1 Projekt Adaptive Filter und Aktive Schallfeldkontrolle

Praktikanten: Stefan Kamphausen und Volker Ahlers
Betreuer: Fabian Evert und Holger Opfer

1.1 Einleitung

Ziel des Projektes ist es, auf einen Holzkanal mit quadratischem Querschnitt gegebenes Rauschen durch Einstrahlen eines amplitudenähnlichen, aber in der Phase um 180° verschobenen Signals zu minimieren. Der Kanal könnte in einer tatsächlichen Anwendung beispielsweise ein Lüftungsschacht sein, in dem das Rauschen einer Klimaanlage beseitigt werden soll. Das einzustrahlende Signal soll dabei mit Hilfe eines adaptiven Filters bestimmt werden, dessen Koeffizienten unter Benutzung des LMS (*least mean square*)-Algorithmus optimiert werden.

Abschnitt 1.2 bietet eine kurze Einführung in die Theorie adaptiver Filter, in der unter anderem der LMS-Algorithmus beschrieben wird. In Abschnitt 1.3 werden dann, neben einigen zusätzlichen praktischen Überlegungen, die durchgeführten Experimente beschrieben.

1.2 Theorie adaptiver Filter

Wie bereits im Theorieteil dieses Protokollbandes angedeutet, werden adaptive Filter dazu verwendet, optimale Filterkoeffizienten eines Filters zu finden, das zu einem gegebenen Eingangssignal x_k ein Ausgangssignal y_k liefert, so daß der mittlere quadratische Fehler (*mean square error*, MSE) zwischen diesem Ausgangssignal und einem gewünschten (*desired*) Signal d_k minimal wird,

$$\langle e_k^2 \rangle = \langle (d_k - y_k)^2 \rangle \stackrel{!}{=} \min. \tag{1.1}$$

Dabei kann sich die gewünschte Filterfunktion, also die Verknüpfung zwischen dem Eingangssignal x_k und dem gewünschten Signal d_k, zeitlich ändern, so daß das Filter ständig nachadaptiert werden muß. Die optimalen Filterkoeffizienten können also im allgemeinen

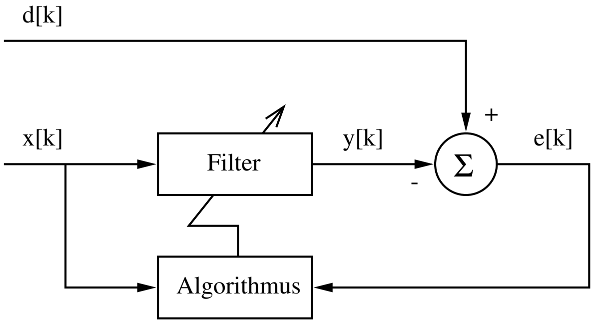

Abb. 1.1: Blockdiagramm eines adaptiven Filters; der Algorithmus soll die Koeffizienten des Filters so anpassen, daß $\langle e_k^2 \rangle$ minimal wird.

nicht vor der Konstruktion des Filters analytisch berechnet werden, wie es beim Wiener-Filter der Fall ist. Abbildung 1.1 zeigt den Aufbau des Filters als Blockdiagramm.[1]

1.2.1 Analytische Berechnung optimaler Filterkoeffizienten

Können die Filterkoeffizienten auch nicht vor Konstruktion des Filters berechnet werden, so wäre es doch denkbar, diese Rechnung in nicht zu großen Zeitabständen bei Betrieb des Filters wiederholt durchzuführen, um es so an die sich ändernden Gegebenheiten anzupassen. x_k und d_k werden dabei als schwach stationäre stochastische Prozesse betrachtet, während sich die Filterkoeffizienten $\vec{w}_k$ nur langsam im Vergleich zu x_k und d_k ändern sollen.

Im folgenden werden ausschließlich FIR (*finite impulse response*)-Filter betrachtet, die sich mathematisch durch

$$y_k = \sum_{i=0}^{l-1} w_k^{(i)} x_{k-i} = \vec{w}_k^\top \vec{x}_k \tag{1.2}$$

darstellen lassen mit der Filterlänge l, dem Filterkoeffizientenvektor

$$\vec{w}_k = (w_k^{(0)}, w_k^{(1)}, \ldots, w_k^{(l-1)})^\top \tag{1.3}$$

und dem Eingangsdatenvektor

$$\vec{x}_k = (x_k, x_{k-1}, \ldots, x_{k-l+1})^\top. \tag{1.4}$$

Setzen wir dies in (1.1) ein und multiplizieren aus, so erhalten wir

$$\begin{aligned} \langle e_k^2 \rangle &= \langle \vec{w}_k^\top \vec{x}_k \vec{x}_k^\top \vec{w}_k \rangle - 2\langle d_k \vec{w}_k^\top \vec{x}_k \rangle + \langle d_k^2 \rangle & (1.5) \\ &= \vec{w}_k^\top \mathbf{R} \vec{w}_k - 2\vec{w}_k^\top \vec{p} + \sigma_d^2, & (1.6) \end{aligned}$$

[1] Aus Gründen der Übersichtlichkeit haben wir uns dazu entschieden, den Abtastindex k nicht wie im Theorieteil in eckige Klammern (à la $x[k]$), sondern als Index zu setzen.

wobei die Mittelung des Koeffizientenvektors $\vec{w}_k$ entfällt, da sich dieser nur langsam ändert. Die Matrix $\mathbf{R}$ beschreibt die Autokorrelation der x_k untereinander,

$$\mathbf{R} = \langle \vec{x}_k \vec{x}_k^\top \rangle = \begin{pmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(l-1) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(l-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(l-1) & r_{xx}(l-2) & \cdots & r_{xx}(0) \end{pmatrix} \tag{1.7}$$

mit der Autokorrelationsfunktion

$$r_{xx}(j) = \langle x_k x_{k-j} \rangle, \tag{1.8}$$

die wegen der angenommenen Stationarität nicht von k abhängt. Der Kreuzkorrelationsvektor $\vec{p}$ zwischen d_k und x_k,

$$\vec{p} = \langle d_k \vec{x}_k \rangle = \Big(r_{dx}(0), r_{dx}(1), \ldots, r_{dx}(l-1) \Big)^\top \tag{1.9}$$

mit der Kreuzkorrelationsfunktion

$$r_{dx}(j) = \langle d_k x_{k-j} \rangle, \tag{1.10}$$

hängt aus dem gleichen Grund ebenfalls nicht von k ab. Die Größe $\sigma_d^2 = \langle d_k^2 \rangle$ beschreibt die mittlere Leistung von d_k.

