] + \alpha_1 R_{xx}[1]$$

und aus der Lösung der mittleren Gleichung von (2.13) für den Prädiktor 1. Ordnung

$$\alpha_1^1 = -\frac{R_{xx}[1]}{R_{xx}[0]} \quad (2.14)$$

kann man das Gleichungssystem folgendermassen zum Levinson-Durbin-Algorithmus verallgemeinern:

$$\begin{aligned} q^{(p-1)} &= R_{xx}[p] + \sum_{m=1}^{p-1} \alpha_m^{(p-1)} \cdot R_{xx}[p-m] \\ E(e_k^2)_{opt}^{(p-1)} &= R_{xx}[0] + \sum_{m=1}^{p-1} \alpha_m^{(p-1)} \cdot R_{xx}[m] \quad \text{für } m = 1 : p-1 \text{ und } p \geq 2 \\ k_p &= -\frac{q^{(p-1)}}{E(e_k^2)_{opt}^{(p-1)}} \\ E(e_k^2)_{opt}^{(p)} &= E(e_k^2)_{opt}^{(p-1)} \cdot (1 - k_p^2) \end{aligned}$$

Das folgende Matlab Programm implementiert den Levinson-Durbin-Algorithmus. Zuerst werden die Variablen mit 0 initialisiert. Der Startwert für α ergibt sich aus der Lösung der mittleren Zeile von (2.13) für einen Prädiktor 1.Ordnung aus (2.14). Die anderen Berechnungen ergeben sich direkt aus obigen Gleichungen. Eine Anwendung des Algorithmus wird in Kapitel 2.8 bezüglich Dekorrelation gezeigt.

2.5 Least-Mean-Square-Algorithmus

Eine anderer Methode die Prädiktor-Koeffizienten zu berechnen stellt der Least-Mean-Square-Algorithmus dar. Hierbei handelt es sich um eine Näherungsverfahren um den Rechenaufwand klein zu halten. Ein Verfahren ist die Methode des steilsten Abstiegs oder auch Gradientenverfahren genannt. Damit nähert man sich sozusagen den optimalen Koeffizienten an. Man beginnt quasi mit einem Satz Koeffizienten an der Stelle $E(e_k^2)|_{b_{start}}$ und nähert sich dem kleinsten mittleren quadratischen Fehler $E(e_k^2)_{opt}$ an (vgl. Abbildung ??).

Allgemein ist das Gradientenverfahren definiert mit:

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \mu \mathbf{G}_k \quad (2.15)$$

Wobei $-\mathbf{G}$ den Gradienten für den Abstieg und μ die Schrittweite darstellt. Für die Berechnung des Gradienten wird nochmal die Berechnung des Schätzfehlers betrachtet. Aus (2.1) und (2.2) erhält man in Matrixschreibweise

$$e_k = x_k - \mathbf{b}_k \mathbf{X}_k. \quad (2.16)$$

Der Gradient an der Stelle \mathbf{b}_k wird über die erste Ableitung mit Hilfe der Kettenregel des mittleren quadratischen Fehlers bestimmt.

$$\mathbf{G}_k = \nabla_{\mathbf{b}_k} E(e_k^2) = E \left(2e_k \frac{de_k}{d\mathbf{b}_k} \right) \quad (2.17)$$

Setzt man (2.16) in (2.17) ein ist das Resultat:

$$\mathbf{G}_k = -2E(e_k \mathbf{X}_k)$$

Für den Erwartungswert geht man einfachshalber von einem Mittelwert der Eingangswerte aus und betrachtet nur $N=1$ Werte.

$$\mathbf{G}_k = -2 e_k \frac{1}{N} \sum_{i=0}^{N-1} x_{k-i} \xrightarrow{N=1} -2 e_k x_k$$

