

## Akustisk kamera

Utvecklandet av en akustisk kamera från enkla komponenter

Kandidatarbete inom institutionen för arkitektur och samhällsbyggnadsteknik

William Olofsson Maronen, Mattias Qian, Ali Elhage

**INSTITUTIONEN FÖR ARKITEKTUR OCH SAMHÄLLSBYGGNADSTEKNIK**



KANDIDATARBETE 2024

# Akustisk kamera

ACEX11-24-60



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

INSTITUTIONEN FÖR ARKITEKTUR OCH SAMHÄLLSBYGGNADSTEKNIK  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg 2024

Akustisk kamera  
Utvecklandet av en akustisk kamera från enkla komponenter  
Ali El Hage, Mattias Qian, William Olofsson Maronen

© ALI EL HAGE, MATTIAS QIAN, WILLIAM OLOFSSON MARONEN 2024.

Handledare: Jens Ahrens  
Examinator: Jens Forssén  
Institutionen för arkitektur och samhällsbyggnadsteknik  
Chalmers Tekniska Högskola  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Omslagbild: Slutprodukten till vänster och en videoutmatning från en utav mätningarna som utfördes i det ekofria rummet till höger.

Skrivet i L<sup>A</sup>T<sub>E</sub>X  
Göteborg 2024

## Förord

Det här kandidatarbetet är författarnas slutmoment av deras kandidatutbildning på Chalmers Tekniska Högskola inom civilingenjörsprogrammet i elektroteknik. Kandidatarbetet skrevs på instutionen för arkitektur och samhällsbyggnadsteknik.

Gällande ansvarsfördelning så ansvarade William för simuleringen av den optimala geometrin samt MATLAB-implementation av den kommersiella lösningen. Ali och Mattias ansvarade för ritningen av PCB:n och konstruktionen av den akustiska kameran. Ansvarsgrupperna skrev och ansvarade även helt själv för sina korresponderade bakgrund- och metoddelar av rapporten.

William Olofsson Maronen, Mattias Qian, Ali Elhage, Göteborg, Maj 2024



## Abstract

The aim with this project was to construct an acoustic camera from simple components and to motivate the optimal geometry of its microphone array given manufacturer's limitations through the use of simulations. In order to minimize needless complexity the project was limited to simple far field beamforming and 2D microphone arrays. Simulations showed that UCA was the best middleground due to its unparalleled frequency range without sacrificing much MSL or BW. Sadly this geometry couldn't be realized due to PCB not being made in time. Instead a commercial solution implemented in MATLAB was opted for. The performance of the finished product was then finally measured inside an anechoic chamber. The resulting acoustic camera ticked all of the goals and performed in line with simulations, even if not as well as the optimal geometry implemented with FPGA hardware would have. There are multiple areas to improve such as latency and ease of use which makes this project suitable for a continuation project.

## Sammanfattning

Målet med det här projektet var att konstruera en akustisk kamera från enkla komponenter och att motivera den optimala geometrin av dess mikrofonuppsättning givet dagens tillverkningsbegränsningar genom simulationer. Projektet avgränsade sig till enkel fjärrfältlobstyrning och 2D-mikrofonuppsättningar. Simulationerna visade att UCA var den optimala geometrin tack vare dess enastående frekvensomfång utan större bekostnad av varken MSL eller BW. Tyvärr kunde inte den optimala geometrin realiseras tack vare att PCB:n aldrig hann ritas och beställas. Istället valdes en kommersiell lösning som implementerades i MATLAB. Prestandan av den akustiska kameran mättes sedan i ett ekofritt rum. Den resulterande akustiska kameran uppfyllde alla mål och i enlighet med simuleringarna om än inte lika bra som en med optimal geometri implementerad i FPGA skulle. Det finns flera förbättringsområden utöver geometrin såsom smidighet och latens vilket gör det här arbetet relevant för att bygga vidare på i ett fortsättningsprojekt

Keywords: acoustic camera, acoustics, beamforming, signal processing.