Da (1.6) eine positiv semidefinite quadratische Form darstellt, die ein eindeutiges Minimum besitzt, läßt sich der optimale Filterkoeffizientenvektor $\vec{w}_{\mathrm{opt}}$ durch Ableiten von (1.6) nach $\vec{w}_k$ finden,

$$\nabla_{\vec{w}} \langle e_k^2 \rangle = 2\mathbf{R}\vec{w}_k - 2\vec{p} \stackrel{!}{=} 0. \tag{1.11}$$

Dies führt auf ein lineares Gleichungssystem

$$\mathbf{R}\vec{w}_{\mathrm{opt}} = \vec{p}, \tag{1.12}$$

das die diskrete Form der Wiener-Hopf-Gleichungen darstellt. Da sich $\mathbf{R}$ und $\vec{p}$ im allgemeinen ändern können (sonst hätten wir es ja mit einem gewöhnlichen Wiener-Filter zu tun), müßte (1.12) immer wieder neu für $\vec{w}_{\mathrm{opt}}$ gelöst werden. Dies ist jedoch trotz der symmetrischen Toeplitz-Struktur der Matrix $\mathbf{R}$ für größere Filterlängen sehr rechenaufwendig (nämlich $\mathcal{O}(l^2)$ Punktoperationen), weshalb diese Vorgehensweise für adaptive Filter verworfen werden muß.

1.2.2 Gradientenverfahren

Gesucht wird also ein weniger rechenaufwendiges Verfahren, um den mittleren quadratischen Fehler $\langle e_k^2 \rangle$ zu minimieren. Es wurde bereits bemerkt, daß die positiv semidefinite quadratische Form (1.6) für den Fehler $\langle e_k^2 \rangle$ in Abhängigkeit von den Filterkoeffizienten $\vec{w}_k$ ein eindeutiges Minimum besitzt. Iterative Gradientenverfahren stellen also gute

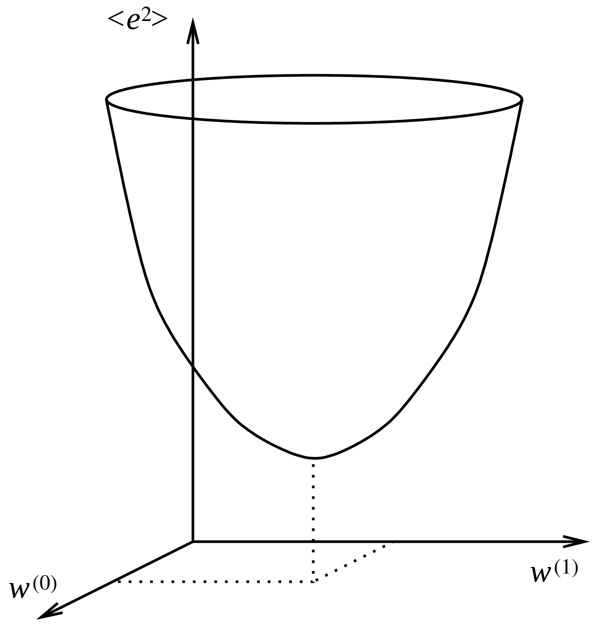

Abb. 1.2: Für $l = 2$ wird die Fehlerhyperfläche durch einen Paraboloiden gebildet.

Kandidaten für unseren adaptiven Filteralgorithmus dar. Im folgenden wird die besonders einfache und wenig rechenintensive Methode des steilsten Abstiegs (*method of steepest descent*) behandelt, ohne daß auf mögliche Alternativen eingegangen wird.

Gleichung (1.6) beschreibt eine l-dimensionale Hyperfläche im $(l + 1)$-dimensionalen Raum, dessen Punkte Koordinaten

$$\left(w_k^{(0)}, w_k^{(1)}, \ldots, w_k^{(l-1)}, \langle e_k^2 \rangle\right)$$

besitzen; für $l = 2$ ergibt sich ein Paraboloid, wie in Abb. 1.2 dargestellt ist. Aus (1.12) erhalten wir

$$\vec{w}_{\text{opt}} = \mathbf{R}^{-1}\vec{p}, \tag{1.13}$$

was in (1.6) eingesetzt einen minimalen mittleren quadratischen Fehler

$$\langle e_k^2 \rangle_{\min} = \sigma_d^2 - \vec{w}_{\text{opt}}^{\top}\vec{p} \tag{1.14}$$

ergibt. Diesem nähert sich das System an, wenn die Filterkoeffizienten in Richtung des steilsten Abstiegs der Fehlerhyperfläche verändert werden; diese Richtung ist durch den negativen Gradienten $-\nabla_{\vec{w}}\langle e_k^2 \rangle$ gegeben.

In jedem Abtastschritt wird der Filterkoeffizientenvektor also um ein gewisses Stück in Richtung des negativen Gradienten verändert,

$$\vec{w}_{k+1} = \vec{w}_k - \frac{\mu}{2}\nabla_{\vec{w}}\langle e_k^2 \rangle, \tag{1.15}$$

wobei μ die Schrittweite beschreibt, die geeignet gewählt werden muß: Ein zu kleiner Wert läßt das System nur sehr langsam adaptieren, während ein zu großer Wert dazu führen kann, daß „über das Ziel hinausgeschossen“ wird. Auf die Frage, welche Werte geeignet sind, wird in Abschnitt 1.2.4 noch genauer eingegangen.