Somit erhält man für das Gradientenverfahren folgenden Algorithmus

$$\mathbf{b}_{k+1} = \mathbf{b}_k + 2\mu e_k x_k \quad (2.18)$$

Für stationäre Signale kann die Schrittweite μ als konstant definiert werden. Jedoch kann in den meisten Anwendungen nicht von einem stationären Signal ausgegangen werden. Somit muss ein Algorithmus gefunden werden welcher zyklisch die Schrittweite anpasst. Dazu betrachtet man den Gradienten \mathbf{G}_k bei der Annäherung an \mathbf{b}_{opt} . Ausgehend von der Änderung des mittleren quadratischen Fehlers (2.9) und setzt (2.10) für $\mathbf{R}_{xx,1}$ ein erhält man

$$\mathbf{G}_k = \frac{dE(e_k^2)}{d\mathbf{b}_k} = 2\mathbf{R}_{xx,2}(\mathbf{b}_k - \mathbf{b}_{opt})$$

Um die Schrittweite μ zu erhalten setzen wir den obigen Gradienten in das Gradientenverfahren (2.15) ein.

$$\mathbf{b}_{k+1} = \mathbf{b}_k - 2\mu (\mathbf{R}_{xx,2}(\mathbf{b}_k - \mathbf{b}_{opt}))$$

Da der letzte Schritt bis zum Erreichen von \mathbf{b}_{opt} berechnet werden soll, kann $\mathbf{b}_{k+1} = \mathbf{b}_{opt}$ gesetzt werden. Das Ergebnis für die Schrittweite ist:

$$\mu = \frac{1}{2} I \mathbf{R}_{xx,2}^{-1} = \frac{1}{2} \mathbf{R}_{xx,2}^{-1}$$

Die Einheitsmatrix entfällt, da es sich bei $\mathbf{R}_{xx,2}^{-1}$ um eine $n \times n$ -Matrix handelt. Da es sich bei der Schrittweite μ um einen Skalar handelt muß aus der Korrelationsmatrix $\mathbf{R}_{xx,2}^{-1}$ ein entsprechender Wert ermittelt werden. Aus der Praxis konvergiert das System für

$$0 < \mu < \frac{1}{\sum_{i=0}^N \lambda_i}$$

Der Nenner aus obigem Bruch stellt die Summe der Eigenwerte aus der Korrelationsmatrix dar. Die Eigenwerte einer $n \times n$ -Matrix können über die Hauptdiagonalen der Matrix, also der Spur, berechnet werden.

$$\sum_{i=0}^N \lambda_i = \text{Spur}(\mathbf{R}_{xx,2}) = \sum_{i=0}^N \mathbf{R}_{xx,2,ii} = (N+1) R_{xx}(0) = (N+1) \sigma_{xx}^2$$

Aus einem mittelwertfreien und stationären Prozess resultiert daraus die Varianz σ_{xx}^2 . Somit kann eine Schrittweite μ im Bereich $0 < \mu < \mu_{max}$

$$0 < \mu < \frac{1}{(N+1)\sigma_{xx}^2} \quad (2.19)$$

für eine konvergierende optimale Koeffizientenermittlung angegeben werden. In der Praxis wird häufig eine Schrittweite von $\mu = 1/10 \mu_{max}$ verwendet.

2.6 Effiziente Varianz Berechnung

Die Varianz ist für Prädiktionssysteme eine wichtige Größe. Wie beispielsweise zur Bestimmung der Schrittweite μ im vorherigen Kapitel wird die Varianz σ_{xx}^2 des Störsignals für die Koeffizientenberechnung nach (2.18) benötigt. Das folgende Kapitel untersucht verschiedene Verfahren zur Varianzbestimmung unterteilt in mittelwertfreie und mittelwertbehaftete Signale.

Für ein mittelwertfreies Signal kann die Varianz über die Autokorrelationsfunktion R_{xx} bestimmt werden (vgl. (2.4)).

$$R_{xx,k}(i) = \frac{1}{N} \sum_{n=0}^{N-1} x_{k-n} x_{k-n-i}$$

$$R_{xx,k}(0) = \sigma_{xx}^2 = \frac{1}{N} \sum_{n=0}^{N-1} x_{k-n}^2 \quad (2.20)$$

Danach muß für jeden Störwert x_k die Korrelation für N -Werte neu berechnet werden. Um den Rechenaufwand zu minimieren bietet sich an zu Beginn einmal die Korrelation für N -Werte zu berechnen um einen realistischen Startwert zu bekommen und danach auf eine Aufwand optimierte Berechnung umzusteigen. Eine Möglichkeit ist die aktuell berechnete Varianz jeweils um einen x_k^2 -Anteil zu erweitern und dafür einen x_{k-N}^2 -Anteil zu vergessen.

