Projektarbeit

Ansteuerungsfilter für den Ikosaederlautsprecher

Christian Jochum Peter Reiner

Graz, am 22. Mai 2007

Betreut durch:

DI Franz Zotter

Institut für Elektronische Musik und Akustik
Universität für Musik und darstellende Kunst Graz

Inhaltsverzeichnis

1 Aufgabenstellung

Seit Beginn 2006 arbeitet das IEM an einem neuen, kugelförmigen Wiedergabesystem, das die Form eines regelmäßigen 20 flächigen Polyeders besitzt. Sämtliche Teilflächen dieses hohlen Ikosaeders sind mit Lautsprechern bestückt, welche diskret bespielt werden können. Bei geeigneter Signalaufbereitung für jeden Einzellautsprecher können akustische Abstrahlungswirkungen unterschiedlichster Ausformung erzielt werden. Diese noch sehr junge Wiedergabemethode ermöglicht völlig neue Perspektiven bei der Wiedergabe von Instrumentalklängen, bei Raumakustik-Messungen, etc.. (sphärische akustische Holographie/Wellenfeldsynthese) Ziel dieses Projekts ist die Umsetzung solcher Anspeisungsfilter in Pure Data. Im Kern der Implementierung soll eine Blockfilterung im Frequenzbereich stehen (OLA / OLS ... overlap-add / overlap-save). Um mehrere Abstrahlungsmuster (Filtersätze) frei kombinieren zu können, soll ein bestehendes Blockdiagramm eingehalten werden.

2 Einleitung

Die Hauptaufgabe der zu berechnenden Filter besteht darin, aus den (geeignet gewichteten) 16 Basisfunktionen der spärischen Harmonischen die 20 Signale für die einzelnen Lautsprecher des Ikosaeders zu berechnen. Gleichzeitig soll aber auch das Übersprechen [1] zwischen den einzelnen Lautsprechern in geeignetem Maße kompensiert werden. Wie in Abbildung 1 schematisch dargestellt, wurde die Ansteuerung des Ikosaederlautsprechers mittels Pure Data realisiert, wobei die benötigten Filterkoeffizienten bzw. Matrizen zuvor in Matlab berechnet wurden (siehe Kapitel 4).

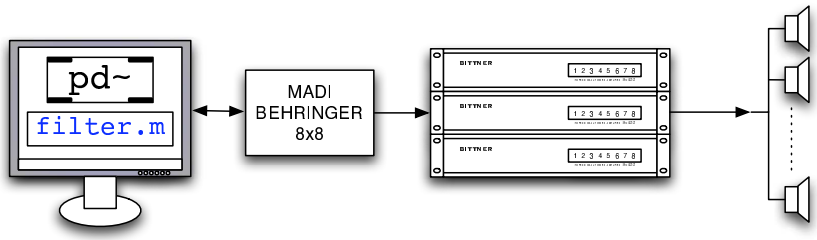

Abbildung 1: Ansteuerung des Ikosaederlautsprechers

Ein Überblick über den Signalfluss lässt sich aus dem in Abbildung 2 dargestellten Blockschaltbild gewinnen, wobei sich im Block *mtx_cmul* das in Abbildung 3 dargestellte Rechenschema verbirgt. Dieses Blockschaltbild entspricht dem in der Aufgabenstellung vorgegebenen insoweit, als das mehrere Abstrahlungsmuster kombiniert werden können. Zugunsten eines geringeren Berechnungsaufwandes wurde allerdings die Reihenfolge der einzelnen Blöcke etwas verändert und die Gewichtung der einzelnen sphärischen Harmonischen dem *OLA* und der *FFT* vorangestellt.

3 Abschätzung des Berechnungsaufwands

3.1 Berechnungsaufwand FFT

Unter der Voraussetzung, dass $N = 2^p$, wobei $p \in \mathbb{N}$ ist, lässt sich der Aufwand für die Berechnung der komplexen (I)FFTs auf

$$N \cdot ld\,(N) \tag{1}$$

Operationen pro FFT (oder pro Block) abschätzen [2]. Um eine Vorstellung über den Rechenaufwand pro Sample zu erhalten, wählen wir für die Hopsize