Mit dem Ausdruck (1.11) für den Gradienten ergibt sich

$$\vec{w}_{k+1} = \vec{w}_k - \mu(\mathbf{R}\vec{w}_k - \vec{p}). \tag{1.16}$$

Theoretisch müssen also nur einmal die Autokorrelationsfunktion der x_k untereinander sowie die Kreuzkorrelationsfunktion der d_k und x_k bestimmt werden, um in jedem Abtastschritt auf einfache Weise den Gradienten berechnen zu können. Erstens sind diese Größen in der Praxis aber oft nicht ohne weiteres zu bestimmen, und zweitens ist die zugrundeliegende Annahme der Stationarität ein theoretisches Konstrukt, das die analytische Behandlung vieler Probleme zwar erleichtert oder erst ermöglicht, in den meisten realen Anwendungen aber nicht erfüllt ist. Die Größen $\mathbf{R}$ und $\vec{p}$ hängen in letzterem Fall von k ab, so daß ein anderer Weg gefunden werden muß, um zumindest eine Abschätzung für den Gradienten zu bekommen.

1.2.3 Der LMS-Algorithmus

Durch Einsetzen der Ausdrücke für $\mathbf{R}$ und $\vec{p}$ aus (1.7) bzw. (1.9) in (1.16) und Vereinfachen erhalten wir

$$\begin{aligned}
\vec{w}_{k+1} &= \vec{w}_k - \mu\langle \vec{x}_k\vec{x}_k^{\top}\vec{w}_k - d_k\vec{x}_k\rangle && (1.17)\\
&= \vec{w}_k - \mu\langle \vec{x}_k(y_k - d_k)\rangle && (1.18)\\
&= \vec{w}_k + \mu\langle \vec{x}_k e_k\rangle. && (1.19)
\end{aligned}$$

Dieser Ausdruck kommt einer praktisch brauchbaren Formel zur iterativen Optimierung der Filterkoeffizienten schon recht nahe, störend ist allerdings noch die Mittelung, die nicht vernünftig durchzuführen ist, wenn $\vec{w}_k$ in jedem Abtastschritt verändert werden soll. Die einfachste Möglichkeit dieses Problem zu lösen, nämlich schlichtweg auf die Mittelung zu verzichten, führt zum sogenannten LMS (*least mean square*)-Algorithmus, der auch stochastisches Gradientenverfahren genannt wird,

$$\vec{w}_{k+1} = \vec{w}_k + \mu\vec{x}_k e_k. \tag{1.20}$$

Dieser Algorithmus, der 1970 von Widrow gefunden wurde, benötigt also nur den Vektor von Eingangssignalwerten $\vec{x}_k$ und den aktuellen Fehler e_k zur Optimierung der Filterkoeffizienten. Somit ist er im Gegensatz zur Methode des steilsten Abstiegs auch in Anwendungen einsetzbar, in denen das gewünschte Signal d_k selbst nicht bekannt ist; dies ist für den im Projekt behandelten Rauschkanal der Fall, wie in Abschnitt 1.3 deutlich werden wird. Die Hauptvorteile des LMS-Algorithmus gegenüber der Methode des steilsten Abstiegs sind aber der schon erwähnte Verzicht auf die Forderung nach Stationarität der Signale sowie die Einfachheit der Rechnung (nur skalare Operationen, keine Matrix-Vektor-Multiplikation), was ihn in den meisten Fällen zum Algorithmus der Wahl für adaptive Filter macht.

1.2.4 Konvergenzeigenschaften des LMS-Algorithmus

Konvergieren die Filterkoeffizienten nur langsam, so kann auf beiden Seiten von (1.20) gemittelt werden, ferner können e_k und x_k als stochastisch unabhängig angesehen werden, wodurch der LMS-Algorithmus formal wieder in die Methode des steilsten Abstiegs überführt werden kann. Diese läßt sich bei langsamer Konvergenz in der Form

$$\langle \vec{w}_{k+1} \rangle = (\mathbf{I} - \mu \mathbf{R}) \langle \vec{w}_k \rangle + \mu \vec{p} \tag{1.21}$$

schreiben, wobei $\mathbf{I}$ die Einheitsmatrix bezeichnet. Mit

$$\vec{v}_k = \vec{w}_k - \vec{w}_{\text{opt}} \tag{1.22}$$

wird daraus unter Verwendung von (1.12)

$$\langle \vec{v}_{k+1} \rangle = (\mathbf{I} - \mu \mathbf{R}) \langle \vec{w}_k \rangle + \mu \mathbf{R} \vec{w}_{\text{opt}} - \vec{w}_{\text{opt}} \tag{1.23}$$

$$= (\mathbf{I} - \mu \mathbf{R}) \langle \vec{v}_k \rangle. \tag{1.24}$$

Ziel des LMS-Algorithmus ist es, $\vec{v}_k$ gegen Null streben zu lassen.

Da $\mathbf{R}$ symmetrisch ist, gibt es eine aus den Eigenvektoren von $\mathbf{R}$ bestehende orthogonale Matrix $\mathbf{Q}$, so daß

$$\mathbf{R} = \mathbf{Q} \mathbf{L} \mathbf{Q}^\top \tag{1.25}$$

ist, wobei $\mathbf{L}$ als einzige Einträge die Eigenwerte $\lambda_0, \ldots, \lambda_{l-1}$ von $\mathbf{R}$ auf der Diagonalen stehen hat,

$$\mathbf{L} = \begin{pmatrix} \lambda_0 & & 0 \\ & \ddots & \\ 0 & & \lambda_{l-1} \end{pmatrix} \tag{1.26}$$

(Hauptachsentransformation). Da $\mathbf{R}$ außerdem positiv semidefinit ist, gilt $\lambda_i \geq 0$ für $i = 0, \ldots, l-1$. Mit $\vec{v}'_k = \mathbf{Q}^\top \vec{v}_k$ gilt

$$\langle \vec{v}'_{k+1} \rangle = (\mathbf{I} - \mu \mathbf{L}) \langle \vec{v}'_k \rangle, \tag{1.27}$$

für die i-te Komponente folgt nach k Abtast- bzw. Iterationsschritten

$$\langle {v'_k}^{(i)} \rangle = (1 - \mu \lambda_i)^k {v'_0}^{(i)}. \tag{1.28}$$

