

# Modulares Mess- und Auswertesystem für Mikrofonarraymessungen

Ennes Sarradj

*Juniorprofessur Aeroakustik, Brandenburgische Technische Universität Cottbus, 03046 Cottbus*

## Einleitung

Bei Messungen zur Lokalisierung und Charakterisierung von Schallquellen ist der Einsatz von Mikrofonarrays zu einem Standardverfahren geworden. Dabei ist der Einsatz vor allem bei Messungen an umströmten und an bewegten Objekten besonders vorteilhaft. Oft unterscheiden sich die Messaufgaben hinsichtlich ihrer Anforderungen sehr stark. So sind je nach konkreter Aufgabenstellung sehr unterschiedliche örtliche Auflösungen nötig, verschiedene Frequenzbereiche von Interesse sowie abweichende mechanische Anforderungen an die Messapparatur vorhanden. Bei der praktischen Umsetzung sieht sich der Anwender jedoch auch weiteren Problemen, wie der Auswahl des richtigen Signalverarbeitungsalgorithmus, der effizienten Durchführung der Messungen und der Speicherung und Verwaltung der beträchtlichen Datenmengen gegenüber. Im Folgenden soll als Lösungsansatz ein modulares Mess- und Auswertesystem für Mikrofonarraymessungen im Überblick dargestellt werden, dass im Verlauf des Einsatzes von Mikrofonarrays bei verschiedenen Untersuchungen an der BTU Cottbus entwickelt wurde. Dabei stand die leichte Anpassbarkeit an neue Messaufgaben, die Erweiterbarkeit sowie eine flexible Datenverarbeitung im Vordergrund.

## Modulares Messsystem

Das Messsystem besteht aus austauschbaren Mikrofonarrays, austauschbarer Signalerfassungshardware und einer modularen Software zur Signalverarbeitung.

Die bislang realisierten Mikrofonarrays besitzen zwischen 8 und 92 Mikrofonen und haben Aperturen (Querabmessungen) zwischen 38 cm und 4 m. Damit ist es möglich, sowohl Messungen im Modellmaßstab für hohe Frequenzen bis ca. 40 kHz durchzuführen, als auch Messungen an Objekten im Originalmaßstab bei tiefen Frequenzen. Bei Mikrofonarrays für hohe Frequenzen ist eine möglichst präzise Positionierung der Mikrofone wichtig. Dem entsprechend wurden für diese Mikrofonarrays jeweils Elektret-Mikrofonkapseln in eine gefräste Aluminiumplatte eingesetzt (siehe Abb. 1) und damit eine Genauigkeit von  $\pm 0,2$  mm erreicht. Für größere Aperturen ( $> 1.5$  m) ist ein solcher Aufbau nicht realisierbar. Gleichzeitig kann eine weniger präzise Positionierung hingenommen werden. Deshalb wurden in diesen Fällen die Arrays kleiner Apertur durch demontierbare Erweiterungsrahmen und -arme ergänzt, auf denen handelsübliche  $\frac{1}{4}$ "-Elektret-Messmikrofone befestigt wurden (Abb. 1).

Die Software zur Signalverarbeitung wurde in erster Linie entwickelt, um mit möglichst geringem Aufwand verschiedene Auswerte-Algorithmen einsetzen zu können. Eine weitere wichtige Anforderung besteht darin, dass die

Software sowohl interaktiv bedienbar ist als auch die Möglichkeit eines automatisierten bzw. programmgesteuerten und parallelen Betriebs in leistungsfähigen Rechnerumgebungen ermöglicht. Während die interaktive Bedienung das einfache Einrichten, die Kalibrierung und den Test des Messaufbaus ermöglicht, bildet die programmgesteuerte Auswertung die Grundlage für die effiziente Auswertung großer Datenmengen, wie sie beispielsweise bei Messserien im Windkanal auftreten.