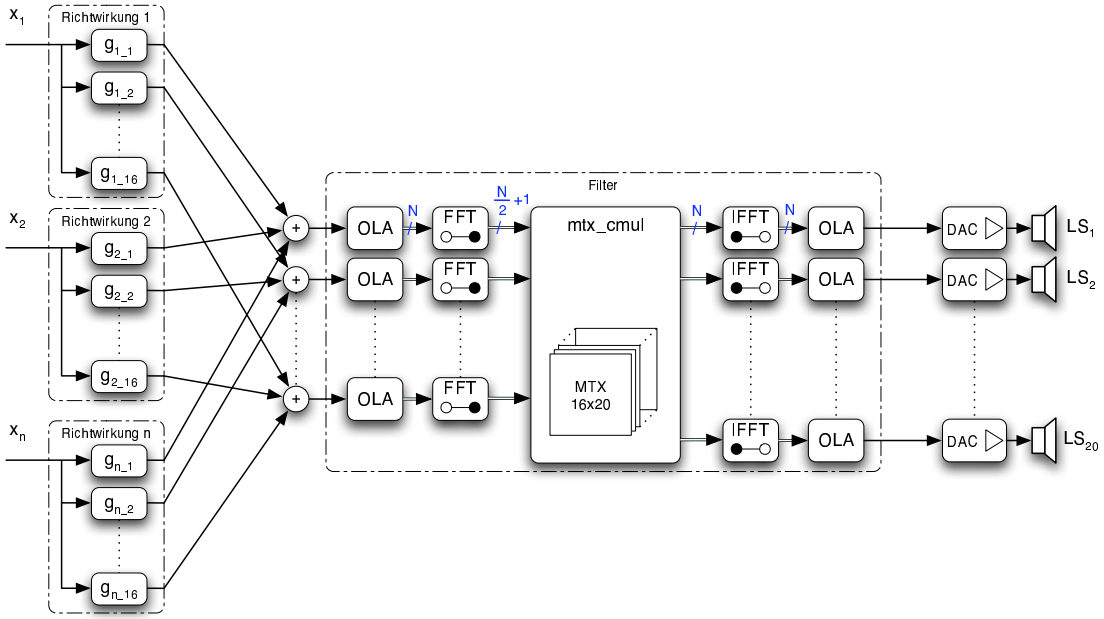

Abbildung 2: Schema der Ikosaederansteuerung

$$\eta = \frac{N}{2} \tag{2}$$

und erhält mit Gleichung 1:

$$2 \cdot ld\,(N) \tag{3}$$

Operationen pro Sample.

Da wir die FFT für die 16 gewichteten Basisfunktionen und die IFFT für 20 Lautsprecher berechnen wollen, ergibt sich daraus ein Transformationsaufwand von ca.

$$640 \cdot ld\,(N) \tag{4}$$

Operationen pro Sample.

3.2 Berechnungsaufwand der Richtcharakteristik

Zwischen FFT und IFFT wird in der in Abbildung 3 dargestellten Art und Weise verfahren. Das bedeutet, dass die Fouriertransformierten der gewichteten Basisfunkionen zeilenweise in eine Matrix geschrieben werden. Die sodann spaltenweise ausgelesenen Frequenzbins werden mit den zugehörigen Matrizen multipliziert und

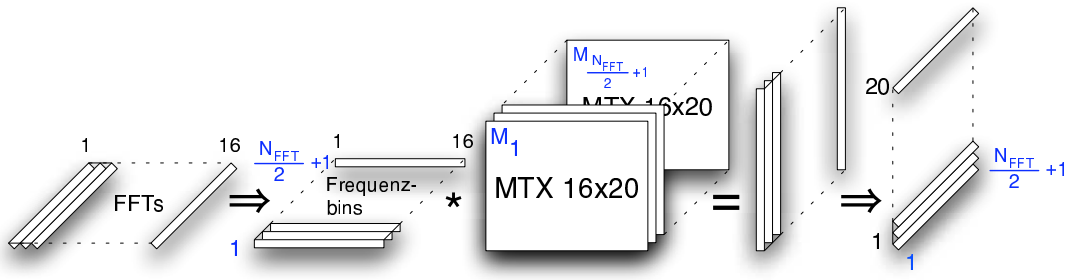

Abbildung 3: Berechnungsschema

wiederum spaltenweise in eine zweite Matrix geschrieben. Die Signale für die 20 Lautsprecher ergeben sich aus den IFFTs der Zeilen dieser zweiten Matrix.

