

Funktionsweise Digitaler Audio Workstations

In einer Digitalen Audio Workstation (DAW) sind Audioeffekte als separate Programme, die sogenannten Plug-ins, realisiert. Die DAW, in diesem Zusammenhang auch oft Host genannt, organisiert Aufnahme, Abspielen, Routing und Mixing von Audio-Dateien, sowie deren Verarbeitung durch die Plug-ins. Dabei erhalten die Plug-ins von der DAW jeweils Blöcke mit N Samples und müssen Blöcke der selben Länge zurückgeben. Da die Festlegung der Blocklänge zentral erfolgt, muss ein Plug-in mit jeder beliebigen Blockgröße umgehen können.

Die Arbeitsweise einer DAW und einfacher Plug-ins wird in den folgenden Tutorien anhand einiger Beispiele simuliert. Dabei wird die Aufgabe des Hosts von einem Matlab-Skript übernommen, das Audiosignale lädt oder erzeugt und Blockweise an eine Funktion – das Plug-in – übergibt. Innerhalb der Plug-ins soll die Verarbeitung der Audiosignale Sampleweise erfolgen und ohne Matlab-interne Funktionen (`filter()`) gearbeitet werden.

Digitale Audioeffekte in Matlab - Iargelmentierung eines digitalen Filters

a) Wir wollen einen digitalen Filter implementieren, der in Abbildung 1 dargestellt ist. Stellen Sie hierzu zunächst die Übertragungsfunktion auf und stellen sie Amplituden- und Phasenfrequenzgang, sowie ein Pol-Nullstellen-Diagramm grafisch dar (z.B. für $\alpha = 0.5$).

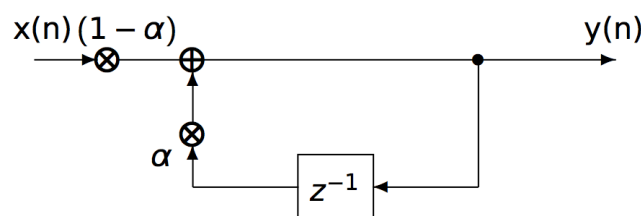


Abbildung 1: Blockschaltbild eines digitalen Filters

b) Erzeugen Sie ein Host-Skript, welches eine DAW mit einem Dummy-Plugin `blockFilter` simuliert. Das heißt ein Skript, das eine Mono-Datei einliest und in Blöcken der Länge N an das Dummy-Plugin weitergibt, welches seinen Namen trägt, weil es nichts mit dem Eingang macht und ihn sofort ausgibt. Implementieren Sie `blockFilter` als Matlab-Funktion der Form: `y = blockFilter(x)`

c) Ändern Sie das Dummy-Plugin so ab, dass es ein Audiosignal mit einem beliebigen Koeffizientensatz (b, a) filtert und das Ergebnis zurückgibt.

$$y = \text{blockFilter}(b, a, x) \quad , \text{ mit}$$

x, y : Ein- und Ausgangssignal

b : nicht rekursive Filterkoeffizienten $[b_0, b_1, \dots, b_N]$

a : rekursive Filterkoeffizienten $[1, a_1, a_2, \dots, a_N]$

Testen Sie die Funktion mit dem Filter aus Abb. 1 und einem Audio-File aus dem Downloadverzeichnis.

d) Überprüfen Sie die Berechnung der ersten beiden Ausgangssamples innerhalb von `blockFilter.m` mit Hilfe der Debugging Funktion von Matlab.