Die Realisierung der Software erfolgte Plattformunabhängig (Linux, Windows) in der Programmiersprache Python, wobei rechenintensive Routinen in C++ implementiert sind. In der Software werden alle für die Durchführung und Auswertung von Mikrofonarraymessungen notwendigen Arbeitsschritte abgebildet und unterstützt:

- Erkennung und Steuerung der Datenerfassungshardware, wobei beliebige Konfigurationen unterstützt werden
- Einlesen von Messdaten in verschiedenen Datenformaten (HDF5, DATX, MATLAB, CSV/ASCII)
- Test und Kalibrierung des Arrays, Generierung von Kalibrierdatensätzen
- Verwaltung von Kalibrierdaten und verschiedenen Array-Mikrofon-Anordnungen
- Verwaltung von auf die Messung bezogenen Strömungsdaten (Strömungsgeschwindigkeit und -richtung)
- Erzeugung von 2D- und 3D-Berechnungsgittern
- verschiedene Beamforming-Algorithmen: Delay & Sum, Capon-Beamforming (adaptive Beamforming), MUSIC, Eigenwert-Beamforming, DAMAS, CLEAN-SC, Orthogonales Beamforming
- Auswertung in Form von Schalldruckpegel-Kartierungen (Maps) für Einzelfrequenzen und Frequenzbänder
- Berechnung von Schalldruckpegel-Spektren einzelner Teilschallquellen mittels Integration über entsprechende Bereiche der Maps
- 3D-Auswertung und Integration
- Kommunikation mit einer SQL-Datenbank zur Verwaltung von Messdaten und Ergebnissen

Die Software ist objektorientiert aufgebaut und verwendet die Technik des „reactive programming“. Damit werden die teilweise recht aufwändigen Berechnungen erst dann durchgeführt, wenn die Ergebnisse tatsächlich benötigt werden. Zusätzlich werden einige bereits berechnete Zwischenergebnisse in einem Zwischenspeicher (Cache) abgelegt und müssen so nicht wiederholt berechnet werden.

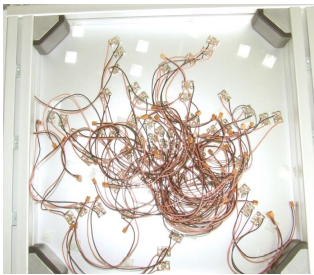


Abbildung 1: Array mit 64 Elektretmikrofonen, Rückseite (offen und ohne Elektronik)

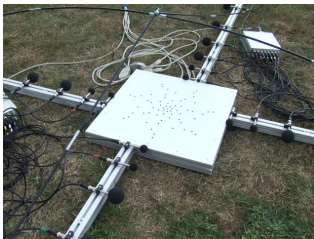


Abbildung 2: Demotierbare Erweiterungsrahmen mit Messmikrofonen

```
t = TimeSamples (name='measured_data.h5')
m = MicGeom (from_file='mic_geom.xml')
g = RectGrid (x_min=-0.6,x_max=0.0,y_min=-0.3,y_max=0.3,z=0.68,increment=0.01)
cal = Calib (from_file='calibration_data.xml')
f = PowerSpectra (time_data=t1,block_size=4096,window="Hanning",overlap='50%',calib=cal)
b = BeamformerBase (freq_data=f1.grid=g,mpos=m,r_diag=False)
map = L_p (b.synthetic(4000,0))
sector = (-0.3,-0.08,-0.2,0.08)
spectrum = b.integrate(sector)
```