Der Berechnungsaufwand hierfür ergibt sich zu

$$\left[\underbrace{(16_{mul} + 15_{add}) \cdot 20}_{\mathbf{v}\cdot\mathbf{M}} \cdot \underbrace{4 + 2 \cdot 16_{add} \cdot 20}_{\Re+\Im^{1}}\right] \cdot \left(\frac{N}{2} + 1\right) \tag{5}$$

Operationen pro Block bzw.

$$3120 \cdot \left(1 + \frac{2}{N}\right) \tag{6}$$

pro Sample.

3.3 Berechnungsaufwand in Summe

Setzt man für die Länge der FFT $N = 1024$, ergibt sich aus der Summe der Gleichungen 4 und 6 ein Aufwand von $6400 + 3127 = 9527$ pro Sample und mit einer Samlingrate von 44,1 kHz ergeben sich daraus 420.140.700 Operationen pro Sekunde. Da unser System für einen obere Gerenzfrequenz von 4 kHz konzipiert wurde [3], können wir über ein Downsampling um den Faktor 4 eine weitere Reduktion des Berechnungsaufwandes auf ca. 105 Millionen Operationen pro Sekunde erreichen.

4 Filterberechnung

Da sowohl die Matrizierung der sphärischen Harmonischen auf die Lautsprecher, die Kompensation der Übertragungsfunktioen der aktiven Lautsprecher als auch

[1] Bei jeweils 2 Vektoren und Matrizen: $\mathbf{v}_{\Re} \cdot \mathbf{M}_{\Re} - \mathbf{v}_{\Im} \cdot \mathbf{M}_{\Im}$, $\mathbf{v}_{\Re} \cdot \mathbf{M}_{\Im} + \mathbf{v}_{\Im} \cdot \mathbf{M}_{\Re}$

die Kompensation des Übersprechens statisch sind, muss die Berechnung der Filtermaritzen nicht in Echtzeit geschehen und kann vorbereitend in Matlab erfolgen.

$$\mathbf{W}(k) = \mathbf{H}^{-1}(k) \cdot \mathbf{Y}_{SH} \cdot e^{-j\frac{2\pi}{N} \cdot k \cdot \Delta N} \tag{7}$$

4.1 Berechnung der Zielfunktionen $\mathbf{Y}_{SH}$

Da uns die ersten 16 sphärischen Harmonischen als Basis für unsere Richtcharakteristika dienen, benötigen wir ein Filter, welches uns eben jene auf unsere 20 Lautsprecher abbildet. Die dafür notwendigen, rekursiven Berechnungsvorschriften konnten aus einem vorliegenden Matlab-Skript übernommen werden. Darin wird ein Set an sphärischen Harmonischen, bis zu einem gewünschten Grad, an den angegebenen Winkelpaaren (Lautsprecherpositionen) ausgewertet. Die daraus gewonnenen Vektoren stehen sodann als Zielfunktion bei der Systeminversion zur Verfügung.

4.2 Berechnung der Systemmatrizen des MIMO-Systems $\mathbf{H}(k)$

Als erstes wurden die in [1] ermittelten Impulsantworten der einzelnen Lautsprecher, mit jeweils einem halben, 50 Samples langen Hann-Fenster am Anfang und am Ende, auf eine Länge von 2^{12} bzw. 4096 Samples zurechtgeschnitten. Da beim Betrieb des Ikosaeder-Richtlautsprechers nicht die gesamte Bandbreite von 22.050 Hz fehlerfrei bespielt werden kann [4], wird eine vierfache Reduktion der Abtastrate mit dem Befehl `resample` auf die Filterantworten angewendet. Dabei sind zur vollständigen Beschreibung des reellwertigen MIMO Systems, aufgrund von Symmetrieeigenschaften der DFT, 513 Frequenzpunkte ausreichend.