$$R_{xx,k}(0) = \frac{1}{N} x_k^2 + R_{xx,k-1}(0) - \frac{1}{N} x_{k-N}^2$$

$$R_{xx,k}(0) = R_{xx,k-1}(0) + \frac{1}{N} (x_k^2 - x_{k-N}^2) \quad (2.21)$$

Die Software Umsetzung erfordert hierfür einen Ringbuffer mit N -Elementen. Bei jedem Neuen Wert x_k^2 wird der Ringbuffer aktualisiert und der Wert x_{k-N}^2 entnommen. Bei einem sehr großen N steigt hierfür auch entsprechend der Speicherbedarf. Eine weitere Optimierung des Speicherbedarfs kann erzielt werden, indem man auf den letzten Wert x_{k-N}^2 verzichtet und die verbleibenden Anteile entsprechend gewichtet.

$$R_{xx,k}(0) = R_{xx,k-1}(0) \frac{N-1}{N} + \frac{1}{N} x_k^2$$

Stellt man die Gleichung (2.22) folgendermaßen um,

$$R_{xx,k}(0) = R_{xx,k-1}(0) + \frac{1}{N} (x_k^2 - R_{xx,k-1}(0)) \quad (2.22)$$

erhält man die Form eines zeitdiskreten PT1-Glieds mit der Verstärkung $K = 1$ und der Zeitkonstante T

$$\frac{\Delta t}{T + \Delta t} = \frac{1}{N} \rightarrow T = (\Delta t N) - \Delta t$$

Um die Unterschiede der verschiedenen Optimierungen darzustellen soll ein mittelwertfreies Rauschsignal mit unterschiedlicher Amplitude erzeugt werden (s. Abbildung).

Der Varianzverlauf der unoptimierten Gleichung zeigt die zweite Darstellung von oben. Die Laufzeit optimierte Version mit dem notwendigen Ringbuffer wird durch den dritten Verlauf und die Laufzeit optimierte plus Speicher optimierte Variante durch die unterste Darstellung gezeigt. Es gibt quasi keinen Unterschied zu den ersten beiden Verläufen. Im untersten Verlauf tritt das PT1 Verhalten mit der Zeitkonstante $T = 100ms \cdot 50 - 100ms = 4.9s$ klar zum Vorschein.

Bei einem mittelwertbehafteten Signal muß der Mittelwert bei der Varianzberechnung entsprechend wie folgt abgezogen bzw. korrigiert werden:

$$\sigma_{xx}^2 = E(\mathbf{X} - \mu)^2 = E(\mathbf{X} - E(\mathbf{X}))^2$$

Multipliziert man die Gleichung folgendermaßen aus

$$\sigma_{xx}^2 = E(\mathbf{X}^2 - 2\mathbf{X}E(\mathbf{X}) + E(\mathbf{X})^2) = E(\mathbf{X}^2) - 2E(\mathbf{X})E(\mathbf{X}) + E(\mathbf{X})^2$$

erhält man einen Term bestehend aus der Autokorrelationsfunktion und dem quadrierten Mittelwert.

$$\begin{aligned} \sigma_{xx}^2 &= E(\mathbf{X}^2) - E(\mathbf{X})^2 = R_{xx,k}(0) - \mu_x^2 \\ \sigma_{xx}^2 &= R_{xx,k}(0) - \left(\frac{1}{N} \sum_{n=0}^{N-1} x_{k-n} \right)^2 \end{aligned} \quad (2.23)$$

Zur Demonstration wie unterschiedlich die Varianz ausfällt wenn der Mittelwert nicht berücksichtigt wird zeigt Abbildung mitte. Das Ausgangssignal ist das gleiche Rauschsignal wie in Abbildung oben dargestellt nur mit einem zeitlich begrenzten Offset zwischen Zeitpunkt 100 und 150 mit einer Amplitude von 500.