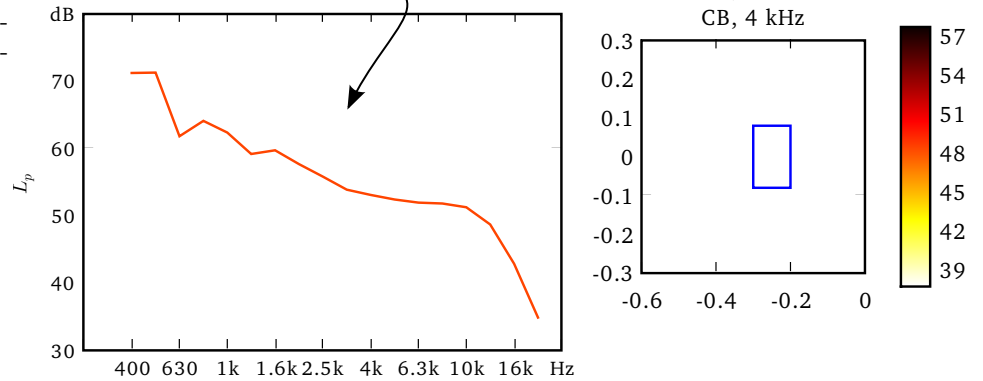


Abbildung 3: Minimalbeispiel für programmgesteuerte Auswertung einer Messung

### Beispiele und Anwendung

In Abb. 3 ist ein Minimal-Beispiel für die programmgesteuerte Auswertung einer Messung dargestellt. Dabei werden zuvor gemessene Schalldruck-Zeit-Funktionen aus einem HDF5-File und eine bestimmte Mikrofongeometrie und die Kalibrierdaten jeweils aus einem XML-File gelesen. Nach der Definition eines rechteckigen 2D-Berechnungsgitters und der Kreuzspektralmatrix mit bestimmten Parametern für die FFT wird der einfache Delay & Sum-Algorithmus (BeamformerBase) ausgewählt. Im Ergebnis des Programmlaufs enthält die Variable *map* eine Schalldruck-Kartierung und die Variable *spectrum* das einem Bereich zugeordnete Schalldruck-Spektrum. Damit können bei Bedarf weitere Berechnungen durchgeführt werden. Die eigentliche Berechnung und Datenverwaltung ist vollständig gekapselt, so dass der Anwender keine speziellen Kenntnisse darüber benötigt.

Ein wichtiger Einsatzfall für die Mikrofonarrayesstechnik sind Messreihen, bei denen z.B. im Windkanal verschiedene Konfigurationen eines Testobjekts bei verschiedenen Versuchsbedingungen untersucht werden. Dabei werden gegebenenfalls mehrere tausend Messungen durchgeführt und gemeinsam ausgewertet (z.B. [1, 2]). Abb. 4 zeigt schematisch den Einsatz des beschriebenen Systems im Zusammenspiel mit einer SQL-Datenbank. Dabei werden die Messwerte vom Mikrofonarray in einem Festplattenspeicher abgelegt (ca. 0.5-1 GByte pro Messung), während weitere zugehörige Messdaten und Versuchsbedingungen in einer SQL-Datenbank abgelegt werden. Die je nach eingesetztem Algorithmus sehr rechenzeitintensive Verarbeitung der Schalldruck-Messwerte erfolgt zeitlich parallel auf einem Rechnercluster. Dazu werden die für eine Berechnung notwendigen weiteren Daten wie Kalibrierdaten, Temperatur, Strömungsgeschwindigkeit usw. aus der Datenbank gelesen, in die auch die Berechnungsergebnisse

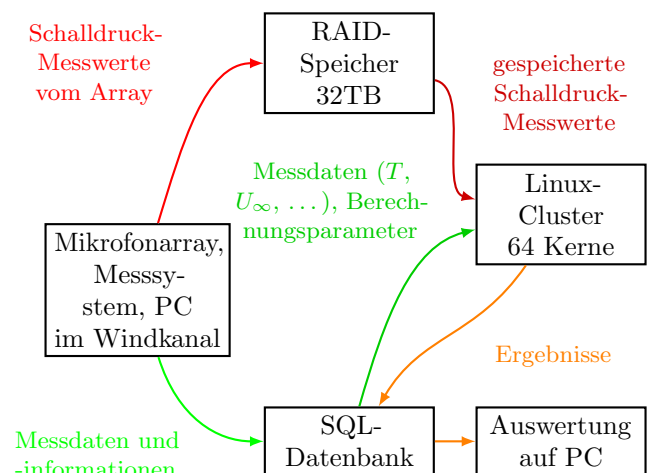


Abbildung 4: Übersicht zur Datenbankanbindung

eingetragen werden. Die weitere Auswertung erfolgt dann durch Zugriff auf die Datenbank. Dadurch ist es beispielsweise möglich, die Ergebnisse zu normieren bzw. deren Abhängigkeit von dimensionslosen Größen zu untersuchen.

### Literatur

[1] Geyer, T. ; Sarradj, E. ; Fritzsche, C.: Measurement of the noise generation at the trailing edge of porous airfoils. In: *Experiments in Fluids* 48 (2010), Nr. 2, S. 291-308

[2] Giesler, J. ; Sarradj, E.: Measurement of broadband noise generation on rod-airfoil-configurations. In: *15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference)*, AIAA paper 2009-3308, 2009