4.3 Phasenterm

Da unsere Filter für die Richtwirkung nur aus den erster 513 Werten der Impulsantwort bestehen und alle übrigen Samples auf 0 gesetzt werden, muß nun noch dafür Sorge getragen werden, dass die wesentlichen Teile der Impulsantwort auch in der ersten Hälfte liegen. Hierzu verschieben wir die Impulsantwort um $\Delta N = 290$ Samples nach vorne (siehe Abbildung 4).

$$\delta\left[n - \Delta N\right] \quad \overset{}{\underset{N}{\longmapsto}} \quad e^{-j\frac{2\pi k}{N} \cdot \Delta N} \tag{8}$$

4.4 Invertieren der Systemmatrizen

Bei der Inversion der Systemmatrizen ergeben sich zwei Probleme. Zum einen ist die numerische Inversion von schlecht konditionierten Matrizen sehr ungenau

und darum an sich äusserst ungünstig. Um dieses, vor allem bei tiefen Frequenzen auftretende Problem zu umgehen, bedienen wir uns der Konditionierung bzw. Regularisierung (4.4.1).

Zum anderen führen die äußerst kleinen Werte in jenen $\mathbf{H}(k)$, welche durch den Antialiasing-Filter bedämpft wurden, zu sehr großen Werten in $\mathbf{H}^{-1}(k)$ und somit auch in unseren Filtern, was sich in weiterer Folge durch Übersteuern schon bei kleinen Eingangssignalen bemerkbar machen würde. Dem begegnen wir wie in 4.4.2 beschrieben.

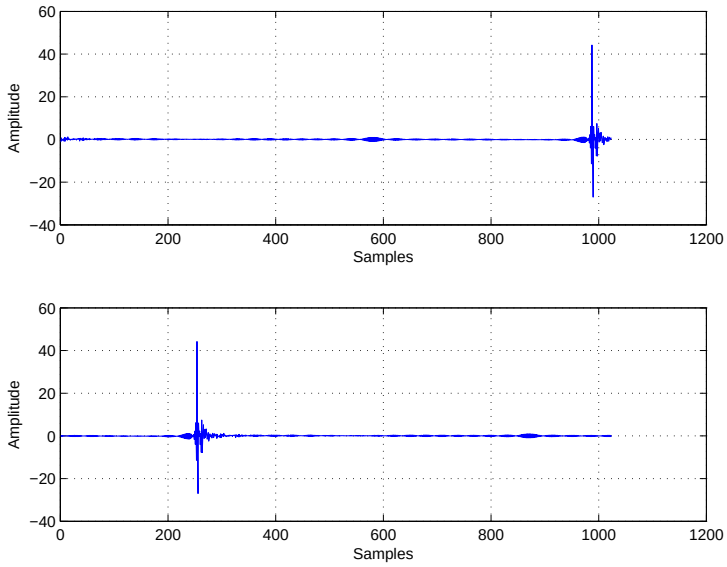

Abbildung 4: Impulsantwort vs. verschobene Impulsantwort

4.4.1 Die Konditionszahl

Die Konditioszahl ist unter anderem ein Maß für die Genauigkeit der Inversion einer Matrix. Sie ist für reguläre Matrizen definiert als

$$\kappa(\mathbf{A}) := \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \tag{9}$$

Je kleiner die Konditionszahl einer Matrix ist, desto robuster reagiert sie auf Änderungen in ihren Einträgen und desto akkurater ist ihre Inversion. Wie aus Abbildung 5 ersichtlich, sind unsere Matrizen, bis auf einige wenige, recht gut konditioniert und darum gut invertierbar. Als Grenzwert für $\kappa(\mathbf{H})$ wurde $g = 30$ gewählt und um einen sanften Übergang zu erhalten, wird eine Arkustangens