# Lista över förkortningar

Nedan är listan av förkortningar som har använts genom arbetets gång i arbiträr ordning:

DAS	Delay-and-Sum
AF	Array factor
RAF	Receiving array factor
ULA	Uniform Linear Array
URA	Uniform Rectangular Array
UCA	Uniform Circular Array
UHA	Uniform Hexagonal Array
HPBW	Half power beamwidth
MSL	Maximal sidelobe level



# Innehåll

<b>Lista över förkortningar</b>	<b>viii</b>
<b>Lista över figurer</b>	<b>xiii</b>
<b>Lista över tabeller</b>	<b>xvi</b>
<b>1 Introduktion</b>	<b>1</b>
1.1 Syfte och mål . . . . .	1
1.2 Avgränsingar . . . . .	2
<b>2 Teori och teknisk bakgrund</b>	<b>3</b>
2.1 Mikrofonuppsättning . . . . .	3
2.2 Signalteori . . . . .	4
2.2.1 Direktivitet . . . . .	4
2.2.1.1 Strålningsdiagram . . . . .	5
2.2.1.2 Lobbredd . . . . .	6
2.2.1.3 Maximala relativa sidlobsnivån . . . . .	6
2.2.1.4 Vikning . . . . .	6
2.2.2 Lobstyrning . . . . .	7
2.2.2.1 Delay and Sum . . . . .	7
2.2.2.2 Fjärrfältslobstyrning av UCA i tidsdomänen . . . . .	8
2.3 Mjukvara . . . . .	9
2.3.1 MATLAB . . . . .	9
2.3.1.1 Sferiska koordinater i MATLAB . . . . .	9
2.3.1.2 Systemobjekt . . . . .	10
2.3.1.3 Phased Array Systems Toolbox . . . . .	10
2.3.1.4 Sensor array analyzer . . . . .	10
<b>3 Metoder och implementation</b>	<b>11</b>
3.1 Bestämningen av optimal uppsättningsgeometri . . . . .	11
3.1.1 Tillverkningsbegränsningar . . . . .	11
3.1.2 Simulering av uppsättningsgeometrier . . . . .	11
3.1.2.1 Skapa UCA-systemobjekt . . . . .	12
3.1.2.2 Skapa URA-systemobjekt . . . . .	12
3.1.2.3 Skapa UHA-systemobjekt . . . . .	12
3.1.2.4 Skapa styrningsvikter . . . . .	13

3.1.2.5	Uträkning av dem maximala och minsta viktningfria uppmättningsfrekvenserna . . . . .	14
3.1.2.6	Ritning av strålningsdiagram . . . . .	14
3.1.2.7	Uträkningen av lobbredd . . . . .	14
3.1.2.8	Uträkningen av den maximala relativa sidlobsnivån . . . . .	15
3.1.3	Evaluering av geometriernas prestandaparametrar . . . . .	16
3.1.3.1	Val av mikrofonantal . . . . .	16
3.2	Hårdvaruutveckling . . . . .	17
3.2.1	PCB . . . . .	17
3.2.2	UMA-16 v2 USB mic array . . . . .	17
3.2.2.1	Tillhörande mikrofoner . . . . .	18
3.3	Mjukvaruutveckling . . . . .	18
3.3.1	FPGA-lösning . . . . .	18
3.3.2	MATLAB-implementation . . . . .	18
3.3.2.1	Inläsning av ljudkort . . . . .	18
3.3.2.2	Inläsning av webbkamera . . . . .	19
3.3.2.3	Superponering över webbkameravideo . . . . .	19
3.3.2.4	Indexering av ljudkanalerna . . . . .	19
3.3.2.5	Strålskanning i realtid . . . . .	20
3.3.2.6	Efterbehandlad strålskanning . . . . .	22
3.4	Konstruktion . . . . .	24
3.4.1	Webbkameramontering . . . . .	24
3.4.2	Stativ . . . . .	25
3.4.3	Kalibrering . . . . .	26
<b>4</b>	<b>Resultat</b>	<b>27</b>
4.1	Prestandamätningar . . . . .	27
4.2	Resultat från prestandamätningar . . . . .	27
4.2.1	Frekvensomfång . . . . .	27
4.2.2	Dynamikomfång . . . . .	31
4.2.3	Ljudurskiljningsförmåga . . . . .	31
4.3	Vardagliga mätningar . . . . .	33
<b>5</b>	<b>Diskussion och slutsats</b>	<b>34</b>
5.1	Mätresultat . . . . .	34
5.1.1	Frekvensomfång . . . . .	34
5.1.2	Dynamikomfång . . . . .	34
5.1.3	Ljudurskiljningsförmåga . . . . .	34
5.2	Kända buggar . . . . .	35
5.3	Användbarhet . . . . .	35
5.4	Potentiella förbättringar . . . . .	35
5.5	Jämförelse mellan den uträknade ideala uppsättningsgeometrin . . . . .	35
5.6	Etiska aspekter . . . . .	35
5.7	Slutsats . . . . .	36
	<b>Bibliografi</b>	<b>37</b>