Die rechte Seite stellt eine geometrische Reihe dar, die genau dann für $k \to \infty$ gegen Null konvergiert, wenn

$$|1 - \mu \lambda_i| < 1 \tag{1.29}$$

gilt, was sich zu

$$0 < \mu < \frac{2}{\lambda_i} \tag{1.30}$$

umformen läßt. Der LMS-Algorithmus konvergiert genau dann, wenn alle Koeffizienten $w_k^{(i)}$ gegen $w_{\mathrm{opt}}^{(i)}$ konvergieren, wenn also

$$0 < \mu < \frac{2}{\lambda_{\mathrm{max}}} \tag{1.31}$$

gilt, wobei λ_{max} der größte Eigenwert der Autokorrelationsmatrix $\mathbf{R}$ der Eingangssignalwerte x_k ist. Nichtkonvergenz des Algorithmus bedeutet im Bild des Fehlerparaboloiden (Abb. 1.2), daß wegen zu großer Schrittweite μ stets über das Minimum hinweggesprungen wird.

Den größten Eigenwert der Matrix $\mathbf{R}$ zu berechnen ist kein allzu einfaches Unterfangen, so daß eine einfachere Abschätzung für die praktische Anwendung des LMS-Algorithmus sehr hilfreich wäre. Ein solches praktisch nutzbares Kriterium läßt sich finden, indem wir den größten Eigenwert gegen die Summe aller Eigenwerte abschätzen, die gleich der Spur der Matrix $\mathbf{R}$ ist,

$$\begin{aligned} \lambda_{\mathrm{max}} \leq \sum_{i=0}^{l-1} \lambda_i &= \mathrm{Tr}\{\mathbf{R}\} && (1.32) \\ &= l\langle x_k^2 \rangle && (1.33) \\ &= l\sigma_x^2, && (1.34) \end{aligned}$$

wobei σ_x^2 die mittlere Leistung von x_k ist, die ebenso wie die Filterlänge l eine leicht zugängliche und oftmals weitgehend konstante Größe darstellt. Die Abschätzung ist allerdings ziemlich grob, wenn die Konvergenz nicht durch eine einzelne starke Mode dominiert wird.

Ohne Herleitungen folgen nun ein paar kurze Überlegungen zur Konvergenzzeit. Die Zeit, in der der mittlere quadratische Fehler $\langle e_k^2 \rangle$ auf $1/e^2$ (e bezeichnet in diesem Ausdruck die Eulersche Zahl!) abgefallen ist, läßt sich durch

$$\tau \leq \frac{T}{\mu \lambda_{\mathrm{min}}} \tag{1.35}$$

abschätzen, wobei T die Abtastperiode ist. Die dominante Mode (und damit der ganze Algorithmus) konvergiert am schnellsten, wenn $\mu = 1/\lambda_{\mathrm{max}}$ ist, da dann der Klammerausdruck in (1.28) für die Komponente i mit $\lambda_i = \lambda_{\mathrm{max}}$ zu Null wird, der optimale Filterkoeffizient $w_{\mathrm{opt}}^{(i)}$ also in einem Schritt erreicht wird. Für die Konvergenzzeit folgt also

$$\tau \leq \frac{\lambda_{\mathrm{max}}}{\lambda_{\mathrm{min}}} T. \tag{1.36}$$

Minimale Eigenwertspreizung $\lambda_{\mathrm{max}}/\lambda_{\mathrm{min}} = 1$ wird für weißes Rauschen erreicht; je stärker das Eingangsrauschen x_k gefärbt ist, umso langsamer konvergiert der LMS-Algorithmus also.

Ebenfalls ohne Herleitung soll noch angemerkt werden, daß das LMS-Verfahren auch bei gemäß (1.31) gewähltem μ den Fehler $\langle e_k^2 \rangle$ nicht auf den durch (1.14) gegebenen theoretischen Wert $\langle e_k^2 \rangle_{\mathrm{min}}$ minimieren kann. Bedingt durch die Schätzung des Gradienten unter Verzicht auf die Mittelung bleibt ein *zusätzlicher* Restfehler (*excess error*)

$$\langle e_k^2 \rangle_{\mathrm{exc}} \approx \frac{\mu}{2} l \sigma_x^2 \langle e_k^2 \rangle_{\mathrm{min}} \tag{1.37}$$

bestehen, worin wieder die Filterlänge l und die mittlere Leistung σ_x^2 des Eingangssignals auftreten. Im Bild des Fehlerparaboloiden (Abb. 1.2) läßt sich der Restfehler so deuten, daß durch ungenaue Schätzung des Gradienten die Filterkoeffizienten $w_k^{(i)}$ in eine falsche Richtung verändert werden, so daß das Minimum nicht erreicht wird, obwohl die Schrittweite μ klein genug ist.

Abschließend läßt sich bemerken, daß alle hier vorgestellten Abschätzungen eher theoretische Bedeutung für den Beweis der Konvergenz und der Stabilität des LMS-Algorithmus haben. In der Praxis sind sie meist von geringem Nutzen. So kann die Konvergenz beispielsweise stark beschleunigt werden, indem anfangs eine große Schrittweite μ gewählt wird, um die Filterkoeffizienten schnell in die Nähe der optimalen Werte zu bringen, die dann verkleinert wird, um die Konvergenz langsam zu verfeinern. Eine Divergenz des Algorithmus kann dadurch verhindert werden, daß bei wachsendem Fehler $\langle e_k^2 \rangle$ die Schrittweite μ verkleinert wird.

Diese praktischen Überlegungen leiten über zum zweiten Teil des Protokolls, in dem die durchgeführten Experimente beschrieben werden.

