

15 Least-Mean-Square-Algorithmus

15.1 Einführung

Dieser Versuch setzt die Überlegungen zum adaptiven FIR-Filter fort. Die Idee der adaptiven Signalverarbeitung wird in Bild 15-1 nochmals veranschaulicht. Für die Adaption der Filterkoeffizienten werden in jedem Iterationsschritt die inverse Korrelationsmatrix und der Gradientenvektor benötigt. In den praktisch relevanten Fällen dynamischer Prozesskenngrößen müssten diese jeweils durch Kurzzeitmittelwerte geschätzt und die inverse Korrelationsmatrix berechnet werden. Ein Aufwand, der einem Einsatz in vielen Anwendungen entgegensteht.

Lernziele

Nach Bearbeiten des Versuchs können Sie

- die Grundlagen des LMS-Algorithmus vorstellen und sein Konvergenzverhalten abschätzen
- den LMS-Algorithmus in realisierbarer Form angeben und in ein MATLAB-Programm umsetzen
- anhand von Simulationen das Konvergenzverhalten und die Fehlanpassung des LMS-Algorithmus aufzeigen
- den LMS-Algorithmus vereinfachen und die Funktion der modifizierten Algorithmen durch Simulation überprüfen
- das Potenzial des LMS-Algorithmus für praktische Anwendungen bewerten

In diesem Versuch wird nach einer aufwandsgünstigen und robusten Alternative zur Adaption der Koeffizienten gesucht. Den Ausgangspunkt liefert die Idee des steilsten Abstieges des mittleren quadratischen Fehlers (MSE). Die Weiterentwicklung führt auf den Least-Mean-Square- (LMS-)Algorithmus, dem vielleicht am weitesten verbreiteten Algorithmus der adaptiven Signalverarbeitung. Seiner Bedeutung angemessen, werden im Folgenden die Grundlagen ausführlich vorgestellt. Zunächst werden die besonderen Eigenschaften der Korrelationsmatrix behandelt. Danach wird die Methode des steilsten Abstieges vorgestellt und das resultierende Konvergenzverhalten untersucht. Die Umsetzung in den LMS-Algorithmus schließt den Grundlagenteil ab.

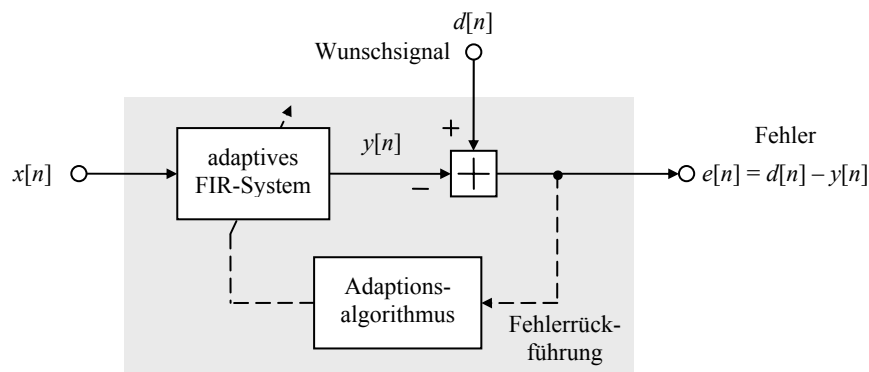


Bild 15-1 Adaptive Signalverarbeitung mit Fehlerrückführung

15.2 Grundlagen

15.2.1 Hauptachsentransformation und Inverse der Korrelationsmatrix

Den Ausgangspunkt zur Herleitung des LMS-Algorithmus liefern Überlegungen zu den Eigenwerten der Korrelationsmatrix \mathbf{R}_{xx} . Die Korrelationsmatrix besitzt Toeplitz-Form. Sie ist damit symmetrisch und nicht-singulär. Ihre Überführung in die Normalform und eine Hauptachsentransformation sind folglich möglich [BSMM99].

Anmerkung: Die Ordnung des adaptiven FIR-Systems wird hier mit N bezeichnet. Dann besitzt es $N + 1$ Koeffizienten und es wird die Korrelationsmatrix mit $N + 1$ Zeilen und Spalten benötigt. Man beachte, in der Literatur wird N auch als die Anzahl der Koeffizienten verwendet, z. B., wenn von einem Prädiktor N -ter Ordnung gesprochen wird.