<b>A</b>	<b>Simulerade prestandaparametrar</b>	<b>I</b>
<b>B</b>	<b>Simulering av UMA-16 v2 USB</b>	<b>XXII</b>

# Lista över figurer

2.1	Vanliga geometrier på mikrofonuppsättningar som kommersiella akustiska kameror använder sig utav. Från vänster till höger: UCA, URA, UHA.	3
2.2	Exempel på strålningsdiagram som används för att visualisera direktiviteten hos sensoruppsättningar.	5
2.3	DAS-algoritmen demonstrerad i tidsdomänen. Bild hämtad ifrån [18]	7
2.4	Geometrin av UCA bestående av $N$ sensorer med radie $a$ och normal åt $z$ -axeln. Bild hämtad från [12].	8
2.5	Geometrin av UCA bestående av $N$ sensorer med radie $a$ och normal åt $z$ -axeln. Bild hämtad från [15].	9
3.1	Visualisering av prestandaparametrarna HPBW och MSL som användes för att motivera mikrofonuppsättningens geometri.	11
3.2	Frontalbild av UMA-16 v2 USB till vänster och ritning till höger. Bilder hämtade från [14].	17
3.3	Givna prestandaparametrar för mikrofonerna i uppsättningen. Bilder hämtade ifrån [14].	18
3.4	Indexering. Bild hämtad ifrån [14].	20
3.5	Bild på hur kameran monterades fast på mikrofonuppsättningen med hjälp av tejp.	25
3.6	Mikrofonuppsättningens montering till stativet a) och den fullständiga akustiska kameran b).	25
4.1	En bild av ekofria rummet med den akustiska kameran placerad 3 m från en utav studiehögtalarna	27
4.2	Videoutmatning från den akustiska kameran vid utmatning av sinuston av varierande frekvenser från en studiehögtalare placerad 3m ifrån kameran. Frekvensintervallet går från 1 kHz till 12 kHz.	29
4.3	Videoutmatning från den akustiska kameran vid utmatning av sinuston på a) 20 kHz samt b) vitt c) rosa och d) brunt brus från en studiehögtalare vid 70 dB <sub>A</sub> placerad 3m ifrån kameran	30
4.4	Videoutmatning från den akustiska kameran vid utmatning av sinuston på 4 kHz från en studiehögtalare vid a) 30 dB <sub>A</sub> och b) 80 dB <sub>A</sub> placerad 3m ifrån kameran.	31
4.5	Videoutmatning från den akustiska kameran vid utmatning av sinuston med varierande frekvens från studiehögtalare i stereokonfiguration med 175 cm mellanrum placerad 3m ifrån kameran.	32