1.3 Die Praxis

1.3.1 Erste Schritte

Nachdem die Theorie bekannt ist, ist es an der Zeit, diese in die Praxis umzusetzen. Es soll versucht werden, mit dem LMS eine Strecke zu erkennen. Diese Strecke ist ein akustisches System. Ein Holzkanal, in den über einen Lautsprecher ein Rauschen gegeben wird. Dieses Rauschen soll am anderen Ende des Kanals nicht mehr zu hören sein. Erreicht werden soll dieses mit einem zweiten Signal, das über einen zweiten Lautsprecher (*Aktuator Lautsprecher*) an einem späteren Punkt in den Kanal gegeben wird und so angepaßt werden soll, daß es das durch die Übertragungsfunktion der Kanalstrecke gefärbte Rauschen auslöscht.

Es wurde zunächst mit einem sehr einfachen Versuchsaufbau überprüft, daß ein LMS-Algorithmus auch wirklich ein System erkennen kann, indem ein Tiefpaß adaptiert wurde. Dazu war es nötig, ein von den Assistenten vorgefertigtes C-Programm fertigzustellen. Das Programm läuft im Rahmen des an diesem Institut entwickelten Programms `ppdsp` (Pentium Pro Digital Signal Processing), das eine graphische Oberfläche zur Verfügung stellt und die Kommunikation mit den DSP-Karten übernimmt. In diesem Programm kann man die Wirksamkeit des adaptiven Filters in drei Fenstern beobachten, in denen verschiedene Größen und ihre Fouriertransformierte dargestellt werden können.

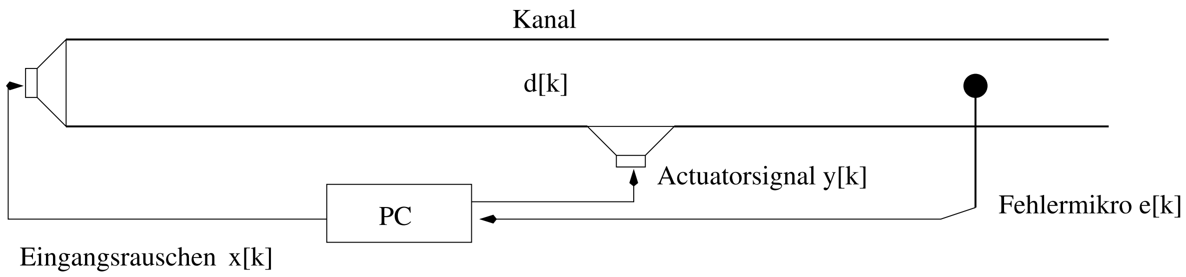

Abb. 1.3: Erster Aufbau des Kanals

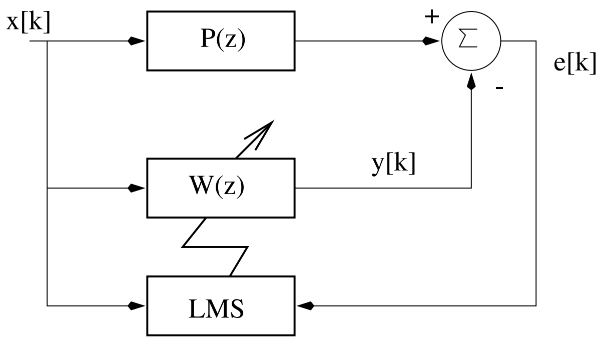

Abb. 1.4: Blockschaltbild der Systemerkennung

1.3.2 Der erste Versuch am Kanal

In Abb. 1.3 sieht man den prinzipiellen Aufbau des Versuches. Es wird mit einem PC ein Rauschen x_k auf den ersten Lautsprecher gegeben, das intern im Rechner auch in den LMS gegeben wird. Dort werden unter Benutzung des Fehlersignals e_k die Filterkoeffinzienten des adaptiven Filters so bestimmt, daß das damit gefilterte Signal y_k das durch die Kanalstrecke gefärbte Rauschen eliminiert. Ist der Fehler noch zu groß, versucht der LMS, das Filter zu optimieren. Dabei besteht noch die Möglichkeit, an dem Konvergenzparameter μ zu drehen und so die Schrittweite zu Beginn hoch zu wählen, bzw. später kleiner zu halten, damit sich das Filter gut anpassen kann. Ein Problem können Quermoden im Holzkanal sein. Treten solche auf, kann es passieren, daß man mit dem Mikro nicht alle Moden aufnimmt oder mit dem Aktuator anregt, wenn sich Mikro oder Lautsprecher gerade in einem Knoten befinden. Da die Samplingfrequenz der Wandler eine Tiefpaßfunktion hat kann man darauf Einfluß nehmen. Die Samplingfrequenz der Wandler wurde so gewählt, daß sie zu der Grenzfrequenz des Holzkanals für Quermoden ($f_g = 850\,\mathrm{Hz}$) paßt: 800 Hz. Zudem wurde noch ein Tiefpaßfilter vorgeschaltet, damit das Samplingtheorem nicht verletzt wird. In Abb.1.4 sieht man das Blockschaltbild einer Systemerkennung. Die Idee ist es nun, daß das adaptive Filter $W(z)$ das physikalische System $P(z)$, das den Weg des Schalls vom Rauschlautsprecher bis zum akustischen Additionspunkt am Aktuatorlautsprecher erkennen und optimieren soll. Bei dieser Herangehensweise war nur eine sehr geringe Rauschbefreiung zu beobachten.