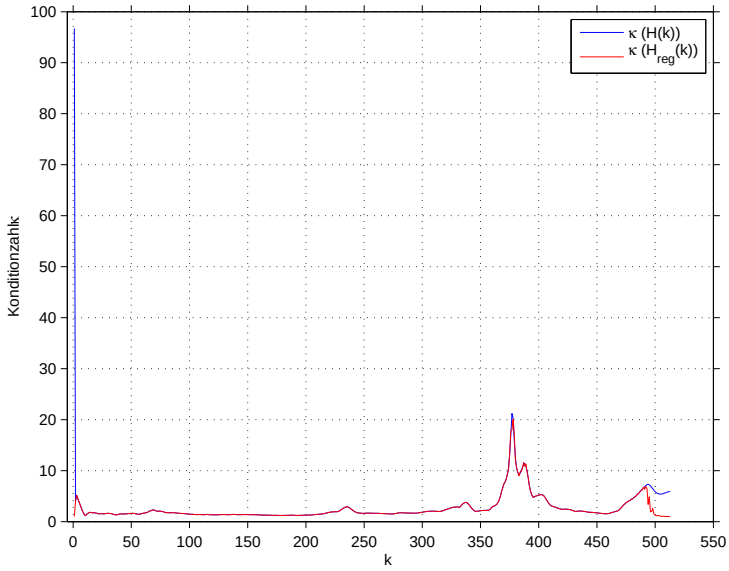

Abbildung 5: Konditionszahlen mit und ohne Regularisierung

Funktion verwendent. $\kappa_g\big(\mathbf{H}(k)\big)$ ergibt sich somit aus

$$\kappa_g\big(\mathbf{H}(k)\big) = 0.5 + \frac{1}{\pi} \cdot arctan\left(\Big(\kappa\big(\mathbf{H}(k)\big) - g\Big) \cdot s\right) \tag{10}$$

wobei mit $s = 10$ die Steilheit der Arkustangens Funktion eingestellt wurde.

4.4.2 Anhebung bei hohen Frequenzen

Da der linearphasige Antialiasing-Filter des Befehls `resample` die Übertragungsfunktionen des gesamten MIMO-Systems Tiefpass filtert, sind die Einträge der Matrizen bei hohen Frequenzen sehr klein. Da dies bei der Inversion zu außergewöhnlich großen Werten führen würde, wird die Hauptdiagonale von $\mathbf{H}(k)$, also die Gewichte für die aktiv bespielten Lautsprecher, ab einer Frequenz von 5265 Hz ($k = 490$) verstärkt. Auch hier wird keine scharfe Grenze verwendet, sondernn anhand einer Sinusquadratfunktion eingefadet.

$$\gamma(k) = \begin{cases} 0, & 1 \leq k < 490 \\ sin^2\left(\frac{\pi}{2} \cdot \frac{k - 490}{513 - 490}\right), & 490 \leq k \leq 513 \end{cases} \tag{11}$$

Die Regularisierungsentscheidung c berechnet sich also zu

$$c\big(\mathbf{H}(k), k\big) = \kappa_g\big(\mathbf{H}(k)\big) + \gamma(k) \tag{12}$$

und die regularisierten Matrizen ergeben siech zu

$$\mathbf{H}_{reg}(k) = \mathbf{H}(k) \cdot \Big(1 - c\big(\mathbf{H}(k), k\big)\Big) + c\big(\mathbf{H}(k), k\big) \cdot \mathbf{I} \cdot e^{-j\frac{2\pi}{N} \cdot k \cdot \Delta N_{reg}} \cdot G_{amp} \tag{13}$$

4.4.3 Fehler durch die Optimierung

Da nunmehr $\mathbf{H}_{reg}$ anstelle von $\mathbf{H}$ invertiert wird, entsteht ein Fehler

$$\mathbf{E} = \left(\mathbf{I} - \mathbf{H}\mathbf{H}_{reg}^{-1}\right) \mathbf{Y}_{SH} \tag{14}$$

Zur Veranschaulichung des selbigen wurde mittels Singulärwertzerlegung

$$svd(\mathbf{E}) = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V} \tag{15}$$

die Varianz σ_e^2 berechnet und in Abbildung 6 dargestellt.