2.7 Störunterdrückung

In vielen Bereichen der Signalverarbeitung ist es notwendig Störsignale wie Rauschen, Netzbrummen oder anderweitige Einflüsse zu unterdrücken. Mit herkömmlicher Filterung z.B. Tiefpaßfilter wird oft das gewünschte Signal zu sehr beeinflusst. Daher macht es Sinn exakt nur das Störsignal zu entfernen ohne das Nutzsignal zu sehr zu beeinflussen. Ein Prädiktionsfilter welches das ermöglicht zeigt Abbildung 2.3.

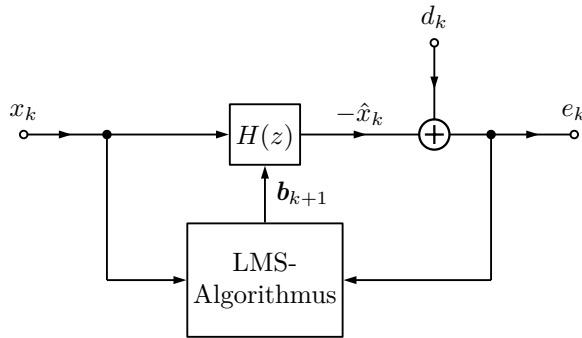


Figure 2.3: Struktur eines adaptiven Prädiktors mit dem zusätzlichen Eingang d_k

Hierfür ist ein weiterer Eingang d_k gegenüber der Struktur in Abbildung 2.2 notwendig um das gewünschte Signal vorzugeben. Eine Voraussetzung ist, dass das Störsignal separat abgreifbar und dem Filter mittels Eingang x_k zugeführt werden kann. Ein Beispiel für ein Störsignal wäre das 50Hz Netzbrummen welches sich oft dem Nutzsignal überlagert. Die Abbildung ?? zeigt als Nutzsignal ein EKG-Signal überlagert mit Netzbrummen als Störsignal.

Das Störsignal am Eingang x_k wird mittels dem Prädiktionsfilter $H_1(z)$ nach (2.1) geschätzt und die dafür benötigten Koeffizienten mit $H_2(z)$ nach (2.18) berechnet. Das resultierende Signal am Ausgang e_k ist das störbefreite Nutzsignal.

Der Algorithmus zur adaptiven Prädiktion setzt sich aus folgenden Operationen zusammen:

1. Berechnung der Varianz mittels $R_{xx,k}(0)$ mit Gleichung (2.20) in der Initialisierungsphase bzw. mit Gleichung (2.22) nach der Initialisierungsphase.
2. Bestimmung der Schrittweite μ nach (2.19) mit $\mu = 1/10 \mu_{max}$
3. Update der Koeffizienten \mathbf{b}_{k+1} mit Gleichung (2.18).
4. Schätzen von \hat{x}_k mittels der Linearkombination nach Gleichung (2.1) aus den Koeffizienten und dem Störsignal.
5. Berechnung des störbefreiten Nutzsignalwertes $e_{k+1} = d_k - \hat{x}_k$

Wendet man diesen Algorithmus auf das EKG-Signal mit dem 50Hz Netzbrummen am Eingang d_k an (Abbildung ?? mitte), so erhält man für das geschätzte Störsignal \hat{x}_k den Verlauf in Abbildung ?? oben und für das störbefreite Signal e_k Abbildung ?? unten.

ToDo: Lernkurve LMS-Algorithmus

2.8 Dekorrelation

Dekorrelation wird eingesetzt wenn die Beziehung einer Wertefolge untereinander aufgehoben werden soll. Sie ist somit das Gegenteil der Korrelation. Allgemein erreicht man mit der Dekorrelation ein konzentrieren von Information. Oder anders ausgedrückt ein konzentrieren diskreter Werte in Richtung Null-Achse. Dies kann ausgenutzt werden um eine anschließenden Datenkompression beispielsweise mittels Entropiecodierung effektiv durchzuführen.

2.8.1 Dekorrelation mittels Prädiktion

Eine Möglichkeit Dekorrelation zu realisieren ist mittels Prädiktion. Das Ziel hierbei ist die Beziehung einer Wertefolge untereinander so aufzulösen, dass dadurch ein Signal um die Null-Achse herum resultiert. Ein Prädiktionsfilter welches das ermöglicht zeigt Abbildung 2.4.