Benötigt werden zunächst die Eigenvektoren der Korrelationsmatrix. Den n -ten Eigenvektor \mathbf{v}_n und seinen Eigenwert λ_n erhält man aus der Bestimmungsgleichung für Eigenvektoren, siehe Versuch 1.

$$(\mathbf{R}_{xx} - \lambda_n \mathbf{I}) \mathbf{v}_n = \mathbf{0} \quad (15.1)$$

Die Eigenvektoren sind orthogonal und können so normiert werden, dass

$$\mathbf{v}_m^T \mathbf{v}_n = \begin{cases} 1 & \text{für } m = n \\ 0 & \text{sonst} \end{cases} \quad (15.2)$$

Im Weiteren wird der Einfachheit halber angenommen, dass alle $N + 1$ Eigenwerte verschieden sind. Aus den Eigenvektoren werden nun die Spalten der *Modalmatrix* gebildet, siehe Versuch 2.

$$\mathbf{M} = (\mathbf{v}_0 \ \mathbf{v}_1 \ \dots \ \mathbf{v}_N) \quad (15.3)$$

Wegen der Orthogonalität der Eigenvektoren ist auch die Modalmatrix orthogonal.

$$\mathbf{M} \mathbf{M}^T = \mathbf{I} \quad \text{und} \quad \mathbf{M}^T = \mathbf{M}^{-1} \quad (15.4)$$

Mit der Modalmatrix kann die *Normalform* und die *Hauptachsentransformation* zur Korrelationsmatrix angegeben werden. Es gilt

$$\mathbf{R}_{xx} = \mathbf{M}^{-1} \mathbf{\Lambda} \mathbf{M} \quad \text{bzw.} \quad \mathbf{\Lambda} = \mathbf{M}^{-1} \mathbf{R}_{xx} \mathbf{M} \quad (15.5)$$

mit der *Diagonalmatrix* $\mathbf{\Lambda}$, die die Eigenwerte auf der Hauptdiagonalen enthält, und ihre Inverse $\mathbf{\Lambda}^{-1}$.

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_0 & 0 & \dots & 0 \\ 0 & \lambda_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_N \end{pmatrix} \quad \text{bzw.} \quad \mathbf{\Lambda}^{-1} = \begin{pmatrix} 1/\lambda_0 & 0 & \dots & 0 \\ 0 & 1/\lambda_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1/\lambda_N \end{pmatrix} \quad (15.6)$$

Die *inverse Korrelationsmatrix* ergibt sich aus Inversion der Normalform (15.5) mit der Orthogonalität der Modalmatrix (15.4). Anwenden der Rechenregel für das Inverse eines Produkts aus zwei Matrizen liefert in zwei Schritten

$$\mathbf{R}_{xx}^{-1} = (\mathbf{M}^T \mathbf{\Lambda} \mathbf{M})^{-1} = (\mathbf{M}^T)^{-1} (\mathbf{\Lambda} \mathbf{M})^{-1} = \mathbf{M} \mathbf{\Lambda}^{-1} \mathbf{M}^T \quad (15.7)$$

Eine explizite Matrixinversion ist somit entfallen. An ihre Stelle tritt die Bestimmung der Eigenwerte. Im endgültigen LMS-Algorithmus wird auch das vermieden. Zur Herleitung der Konvergenzeigenschaften des LMS-Algorithmus wird jedoch von den Eigenwerten Gebrauch gemacht.

15.2.2 Methode des steilsten Abstiegs

Der zweite Schritt zur Vereinfachung des Adaptionalgorithmus befasst sich mit einer alternativen Anpassung des Koeffizientenvektors. Dazu wird von der zentrierten Form des Koeffizientenvektors ausgegangen und die Rekursionsformel der Adaption (14.9) vereinfacht, indem auf die Gewichtung mit der inversen Korrelationsmatrix verzichtet wird.

$$\mathbf{c}[n+1] = \mathbf{c}[n] - \mu \cdot \mathbf{g}[n] \quad (15.8)$$

Während nach (14.11) die Momentanwerte des Koeffizientenvektors vom Startvektor auf einer Geraden in die MMSE-optimale Lösung laufen, ändern sich die Momentanwerte des Koeffizientenvektors nun jeweils in Richtung des Momentanwertes des Gradienten des MSE. Man spricht deshalb von der *Methode des steilsten Abstiegs*.

Für den Gradienten bezüglich der zentrierten Form $\mathbf{g}[n]$ ergibt sich aus (14.6) mit (14.3) der Zusammenhang

$$\mathbf{g}[n] = 2\mathbf{R}_{xx}\mathbf{b}[n] - 2\mathbf{r}_{xy} = 2\mathbf{R}_{xx}(\mathbf{c}[n] + \mathbf{b}_{\text{opt}}) - 2\mathbf{R}_{xx}\mathbf{b}_{\text{opt}} = 2\mathbf{R}_{xx}\mathbf{c}[n] \quad (15.9)$$

