School of Engineering

Lab 6: Recursive Least Squares

1. Einleitung

Wie Sie letzte Woche gesehen habe, kann man mit einem adaptiven Filter tieffrequente Störungen in einem Audio- oder Sprachsignal unterdrücken, welche von «aussen» eindringen, siehe Blockdiagramm in Abbildung 1.

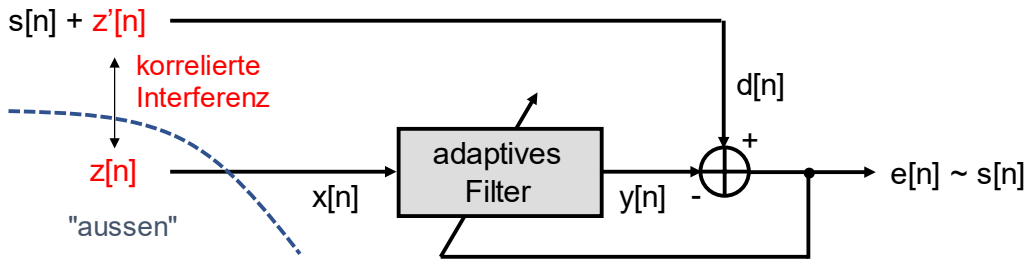

Abbildung 1: Blockdiagramm Active Noise Cancelling (ANC).

Ein Vorteil des *Rekursive Least Squares (RLS)-Algorithmus* ist dessen schnellere Konvergenz im Vergleich zum LMS-Algorithmus.

2. Aufgabe: ANC mit RLS

a) Nehmen Sie das Python File anc_rls.py als Vorlage, um eine Active Noise Cancellation mittels RLS (anstatt LMS) zu implementieren.

b) Messen Sie die Laufzeit (z.B. mit der time or timeit Library) der LMS und RLS Implementierung.

c) Vergleichen Sie das Konvergenzverhalten der beiden Implementierungen.

3. Aufgabe: Polynomial Fit mittels Least Squares

Im Folgenden soll die Funktion $f(x) = sin(\pi x)$ im Intervall $-1 \leq x \leq 1$ mit einem Polynom 5. Grades $p(x) = b_5 \cdot x^5 + b_4 \cdot x^4 + \ldots + b_1 \cdot x + b_0$ im Least-Squares-Sinn approximiert werden.

a) Werten Sie hierfür die Funktion d = f(x) an $N = 101$ Stellen $x_n = -1 + 0.02(n-1)$, $n = 1, \ldots, 101$ aus.

b) Stellen Sie nun das LS-Problem auf und bestimmen Sie die Koeffizienten b_{LS} durch Lösen der Normalengleichung

$$b_{\mathrm{LS}} = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}y$$

Tipp: Bringen Sie
$d_n = f(x_n)$ und
$y_n = p(x_n) = b_5 \cdot {x_n}^5 + b_4 \cdot {x_n}^4 + \ldots + b_1 \cdot x_n + b_0$
für n=1, …, 101 in Matrixform.

c) Plotten Sie die Originalfunktion f(x) sowie das Polynom p(x) mit den bLS-Koeffizienten. Plotten Sie ebenfalls den Fehler e(x) = f(x) – p(x). Was beobachten Sie?

d) Bestimmen Sie nun den Koeffizientenvektor b mit numpys `polyfit` Funktion und vergleichen Sie die erhaltenen Polynomkoefizienten.

e) **Optional**: Versuchen Sie den Sourcecode der `polyfit` Implementierung https://github.com/numpy/numpy/blob/v2.0.0/numpy/lib/_polynomial_impl.py#L442-L687 nachzuvollziehen. Ist die `polyfit` komplett in Python implementiert (pure Python?)