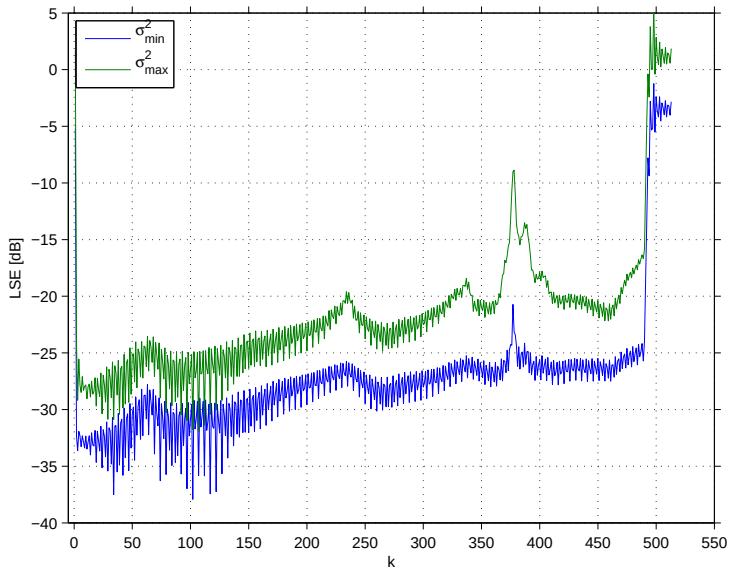

Abbildung 6: Fehler der SH über der Frequenz

4.5 Filterlänge

Wie in 4.3 kurz erwähnt muss sichergestellt sein, dass sich die Länge der berechneten Filter auf die ersten 513 Werte beschränkt und alle übrigen Werte auf Null gesetzt werden. Im Gegensatz zur Verschiebung, welche mittels Phasenterm auch im Frequenzbereich durchführbar ist, ist für das Nullsetzen der restlichen 511 Samples

der Umweg über den Zeitbereich unumgänglich. Die in zurück in den Zeitbereich transformierten Filtersätze (Abbildung 8) werden, wie schon in 4.2, unter Zuhilfenahme eines halben, 50 Samples langen Hann-Fensters ein- und ab dem 461 Sample auf Null ausgefadet.

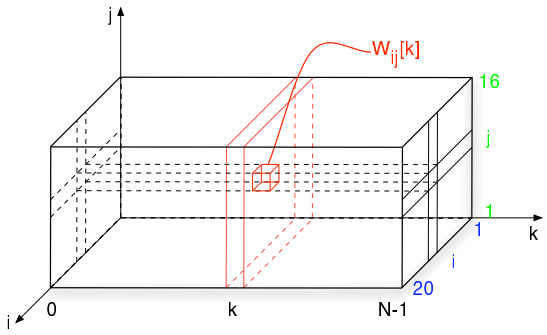

Abbildung 7: Filtersatz im Frequenzbereich

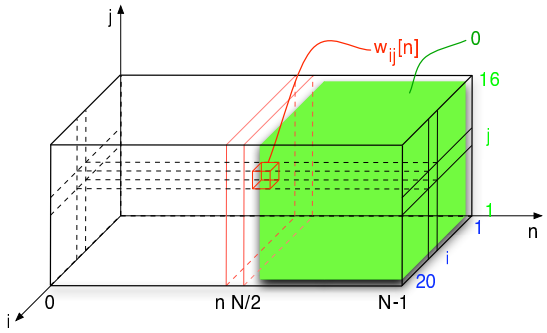

Abbildung 8: Filtersatz im Zeitbereich

4.6 Exportieren der Daten für pd

Um die in Matlab generierten Filterdaten in pd weiterverarbeiten zu können, ist es sinnvoll diese in Real- und Imaginärteil zu trennen und in der in Abbildung 9 dargestellten Form zu exportiren. Damit die Files von pd gleich als Matrizen erkannt werden, kann mittels Texteditor die Größe vornan gestellt werden.

5 Implementierung in Pure Data

Wie in Abbildung 10 schematisch dargestellt ist, besteht die Implementierung aus acht Blöcken wobei nebst *pd* die Externals *iemlib, iemmatrix, iemsp* und *zexy*

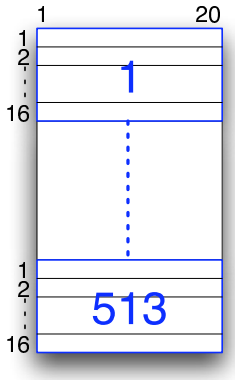

Abbildung 9: Exportformat für pd

benötigt werden.

5.1 iko_filter

Hier werden die 16 gewichteten sphärischen Harmonischen der unterschiedlichen Abstrahlcharakteristika mittels `catch~` aufsummiert und an `iko_calculation` weitergegeben. Die daraus berechneten 20 Lautsprechersignale werden tiefpassgefiltert und sodann ausgegeben.

5.2 iko_calculation

In diesem Subpatch erfolgt ein Downsampling der Zuspielsignale um den Faktor 4 mittels `block` sowie das zeilenweise Befüllen der Matrizen wobei die verwendete Blockgröße auf 512 Samples erhöht wird. Die sequentielle Anordnung der `iko_order` Blöcke erzwingt die Reihenfolge der Abarbeitung mittels "order-forcing" und stellt sicher, dass die 16×512 Matrix korrekt und vollständig gefüllt wird bevor die Bearbeitung des nächsten Blocks beginnt. Weiters werden hier die via Matlab generierten Filtersätze geladen und zur weiteren Berechnung an `iko_mul` weitergegeben. Die so aus den Signalen entstandenen Matrizen werden dann der Reihe nach, von den in Abbildung 10 dargestellten Blöcken weiterverarbeitet.