Für die Rekursion der zentrierten Koeffizienten resultiert

$$\mathbf{c}[n+1] = \mathbf{c}[n] - 2\mu \cdot \mathbf{R}_{xx} \mathbf{c}[n] = (\mathbf{I} - 2\mu \cdot \mathbf{R}_{xx})\mathbf{c}[n] \quad (15.10)$$

Das Gleichungssystem kann durch Hauptachsentransformation der Korrelationsmatrix vereinfacht werden. Dazu wird die Rekursionsformel von links mit der transponierten Modalmatrix multipliziert.

$$\mathbf{M}^T \mathbf{c}[n+1] = \mathbf{M}^T (\mathbf{I} - 2\mu \cdot \mathbf{R}_{xx})\mathbf{c}[n] = (\mathbf{M}^T - 2\mu \cdot \mathbf{M}^T \mathbf{R}_{xx})\mathbf{c}[n] \quad (15.11)$$

Nun wird die Koordinatentransformation

$$\tilde{\mathbf{c}}[n] = \mathbf{M}^T \mathbf{c}[n] \quad \text{und} \quad \mathbf{c}[n] = \mathbf{M} \tilde{\mathbf{c}}[n] \quad (15.12)$$

angewendet. Es ergibt sich mit

$$\tilde{\mathbf{c}}[n+1] = (\mathbf{M}^T - 2\mu \cdot \mathbf{M}^T \mathbf{R}_{xx})\mathbf{M} \tilde{\mathbf{c}}[n] = (\mathbf{I} - 2\mu \cdot \mathbf{\Lambda}) \tilde{\mathbf{c}}[n] \quad (15.13)$$

ein System von $N + 1$ unabhängigen Differenzgleichungen 1. Ordnung. Die Diagonalmatrix $\mathbf{\Lambda}$ beschreibt mit den Eigenwerten λ_k die linearen Abhängigkeiten im Eingangsprozess. Für die k -te Zeile des Gleichungssystems erhält man die Differenzgleichung 1. Ordnung

$$\tilde{c}_k[n+1] = 1 - 2\mu \cdot \lambda_k \cdot \tilde{c}_k[n] \quad \text{für} \quad k = 0, 1, \dots, N \quad (15.14)$$

Wie in Versuch 14 kann nun das Konvergenzverhalten der Methode des steilsten Abstieges untersucht werden. Die Differenzgleichungen konvergieren für

$$0 < \mu < \frac{1}{\max_{k=0, \dots, N} \lambda_k} \quad (15.15)$$

Der größte Eigenwert bestimmt den zulässigen Maximalwert des Konvergenzparameters.

Die Konvergenzgeschwindigkeit bezüglich der Koeffizienten $\tilde{\mathbf{c}}$ und des momentanen MSE kann wie in Versuch 14 abgeschätzt werden. Mit (14.14) und (14.20) gilt für die Zeitkonstanten der Methode des steilsten Abstieges

$$\tau_{b,k} = \frac{1}{2\mu \cdot \lambda_k} \quad \text{und} \quad \tau_{MSE,k} = \frac{1}{4\mu \cdot \lambda_k} \quad \text{für } k = 0, 1, \dots, N \quad (15.16)$$

Die Konvergenzgeschwindigkeit wird durch den kleinsten Eigenwert bestimmt.

15.2.3 LMS-Algorithmus

Um die Methode des steilsten Abstieges zuverlässig anwenden zu können, ist zur Einstellung des Konvergenzparameters (15.15) die Kenntnis des jeweils größten Eigenwertes erforderlich. Diese Information ist in den Anwendungen meist nicht verfügbar, sodass ein alternativer Weg gefunden werden muss. Die Überlegungen führen schließlich zum LMS-Algorithmus.

Zunächst wird an die Rechenoperation Spur einer Matrix, englisch trace, erinnert. Für die Diagonalmatrix und die Korrelationsmatrix gelten

$$\text{Tr}(\mathbf{\Lambda}) = \prod_{k=0}^N \lambda_k \quad \text{bzw.} \quad \text{Tr}(\mathbf{R}_{xx}) = \prod_{k=0}^N R_{xx}(0) = (N+1) \cdot \sigma_x^2 \quad (15.17)$$

Darin ist σ_x^2 die Varianz des (Eingangs-)Prozesses $x[n]$, eine statistische Größe, die als Kurzzeitmittelwert geschätzt werden kann.