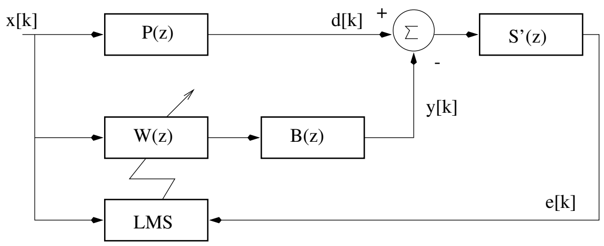

Abb. 1.5: Blockschaltbild mit Sekundärstrecke

1.3.3 Die Sekundärstrecke

Die Erklärung für das Nichtfunktionieren findet man, wenn man sich die Systeme noch einmal genauer betrachtet. Zunächst ist das System $P(z)$ nicht nur die Wegstrecke vom Rauschlautsprecher bis zum Aktuator. Es kommen noch die Übertragungsfunktionen des DA-Wandlers, des Tiefpaß und des Rauschlautsprechers dazu, denn das Signal, das PC-intern als x_k verwendet wird ist das dort erzeugte Rauschen, das erst noch durch die Strecke PC→DA→TP→LS→Kanal bis Aktuator muß.
Weiter ist das Fehlersignal e_k, mit dem man im Programm rechnet, nicht der tatsächliche Fehler an der Stelle des Mikros. Das Signal geht also vom Aktuator durch den Kanal ins Mikro, über den Tiefpaß und durch den AD Wandler in den PC. Berechnet man nun das Aktuatorsignal y_k, muß man noch berücksichtigen, daß das am PC ausgegebene Signal wieder durch einen DA Wandler geht, durch einen Tiefpaß und über den Aktuatorlautsprecher.

Abb. 1.5 zeigt diese Strecken:
$B(z)$: PC→DA→TP→AktuatorLS.
$S'(z)$: Kanal vom Aktuator bis Mikro→TP→AD→PC.
Das ganze System $S(z) = S'(z)B(z)$ wird als *Sekundärstrecke* bezeichnet.

Nun zieht man die Addition, die bisher akustisch im Kanal stattgefunden hat, in den PC mit hinein. Das beschreibt die Abb. 1.6. Das Aktuatorsignal geht aus dem PC durch die Sekundärstrecke und wird zu dem Rauschen addiert, das den Kanal und das Mikro passiert hat. Dabei wurde der Einfluß, den der Aktuatorlautsprecher noch auf dieses Signal hat, entsprechend berücksichtigt: $\frac{P(z)S(z)}{B(z)}$. Geht man davon aus, daß sich die Systeme zeitlich nicht oder nur sehr langsam ändern, kann man die Sekundärstrecke $S(z)$ auch vor die Systeme $W(z)$ bzw. $\frac{P(z)}{B(z)}$ ziehen, wie es in Abb. 1.6 zu sehen ist. Dann fehlt nur noch eine Schätzung $\hat{S}(z)$, mit der man das Signal x_k vorfiltert, bevor man es in den LMS gibt, damit man wieder das Schaltbild einer Systemerkennung erhält. Diese besorgt man sich durch eine Streckenerkennung bevor man mit der eigentlichen ANC (*Active Noise Control*) beginnt. Dabei gibt man ein Rauschen auf den Aktuator und nimmt das Signal am Fehlermikro wieder auf.

Hat man sich einmal die Filterkoeffezienten von $\hat{S}(z)$ besorgt, kann in einem zweiten Schritt die eigentliche ANC beginnen. Und sie funktioniert auch sehr gut. Es werden dabei Rauschunterdrückungen von bis zu 30 db erreicht.

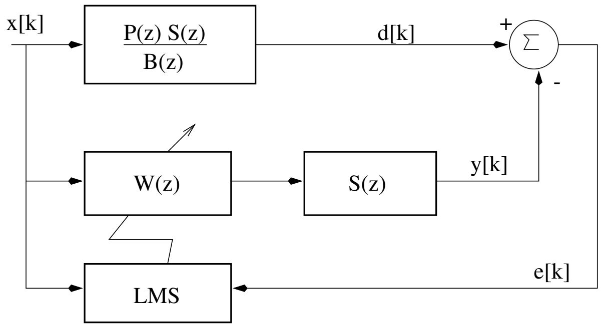

Abb. 1.6: Formales Durchziehen der Sekundärstrecke ..Teil 1

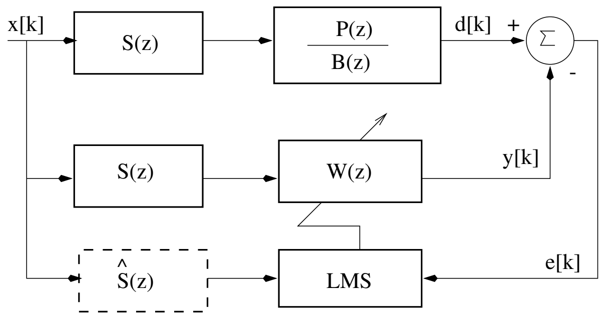

Abb. 1.7: Formales Durchziehen der Sekundärstrecke ..Teil 2

1.3.4 Die Rückkopplungsstrecke

Für gewöhnlich erzeugt man sich das Rauschen allerdings nicht selber. Vielmehr hat man einen Kanal (z.B. den Lüftungsschacht einer Klimaanlage), in dem es rauscht, und möchte diesen Lärm eindämmen. Dazu nimmt man das Rauschen im Kanal auf, um es als Eingangssignal x_k (durch $\hat{S}(z)$ gefiltert) dem LMS zuzuführen. Dabei sollte man beachten, daß das Signal an den Tiefpässen, den Wandlern und in der digitalen Signalverarbeitung im PC einige Zeit benötigt. Diese Zeit darf nicht länger sein als die, die der Schall vom Referenzmikro bis zum Aktuator braucht, da man sonst ein akausales Filter konstruiert.

Die Abb. 1.8 zeigt, wie das Referenzmikro in den Versuchskanal eingebracht wurde. Mit diesem Signal als Eingangsrauschen wurde das Programm erneut durchlaufen und ergab als Ergenbnis so gut wie keine Rauschunterdrückung.