5.3 iko_order

In `iko_order` wird das Zuspielsignal in eine Liste verpackt, welcher dann die jeweilige Reihe, in der sie in die Matrix geschrieben werden soll, vornan gestellt wird. Die darin enthaltenen `dummy` Ein- und Ausgänge dienen wiederum der in 5.2 angesprochenen Erzwingung der Reihenfolge.

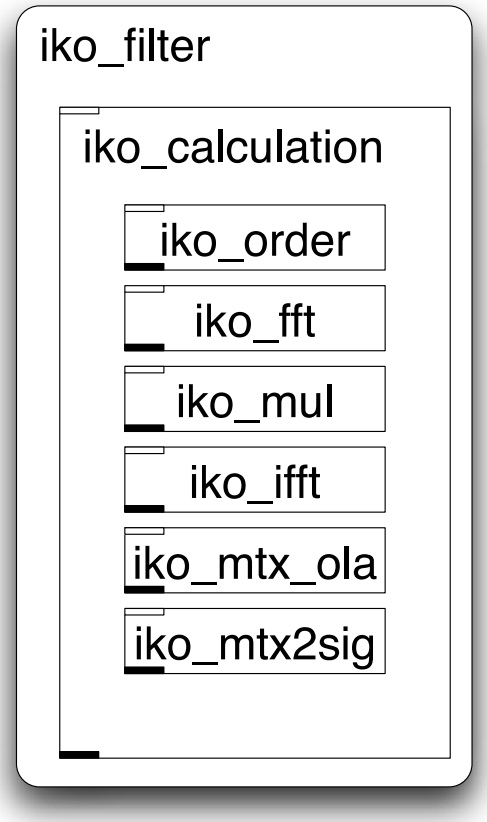

Abbildung 10: Implementierungsschema

5.4 iko_fft

Wie der Name schon sagt, erfolgt hier die FFT der Zuspielsignale. Zusätzlich wird mittels `mtx_resize` die Fensterung für Overlap Add [5] vorgenommen, indem mit Nullen auf die doppelte Länge erweitert wird.

5.5 iko_mul

Hier findet die eigentliche (komplexe) Matrixmultiplikation, wie in Abbildung 3 skizziert, statt. Frequenz für Frequenz werden sowohl Real- als auch Imaginärteil aus den jeweiligen Signalmatrizen ausgelesen und mit den passenden, aus der Filtermatrix ausgelesenen Werten multipliziert. Die so entstehenden Matrizen werden mit zugehörigem Frequenzindex ausgegeben. Sind alle Frequenzen eines Blockes abgearbeitet, wird dies mittels `ready` signalisiert und `iko_ifft` kann mit der Bearbeitung beginnen.

5.6 iko_ifft

Um mittels `mtx_rowrifft` alle aus `iko_mul` entstandenen Matrizen auf einmal in den Zeitbereich rück zu transformieren, werden sie zuerst in jeweils eine Matrix für Real- und Imaginärteil zusammengefasst. Die IFFT erfolgt dann zum jeweils durch `ready` vorgegebenen Zeitpunkt.

5.7 iko_mtx_ola

Die jeweils erste Hälfte der jeweiligen Matrix wird mit der Zweiten ihrer Vorhergegangenen addiert.

5.8 iko_mtx2sig

Die den Lautsprechersignalen entsprechenden Einträge in den Matrizen werden Reihe für Reihe ausgelesen, in Signale gewandelt und den entsprechenden Ausgängen zugeführt.