Ersetzt man die Diagonalmatrix in (15.17) durch die Hauptachsentransformation (15.5) und beachtet die Rechenregel für die Spur eines Produkts zweier Matrizen, so resultiert schließlich für das Produkt der Eigenwerte der Zusammenhang mit der Varianz

$$\text{Tr}(\mathbf{\Lambda}) = \text{Tr}(\mathbf{R}_{xx}) = (N+1) \cdot \sigma_x^2 \quad (15.18)$$

Damit ist es möglich, den maximalen Eigenwert für die Wahl des Konvergenzparameters zu schätzen. Die konservative Abschätzung

$$\mu = \frac{\alpha}{(N+1)\sigma_x^2} \quad \text{für } 0 < \alpha < 1 \quad (15.19)$$

liefert eine konvergente Version der Methode des steilsten Abstieges

$$\mathbf{b}[n+1] = \mathbf{b}[n] - \frac{\alpha}{(N+1)\sigma_x^2} \cdot \mathbf{g}[n] \quad (15.20)$$

Die Methode des steilsten Abstieges setzt in jedem Iterationsschritt die Kenntnis des momentanen Gradienten sowie der Varianz des Eingangssignals des adaptiven FIR-Systems voraus. Für einen einfachen und robusten Algorithmus müssen der Gradientenvektor und die Varianz durch relativ einfache, aus dem Signal in Bild 15-1 bestimmbare Größen ersetzt werden.

Für den Gradienten kann die partiellen Ableitungen des MSE in (10.23) benutzt werden. Bei der Herleitung der Orthogonalitätsbeziehung ergab sich dort

$$\frac{\partial}{\partial b_k} MSE = \frac{\partial}{\partial b_k} E(e^2[n]) = 2E\left(e[n] \cdot \frac{\partial}{\partial b_k} e[n]\right) = -2E(e[n] \cdot x[n-k]) \quad (15.21)$$

für $k = 0, 1, \dots, N$. In Matrix-Vektorform resultiert daraus mit dem Vektor des Eingangssignals

$$\mathbf{x}[n] = (x[n], x[n-1], \dots, x[n-N])^T \quad (15.22)$$

der gesuchte Gradient als Erwartungswert

$$\mathbf{g}[n] = -2E(e[n] \cdot \mathbf{x}[n]) \quad (15.23)$$

Da der Erwartungswert unbekannt ist, wird der Kurzzeitmittelwert eingesetzt. Es hat sich als günstig erwiesen, die einfachste Form der Näherung, den Momentanwert, zu verwenden.

Mit dem einfachen Schätzwert für den Gradienten und einem geeigneten Schätzwerten der momentanen Varianz resultiert der *LMS-Algorithmus* zur Adaption der Koeffizienten

$$\mathbf{b}[n+1] = \mathbf{b}[n] + \frac{2\alpha}{(N+1) \cdot \hat{\sigma}_x^2[n]} \cdot e[n] \cdot \mathbf{x}[n] \quad (15.24)$$

15.2.4 Varianten des LMS-Algorithmus

Eine übliche Methode zur Schätzung der momentanen Varianz ist der Kurzzeitmittelwert über den aktuellen Eingangsvektor in Verbindung mit einer Mittelung über einen längeren Zeitraum mit exponentiellem Vergessen durch eine Differenzgleichung 1. Ordnung.

$$\tilde{\sigma}_x^2[n+1] = (1-\beta) \cdot \tilde{\sigma}_x^2[n] + \beta \cdot \mathbf{x}^T[n] \mathbf{x}[n] \quad \text{für } 0 < \beta < 1 \quad (15.25)$$

Der Abklingfaktor darin ist $(1-\beta)$.

Bei der Schätzung der Varianz durch das Skalarprodukt in (15.25) wird die Normierung durch den Stichprobenumfang weggelassen und so der Faktor $(N+1)$ in (15.24) bereits berücksichtigt. Damit ergibt sich eine praktisch einsetzbare Variante des LMS-Algorithmus.

$$\mathbf{b}[n+1] = \mathbf{b}[n] + \frac{\alpha}{\tilde{\sigma}_x^2[n]} \cdot e[n] \cdot \mathbf{x}[n] \quad \text{für } 0 < \alpha < 2 \quad (15.26)$$

Die Umsetzung des Algorithmus veranschaulicht das Blockdiagramm in Bild 15-2. Der Algorithmus wird für ein adaptives FIR-System 2. Ordnung, d. h. $N = 2$, dargestellt. Auf den zweiten Blick erkennt man die einfache Struktur. Sie zeigt sich in der Regelmäßigkeit und den elementaren Rechenoperationen Addition, Multiplikation und Verzögerung; allein die Division fällt etwas aus dem Rahmen.

Anmerkung: Da die Adaption auf Schätzwerten beruht, kann die Division vereinfacht werden, z. B. durch Näherung des Divisors als Zweierpotenz-Zahl und entsprechender Schiebeoperation. Um eventuelle numerische Probleme bei sehr kleinen Schätzwerten $\hat{\sigma}_x^2[n]$ im Nenner zu vermeiden, kann eine kleine Konstante hinzugefügt werden.