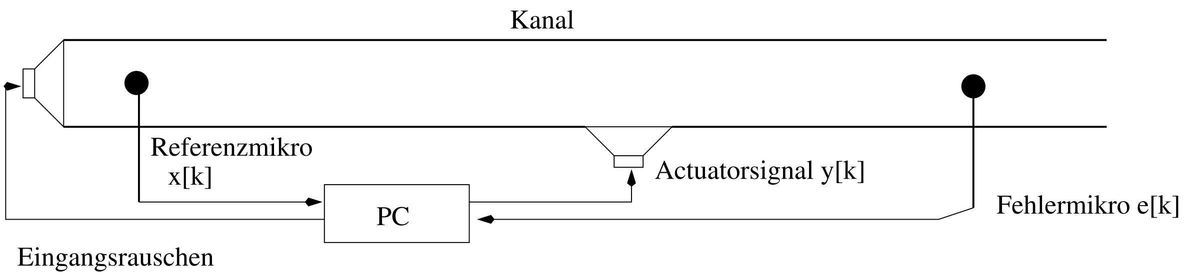

Abb. 1.8: Endgültiger Aufbau des Kanals

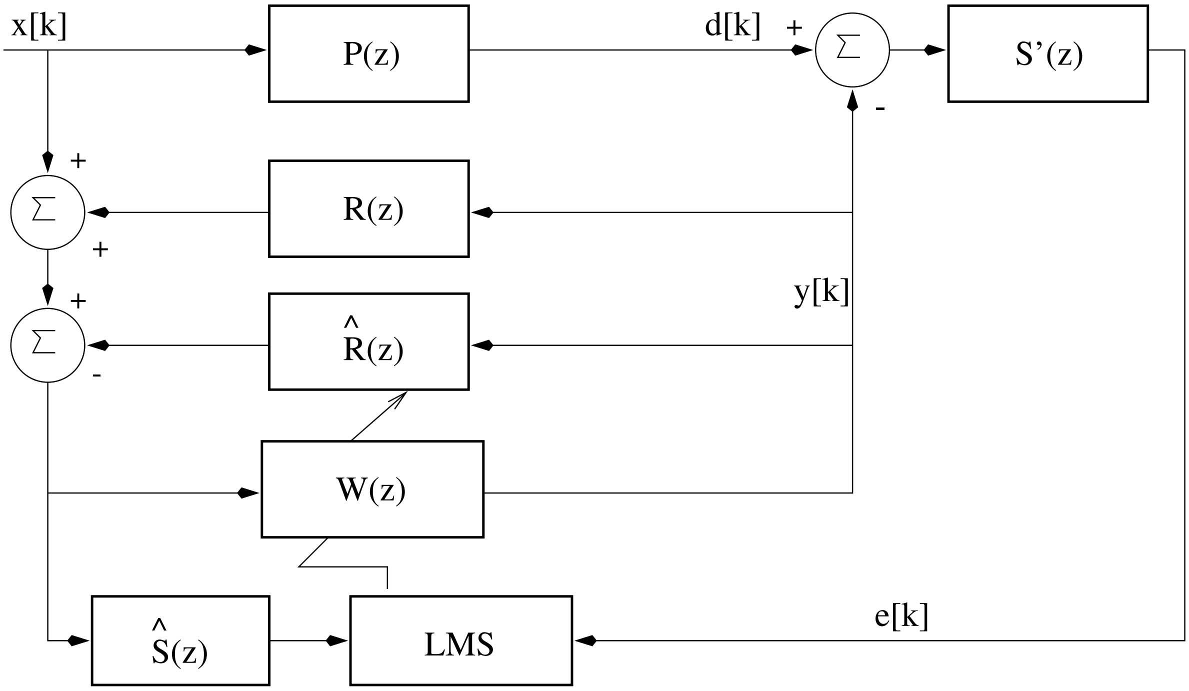

Abb. 1.9: Vollständiges Blockschaltbild mit Rückkopplungsstrecke

Die Lösung dieses Problems ist die Erkennung der *Rückkopplungsstrecke*, die beschreibt, daß das vom Aktuator erzeugte Rauschen auch am Referenzmikro gemessen wird. Dieses Signal ist aber in dem Rauschen, das dem Kanal folgt und beseitigt werden soll, nicht vorhanden, darf also auch nicht als Bestandteil des Eingangsrauschens x_k betrachtet werden. Mit einer Schätzung $\hat{R}(z)$ der Rückkopplungsstrecke kann man nun, wie in Abb. 1.9 gezeigt, diesen Einfluß eliminieren. Es wird eine weitere Streckenerkennung vorgeschaltet, um die Schätzung der Rückkopplungsstrecke zu erhalten, indem ein Rauschen auf den Aktuator gegeben wird, das man am Referenzmikro wieder aufnimmt, und diese Strecke mit dem LMS erkennt.

Mit dem nun kompletten System (mit Sekundärstrecke und Rückkopplungsstrecke) wurde erneut versucht, eine aktive, adaptive Schallfeldkontrolle zu erzielen. Wie Abb. 1.10 zeigt, wurden so sehr gute Rauschbefreiungen bis zu der durch die Samplingrate festgelegten Grenzfrequenz erzielt.

1.3.5 Anwendung

Eine klassische Anwendung dieses Prinzips liegt sicherlich in der „Entlärmung" von Klimaanlagen. Das Verfahren eignet sich sehr gut dazu, mit herkömmlichen, passiven, Absorbern kombiniert eingesetzt zu werden, da diese für tiefe Frequenzen sehr groß werden. Die aktiven Absorber hingegen eignen sich eher für die tiefen Frequenzen. Da man für hohe Frequenzen eine höhere Samplingrate benötigt, wird die zur Verfügung stehende Rechenzeit kürzer. Man bräuchte also mehr Rechenleistung.