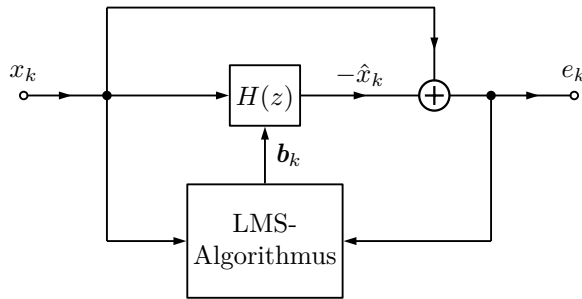


Figure 2.4: Struktur eines Dekorrelationsfilters mittels Prädiktion

Es handelt sich hierbei um einen linearen Prädiktor $H(z)$ nach Gleichung (2.1) wie in Abbildung 2.2 dargestellt plus dem adaptiven Teil $H_2(z)$ zur Bestimmung der Filterkoeffizienten. Es handelt sich hierbei um den LMS-Algorithmus mittels Gradientenverfahren realisiert wie in Kapitel 2.5 vorgestellt. Am Eingang x_k wird das zu dekorrelierende Signal zugeführt. Der Prädiktor $H(z)$ schätzt das zugeführte Signal und das Ergebnis wird vom Originalsignal x_k abgezogen. Das daraus resultierende Fehlersignal e_k entspricht dem dekorreliertem Signal. Die Koeffizienten werden mittels LMS-Algorithmus nach (2.18) berechnet. Um den Rechenaufwand zu optimieren reicht eine geringere Aktualisierungsrate f_b der Koeffizientenberechnung gegenüber der Rate $f_{\hat{x}}$ der Schätzwertberechnung aus.

$$f_b < 0.1 f_{\hat{x}} \quad (2.24)$$

Denn für kurze Zeitabschnitte kann das Signal als korreliert betrachtet werden.

Wendet man die Dekorrelation nach Struktur in Abbildung 2.4 auf ein EKG-Signal an, erhält man ein dekorreliertes Signal in Abbildung ?? unten.

Man kann erkennen, dass die Werte des dekorrelierten Signals um die Null-Achse (mittelwertfrei) angeordnet sind.

2.8.2 Quantisierte Prädiktion

Um eine Datenkompression mittels einer Entropiecodierung beispielsweise durch führen zu können, müssen die Werte des dekorrelierten Signals (siehe Abbildung ?? unten) ganzzahlig vorliegen. Dadurch ist es möglich jedem Signalwert einen eindeutigen Code zu zuweisen. Eine Möglichkeit die Dekorrelation mit ganzzahligen Signalwerten durchzuführen zeigt Abbildung 2.5.

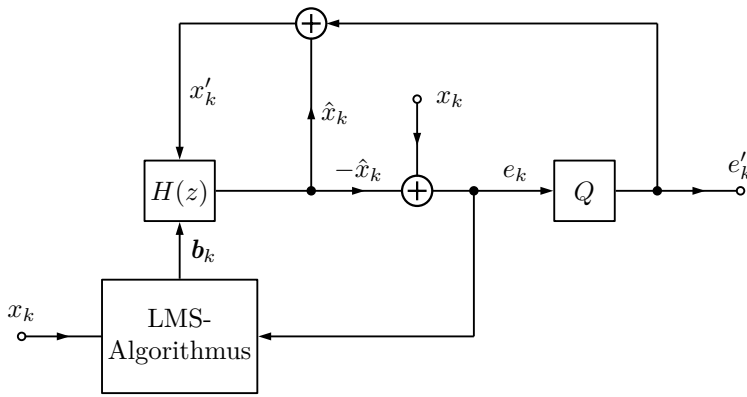


Figure 2.5: Struktur eines Dekorrelationsfilters mittels quantisierter Prädiktion

Im Prinzip baut diese Struktur auf die Struktur in Abbildung 2.4 auf. Neu hinzugekommen ist der Block Q für die Quantisierung. Dieser bekommt als Eingang das nicht quantisierte Dekorrelationssignal e_k und quantisiert es zu e'_k (siehe Abbildung ??).