Die drei Schritte des LMS-Algorithmus fasst die folgende Aufstellung zusammen.

LMS-Algorithmus

① Schätzung der lokalen Varianz des Eingangssignals

$$\hat{\sigma}_x^2[n+1] = (1-\beta) \cdot \hat{\sigma}_x^2[n] + \beta \cdot \mathbf{x}^T[n] \mathbf{x}[n] \quad \text{für } 0 < \beta < 1 \quad (15.27)$$

② Berechnung des momentanen Fehlers

$$e[n] = d[n] - \mathbf{b}^T[n] \mathbf{x}[n] \quad (15.28)$$

③ Anpassen der Koeffizienten

$$\mathbf{b}[n+1] = \mathbf{b}[n] + \frac{\alpha}{\hat{\sigma}_x^2[n]} \cdot e[n] \cdot \mathbf{x}[n] \quad \text{für } 0 < \alpha < 2 \quad (15.29)$$

Da sich der LMS-Algorithmus in der Praxis als robust erweist, bietet es sich an, seine Komplexität durch weitere Vereinfachungen zu reduzieren.

$$\mathbf{b}[n+1] = \mathbf{b}[n] + \Delta[n] \quad (15.30)$$

Dabei wird der Aufwand durch Näherung von Signalwerten mit der Signum-Funktion, d. h. durch +1 bzw. -1, auf Kosten der Konvergenz deutlich reduziert. Die Signum-Funktion kann für den Fehler, das Signal oder beides eingesetzt werden.

$$\begin{aligned} \Delta_1[n] &= \alpha \cdot \text{sgn}(e[n]) \cdot \mathbf{x}[n] \\ \Delta_2[n] &= \alpha \cdot e[n] \cdot \text{sgn}(\mathbf{x}[n]) \\ \Delta_3[n] &= \alpha \cdot \text{sgn}(e[n]) \cdot \text{sgn}(\mathbf{x}[n]) \end{aligned} \quad (15.31)$$

Der Parameter α wird als kleine Zweierpotenz-Zahl gewählt. Auf die Anpassung der Varianz wird ganz verzichtet.

15.2.5 Fehlanpassung

Der LMS-Algorithmus beruht auf der Anpassung des Koeffizientenvektors $\mathbf{b}[n]$ in jedem Takt. Ist der MMSE-optimale Koeffizientenvektor \mathbf{b}_{opt} eingestellt, führen weitere Anpassungen in der Regel zu einer Fehleinstellung. Um den Einfluss der fehlerhaften Anpassung quantitativ zu erfassen, wird die *Fehlanpassung*, englisch *misadjustment*, im Sinne eines Scharmittelwertes als relative Zunahme des mittleren quadratischen Fehlers definiert.

$$M = \frac{MSE - MSE_{\text{opt}}}{MSE_{\text{opt}}} \quad (15.32)$$

Theoretische Überlegungen und Experimente zeigen, z. B. in [WiSt85], dass die Fehlanpassung zunimmt, wenn die Zahl der Prädiktionskoeffizienten zunimmt und/oder der Konvergenz-

parameter abnimmt. Letzteres heißt für den praktischen Einsatz, dass ein Kompromiss zwischen Konvergenzgeschwindigkeit und Fehlanpassung gewählt werden muss.

Anmerkung: Auch die Eigenwerte der Korrelationsmatrix haben einen Einfluss auf die Fehlanpassung.