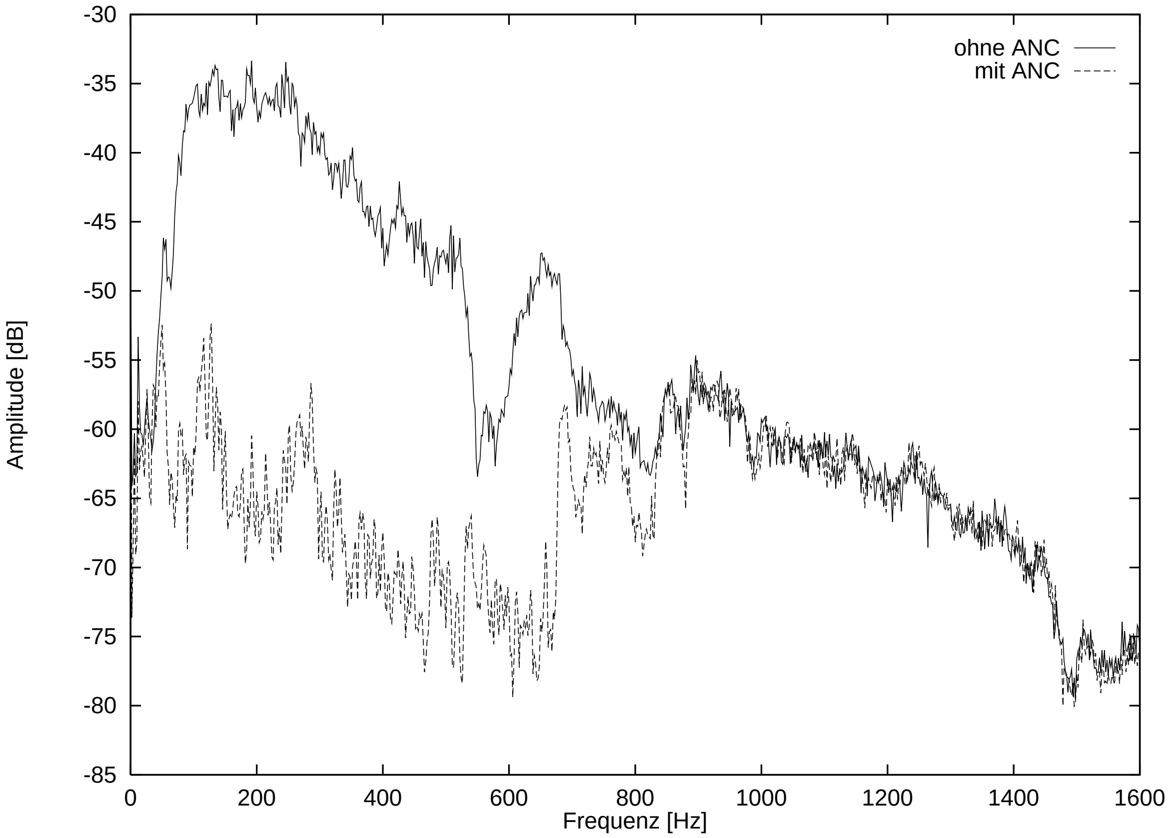

Abb. 1.10: Spektrum des Rauschens mit und ohne ANC

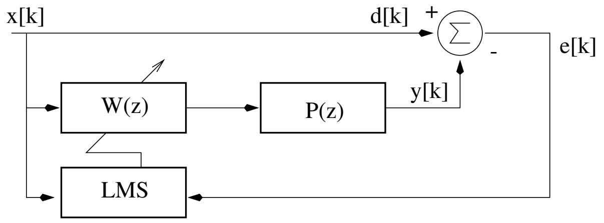

Abb. 1.11: Problem der inversen Streckenerkennung

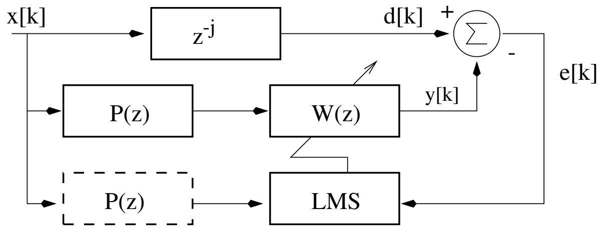

Abb. 1.12: Lösung der inversen Streckenerkennung

1.3.6 Inverse Streckenerkennung

Bei der *inversen Streckenerkennung* geht es darum, ein Signal so vorzufiltern, daß der Einfluß eines Systems eliminiert wird. Auch in diesem Bereich läßt sich das Prinzip der adaptiven Filter einsetzen. Dazu wurden der Aufbau und das Programm leicht modifiziert, so daß die Übertragungsfunktion eines Tiefpasses neutralisiert werden konnte. Es lassen sich auch weiterführende Anwendungen denken, die hier aber nicht weiter verfolgt wurden. Z.B. könnte die Sekundärstrecke invertiert werden, so daß man das Eingangssignal nicht mehr durch die Schätzung von $S(z)$ vorfiltert.

Abb. 1.11 zeigt die Problemstellung für die inverse Streckenerkennung als Blockschaltbild, mit $P(z)$ dem zu invertierenden Filter und $W(z)$ dem adaptiven. Dabei soll das Filter $W(z)$ so adaptiert werden, daß es den Einfluß des Filters $P(z)$ aufhebt. Würde man sich also die Signale y_k und $d_k{=}x_k$ auf einem Oszilloskop ansehen, sollte man keinen Unterschied sehen. Man erkennt sofort ein Problem: die Zeiten durch die Filter und den Rechner sind nicht beliebig klein, wohingegen das urpsrüngliche Signal, das zum Vergleich herangezogen wird, unverzögert zum Additionspunkt gelangt. Man braucht also auf jeden Fall ein Verzögerungsglied, mit dem das Signal auf dieser Strecke aufgehalten wird. Außerdem kann der LMS $P(z)$ so nicht richtig erkennen.

Die Abb. 1.12 zeigt die Lösung des Problems: man schaltet wieder eine Streckenerkennung vorweg und filtert dann das dem LMS übergebene Signal mit der Systemabschätzung vor. Zudem verzögert man das Eingangssignal mit z^{-j}. Dann vergleicht der LMS die richtigen Werte und man kann den Einfluß des – in diesem Falle – Tiefpasses aufheben.

Es ist sicherlich interessant, die Sekundärstreckenintegration bei der ANC einmal mit diesem Verfahren zu realisieren.