6 Test

Um die Funktion unseres Filters zu testen gibt es mehrere Möglichkeiten. So könnte man zum Beispiel den für die Bestimmung des MIMO-Systems [3] verwendeten Messaufbau mit Mikrofonen verwenden. Auch eine Messung mittels Laservibrometer, wie jene zur Bestimmung des Übersprechens zwischen den einzelnen Lautsprechern, währe denkbar. Wir haben uns aber für eine Verifikation mittels des menschlichen Gehörs entschieden und zu diesem Zwecke einen weiteren pd-Patch geschrieben. Hierzu wird eine maximale Hauptkeule generiert, indem als Zielfunktion $b(\varphi, \vartheta)$ ein Punkt bzw. ein Diracimpuls auf der Kugeloberfläche bewegt wird.

$$b\left(\varphi, \vartheta\right) = \delta\left(\varphi - \varphi_t\right) \delta\left(\vartheta - \vartheta_t\right) \tag{16}$$

Die Koeffizienten der Zerlegung B_{nm} in Kugelflächenfunktionen (Spherical Harmonics) berechnen wir zu

$$B_{nm} = \mathcal{SHT}\left\{\delta\left(\varphi - \varphi_t\right) \delta\left(\vartheta - \vartheta_t\right)\right\} \tag{17}$$

bzw.

$$B_{nm} = \iint_{\mathbb{S}^2} \delta\left(\varphi - \varphi_t\right) \delta\left(\vartheta - \vartheta_t\right) \cdot Y_{nm}^{*}\left(\varphi, \vartheta\right) \cdot d\Omega \tag{18}$$

Dieses Integral lässt sich auflösen auf

$$B_{nm} = Y_{nm}^{*}\left(\varphi_t, \vartheta_t\right) \tag{19}$$

Sofern die Inversion unseres Systems korrekt ist, erhalten wir im Idealfall

$$b(\varphi, \vartheta) \approx \sum_{n=0}^{3} \sum_{m=-n}^{n} B_{nm} \cdot Y_{nm}(\varphi, \vartheta) \tag{20}$$

`test_iko_filter` berechnet die Gewichtung der sphärisch harmonischen Basisfunktionen, wobei Azimuth und Elevation via Slider vorgegeben werden können.

7 Diskussion

Die anfängliche Sorge, dass der Berechnungsaufwand unverhältnismäßig groß und somit die Realisierung in Echtzeit impraktikabel werden würde, stellte sich als unbegründet heraus. Dies liegt vor allem daran, dass ein Gutteil der Berechnungen mit Matlab durchgeführt werden konnte und weiters, mittels Downsampling, eine wesentliche Reduktion der Berechnungen in pd erreicht wurde.

8 Erkenntnisse und Ausblick

Der Test unseres Filters verlief sehr zufriedenstellend. Die sich bewegende Hauptkeule konnte gut geortet und ihre Position erahnt werden, ohne das dabei etwaig auftretende Nebenkeulen auffällig gewesen wären. Dies war insbesondere in unmittelbarer Nähe des Ikosaederlautsprechers gut hörbar - was sich wiederum dadurch erklärt, als dass sich unser Berechnungen immer auf die Lautsprecheroberfläche, also r_0, nicht aber auf einen bestimmten Radius r im Raum bezogen. Es konnte gezeigt werden, dass sich auch die exzentrische Kombination aus kleinem und großem Ikosaederlautsprecher sehr gut zur gerichteten Abstrahlung eignen. Es wird erwartet, dass auch eine messtechnische Überprüfung der Abstrahlungsformen mittels Mikrofonen oder Laservibrometrie zu einem sehr positiven Ergebnis führen wird.

Literatur

[1] P. Reiner and C. Jochum, "Erfassung und Evaluierung der Übersprecheigenscheften des Ikosaederlautsprechers," Projektarbeit, IEM, Feb. 2007. [Online]. Available: http://iem.at/projekte/acoustic/awt/erfassung/erfassung.pdf

[2] A. V. Oppenheim and R. W. Schafer, *Zeitdiskrete Signalverarbeitung*, 3rd ed. Oldenburg, 1999, ch. 9.3.

[3] F. Zotter and A. Sontacchi, "Report 39/07 Icosahedral Loudspeaker Array," IEM, Jan. 2007. [Online]. Available: http://iem.at/projekte/publications/iem_report/report39_07/

[4] F. Zotter, A. Sontacchi, and R. Höldrich, "Modeling a Spherical Loudspeaker System as Multipole Source," *Tagungsbeitrag zur Jahrestagung DAGA 07, Deutsche Gesellschaft für Akustik*, 2007. [Online]. Available: http://iem.at/projekte/publications/paper/modeling/daga07.pdf

[5] S. W. Smith, *The scientist and engineer's guide to digital signal processing.* California Technical Publishing, 1999, ch. 18, pp. 311–314. [Online]. Available: http://www.dspguide.com/CH18.PDF