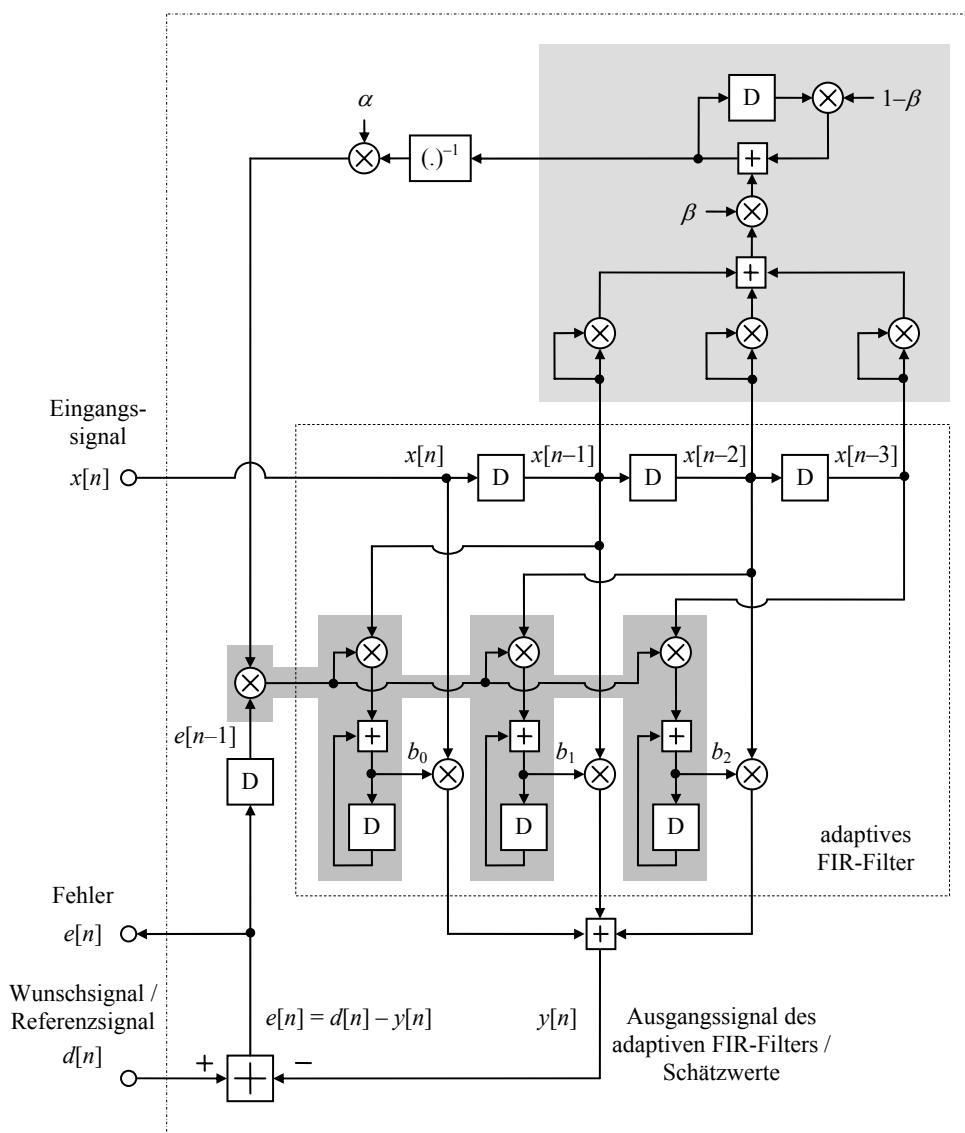


Bild 15-2 Blockdiagramm eines adaptiven FIR-Systems 2. Ordnung mit vereinfachtem LMS-Algorithmus

15.3 Versuchsdurchführung

Im Versuch sollen Eigenschaften und die Wirksamkeit des LMS-Algorithmus durch eine Simulation untersucht werden.

Für die Simulation gehen Sie wie in Bild 15-3 vor. Verwenden Sie als Modellsystem das IIR-System aus Versuch 10 mit der Übertragungsfunktion

$$H(z) = 0.24192 \cdot \frac{z^{-2} + 2z^{-1} + 1}{z^{-2} - 0.71z^{-1} + 0.25}$$

Dort wurde auch der MSE-optimale Prädiktor 2. Ordnung bestimmt.

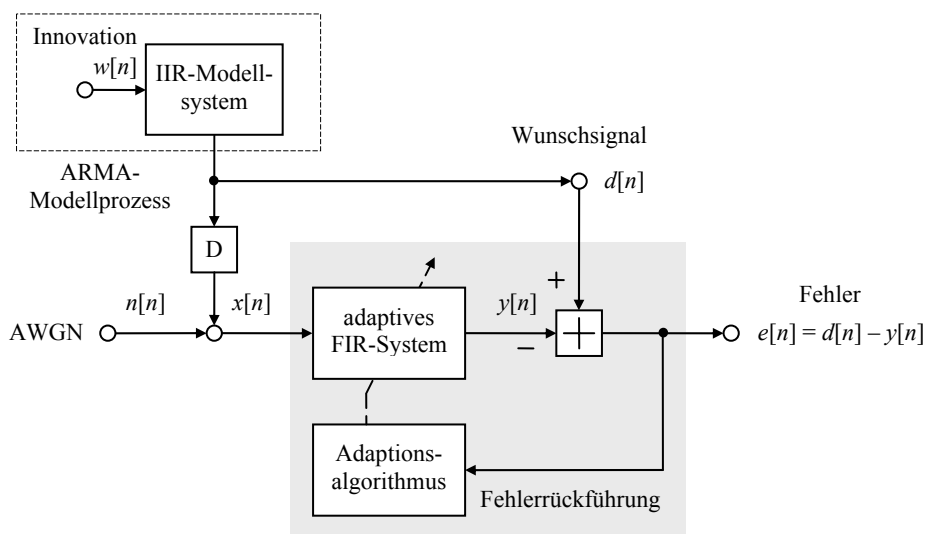


Bild 15-3 Simulation der adaptiven Prädiktion mit Fehlerrückführung und Rauschstörung

M15.1 Schreiben Sie ein MATLAB-Programm zur Bestimmung der MSE-optimalen Prädiktionkoeffizienten und des resultierenden MSE. Ergänzen Sie das Programm um die Berechnung und Ausgabe der Eigenwerte der Korrelationsmatrix.