Der Prädiktor $H(z)$ rechnet nun mit einem quantisierten Eingangssignal x'_k um einen entsprechenden Schätzwert \hat{x}_k zu berechnen. Der Prädiktor hat für dieses Beispiel eine Filterordnung von $N = 3$, d.h. es werden vier Koeffizienten vom LMS-Algorithmus berechnet. Der Dekorrelations-Algorithmus nach Struktur in Abbildung 2.5 setzt sich aus folgenden Operationen zusammen:

1. Berechnung der Varianz mittels $R_{xx,k}(0)$ mit Gleichung (2.20) in der Initialisierungsphase bzw. mit Gleichung (2.22) nach der Initialisierungsphase.
2. Bestimmung der Schrittweite μ nach (2.19) mit $\mu = 1/10 \mu_{max}$
3. Update der Koeffizienten \mathbf{b}_{k+1} mit Gleichung (2.18) mit der Aktualisierungsrate f_b nach Gleichung (2.24).
4. Schätzen von \hat{x}_k mittels der Linearkombination nach Gleichung (2.1) aus den Koeffizienten und dem quantisierten Eingangssignal.
5. Berechnung des dekorrelierten Signals $e_{k+1} = x_k - \hat{x}_k$.
6. Quantisierung des dekorrelierten Signals $e'_k = quant(e_k)$.
7. Berechnung des quantisierten Eingangssignals $x'_k = e_{k+1} + \hat{x}_k$.

2.8.3 Signalarückgewinnung

Um aus dem dekorrelierten Signal e'_k das ursprüngliche Signal zurück zu gewinnen stellt man den Signalfluß nach Abbildung 2.2 entsprechend um.

$$e_k = x_k - H(z)x_k$$

$$x_k = \frac{e_k}{(1 - H(z))} = \frac{e_k}{(1 - \sum_{i=0}^N b_i x_{k-i})}$$

Mit $H(z)$ berechnet aus dem Signalfluß in Abbildung (2.1) und aufgelöst nach x_k erhält man das IIR-Filter

$$x_k = b_1 x_{k-1} + b_2 x_{k-2} + \dots + e_k$$

Setzt man die Koeffizienten die man vorher mittels LMS-Algorithmus bestimmt hat ein so erhält man das ursprüngliche Signal wie in Abbildung ?? gezeigt zurück.

Da bekanntlich IIR-Filter dazu neigen instabil zu werden muss auf Stabilität geprüft werden. Dies erreicht man indem man auf die Lage der Pole im Pol-Nullstellen-Diagramm schaut. Die Pole müssen sich innerhalb des Einheitskreises befinden, dann ist das Filter stabil. Dies ist laut Abbildung ?? der Fall.

2.8.4 Prädiktionsgewinn

Eine Kenngröße der Prädiktion ist der Prädiktionsgewinn. Diese Größe gibt Auskunft wie effektiv das Fehlersignal e_k komprimiert werden kann. Der Prädiktionsgewinn ist definiert mit:

$$G_p = \frac{\sigma_{xx}^2}{\sigma_{ee}^2} = \frac{R_{xx}(0)}{R_{ee}(0)}$$

Für das dekorrelierte Signal in Abbildung ?? erhält man einen Prädiktionsgewinn von $G_p = 22.35$ (13.49 dB). Das bedeutet eine Reduktion der Amplitude des dekorrelierten

Signals e_k nach

$$10^{\frac{G_p}{20}} = 10^{\frac{13.49dB}{20}} = 4.73$$

um den Faktor 4.73. Dies kann auch veranschaulicht werden indem man die Maximum- und Minimum-Werte des Ursprünglichen Signals ($[22,-7]=29$) durch die Maximum- und Minimum-Werte des Dekorrelierten Signals ($[3,-3] = 6$) dividiert ($=4.8$). Damit können in der Datenkompression 2 Bit in der Codewortlänge eingespart werden (Faktor 4 entspricht 2 Bit).

Letztendlich führt die Quantisierung Q in Abbildung 2.5 zu diskreten Werte die mit 3 Bit (0...7) dargestellt werden können.