Hinweis: Verwenden Sie soweit möglich Teile aus Programmen zu früheren Versuchen.

M15.2 Erstellen Sie eine MATLAB-Programm für den LMS-Algorithmus nach (15.26) und Bild 15-2. Dabei soll der LMS-Algorithmus als MATLAB-Funktion für ein adaptive Prädiktion der Ordnung p (FIR-System der Ordnung $N = p - 1$) umgesetzt werden. Gestalten Sie die Funktion so, dass ein wiederholter Funktionsaufruf für eine fortlaufende Simulation ohne Blockrandeffekte möglich wird.

Wählen Sie die Parameter α und β geeignet und führen Sie eine Simulation für ein adaptives FIR-System mit zwei Koeffizienten durch.

Stellen Sie das Wunschsinal $d[n]$ und den Fehler $e[n]$ bildlich dar.

Tragen Sie auch den Verlauf der Schätzwerte der Varianz grafisch auf. Welchen Wert hat die Varianz von $x[n]$? Welchen Wert sollte der Schätzwert nach (15.25) annehmen?

Hinweis: Definieren Sie alle den LMS-Algorithmus betreffenden Größen in einer Struktur, siehe Programmbeispiel 15-1. Für den Rücksprung sehen Sie eine Variable als Container vor, die alle für die weiteren Analysen und grafischen Ausgaben interessierenden Zwischenwerte aufnehmen kann.

Programmbeispiel 15-1 Funktionsaufruf des LMS-Algorithmus (Programmausschnitt)

```
% .alpha - convergence parameter,
% .beta  - weighting parameter of variance estimator
% .b     - FIR filter coefficients (N+1 initial/final values)
% .x     - input signal (N+1 initial/final values)
% .e     - error (initial/final value)
% .sig2  - variance (initial/final value)
% .Mode  - mode for updating FIR filter coefficients,
%          Mode = 'e*x' no sign function used
%          Mode = 'x*sign(e)', 'e*sign(x)' and 'sign(e*x)' uses
%          sign function
lms = struct('alpha',1/16,'beta',1/32,'b',zeros(p,1),'e',0,'x',...
           zeros(p,1),'sig2',p,'Mode','e*x');
[y,e,lms,SAVEb] = lms_algorithm(x,d,lms);           % LMS algorithm
```

M15.3 Ergänzen Sie das Simulationsprogramm so, dass zu den temporären Prädiktionskoeffizienten auch die Lernkurve grafisch dargestellt wird.

Hinweis: Bild 15-4 zeigt an einem Simulationsbeispiel, wie Ihr Ergebnis aussehen könnte.

M15.4 Erweitern Sie das Simulationsprogramm um die zweidimensionale Darstellung des MSE in Abhängigkeit von den Koeffizienten (Höhenliniendarstellung in der (c_0, c_1) -Ebene) und tragen Sie die Entwicklung der Koeffizienten ein. Wodurch unterscheidet sich das Konvergenzverhalten grundlegenden von dem in Versuch 14, Bild 14-4? Worauf ist dies zurückzuführen?

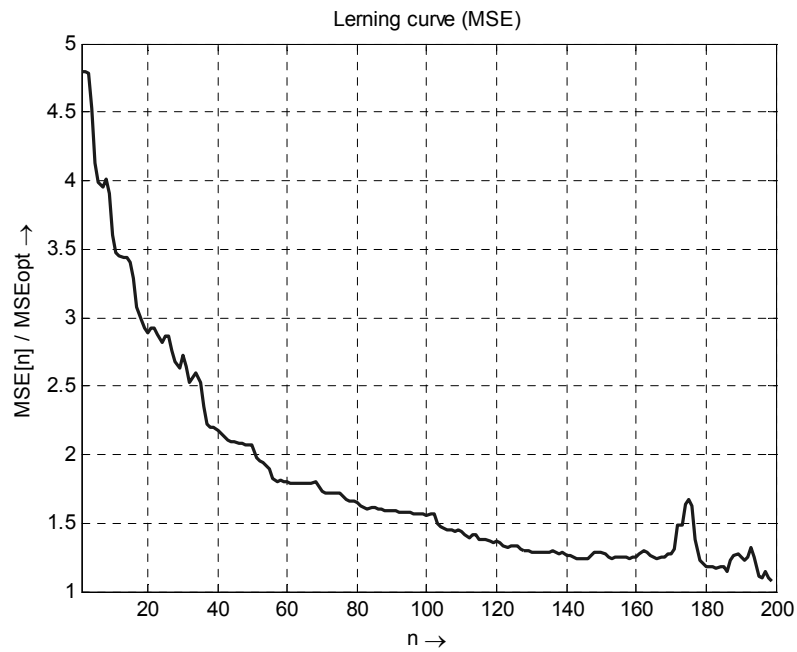
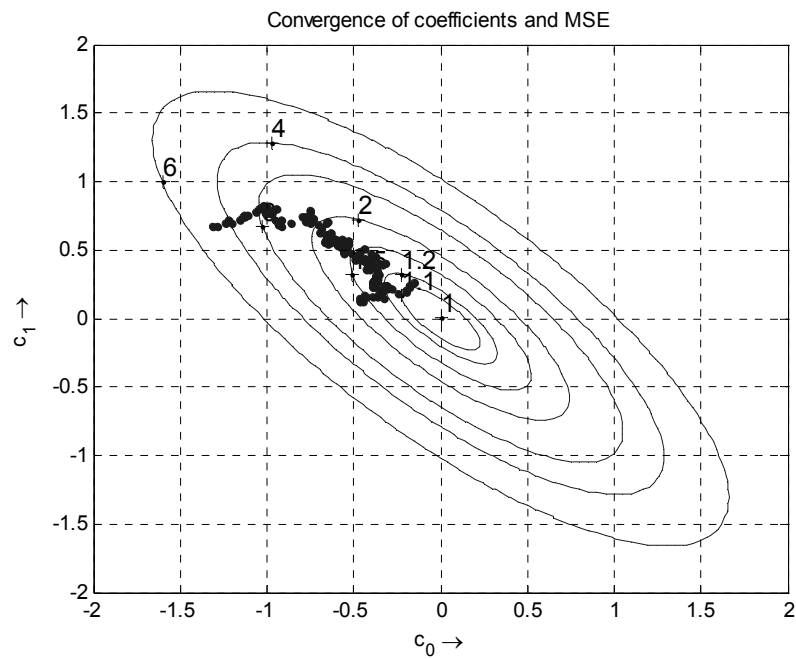
Hinweis: Bild 15-5 zeigt an einem Simulationsbeispiel, wie Ihr Ergebnis aussehen könnte.

Nachdem Sie nun ein MATLAB-Programm zur Simulation des LMS-Algorithmus und zur grafischen Ausgabe interessanter Größen erstellt haben, können Sie sich ein Bild über die Eigenschaften des LMS-Algorithmus machen.

M15.5 Untersuchen Sie durch Simulation den Einfluss der Parameter α und β und des Startwerts für den Schätzwert der Varianz auf die Lernkurve und die Entwicklung der Koeffizienten. Diskutieren Sie die Ergebnisse.

M15.6 Untersuchen Sie durch Simulation das Verhalten des Algorithmus bei nahezu optimaler Anpassung. Geben Sie dazu die FIR-Filterkoeffizienten der MSE-optimalen Lösung und die Varianz des Eingangsprozesses als Startwerte vor. Beobachten Sie die weitere die Entwicklung der Koeffizienten in der (c_0, c_1) -Ebene und bestimmen Sie die Fehlanpassung quantitativ durch Schätzung des mittleren quadratischen Fehlers. Diskutieren Sie die Ergebnisse.

M15.7 Untersuchen Sie, ob die drei vorgeschlagenen Vereinfachungen in (15.31) prinzipiell geeignet sind. Diskutieren Sie die Ergebnisse.

**Bild 15-4** Simulation des LMS-Algorithmus – Lernkurve**Bild 15-5** Simulation des LMS-Algorithmus – Konvergenz der Prädiktionkoeffizienten und MSE

Sie können nach eigenem Interesse auch weitergehende Untersuchungen vornehmen. Folgende Fragen sind von allgemeinem Interesse: Wie verhält sich der LMS-Algorithmus

- bei mit Rauschen (AWGN) gestörtem Eingangssignal, siehe Bild 15-3? Sind die vereinfachten Verfahren robuster gegen eine Rauschstörung?
- bei Prädiktion höherer Ordnung?
- bei unterschiedlichen Modellsystemen (höhere Ordnung, Eigenwerte) ?
- bei einem Audiosignal?

Hinweise zu MATLAB

Siehe auch frühere Versuche.

MATLAB functions | `contour, eig, inv, sign, \`



Online-Ressourcen zum Versuch

`lab15_1.m`, `lab15_2.m`, `lab15_3.m`, `lms_algorithm.m`, `handel.wav`, `speech.wav` und Programme mit Ergebnissen im HTML-Format