4.6	Videoutmatning från den akustiska kameran vid utmatning av sinus-ton med varierande frekvens från studiohögtalare i stereokonfiguration med 85 cm mellanrum placerad 3m ifrån kameran. . . . .	32
4.7	Videoutmatning från den akustiska kameran vid utmatning av sinus-ton med varierande frekvens från studiohögtalare i stereokonfiguration med 30 cm mellanrum placerad 3m ifrån kameran. . . . .	33
4.8	Videoutmatning från den akustiska kameran vid mätningar av vardagliga objekt i ett vanligt rum. . . . .	33
A.1	Prestandaparametrarna av UCA med N=8 mikrofoner och radius r=0.175m. Ickestyrd. . . . .	I
A.2	Prestandaparametrarna av UCA med N=8 mikrofoner och radius r=0.175m. Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	II
A.3	Prestandaparametrarna av UCA med N=16 mikrofoner och radius r=0.175m. Ickestyrd. . . . .	III
A.4	Prestandaparametrarna av UCA med N=16 mikrofoner och radius r=0.175m. Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	IV
A.5	Prestandaparametrarna av UCA med N=32 mikrofoner och radius r=0.175m. Ickestyrd. . . . .	V
A.6	Prestandaparametrarna av UCA med N=32 mikrofoner och radius r=0.175m. Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	VI
A.7	Prestandaparametrarna av 3x3 URA med N=9 mikrofoner och elementavstånd $d=0.35/2=0.175m$ . Ickestyrd. . . . .	VII
A.8	Prestandaparametrarna av 3x3 URA med N=9 mikrofoner och elementavstånd $d=0.35/2=0.175m$ . Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	VIII
A.9	Prestandaparametrarna av 4x4 URA med N=16 mikrofoner och elementavstånd $d=0.35/3=0.116m$ . Ickestyrd. . . . .	IX
A.10	Prestandaparametrarna av 4x4 URA med N=16 mikrofoner och elementavstånd $d=0.35/3=0.116m$ . Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	X
A.11	Prestandaparametrarna av 6x6 URA med N=36 mikrofoner och elementavstånd $d=0.35/5=0.07m$ . Ickestyrd. . . . .	XI
A.12	Prestandaparametrarna av 6x6 URA med N=36 mikrofoner och elementavstånd $d=0.35/5=0.07m$ . Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	XII
A.13	Prestandaparametrarna av 2-sidig UHA med N=7 mikrofoner och radius r=0.175m. Ickestyrd. . . . .	XIII
A.14	Prestandaparametrarna av 2-sidig UHA med N=7 mikrofoner och radius r=0.175m. Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	XIV
A.15	Prestandaparametrarna av 3-sidig UHA med N=19 mikrofoner och radius r=0.175m. Ickestyrd. . . . .	XV
A.16	Prestandaparametrarna av 3-sidig UHA med N=19 mikrofoner och radius r=0.175m. Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	XVI
A.17	Prestandaparametrarna av 4-sidig UHA med N=37 mikrofoner och radius r=0.175m. Ickestyrd. . . . .	XVII
A.18	Prestandaparametrarna av 4-sidig UHA med N=37 mikrofoner och radius r=0.175m. Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	XVIII

B.1	Prestandaparametrarna av 4x4 URA med N=16 mikrofoner och elementavstånd r=44 mm samma som UMA16 v2. Ickestyrd. . . . .	XXII
B.2	Prestandaparametrarna av 4x4 URA med N=16 mikrofoner och elementavstånd r=44 mm samma som UMA16 v2. Styrd till $\varphi = 30^\circ$ och $\theta = 0^\circ$ . . . . .	XXIII

# Lista över tabeller

A.1	MSL och HPBW av UCA, URA och UHA ickestyrda och styrda till $\varphi = 30^\circ$ , $\theta = 0^\circ$ med 8, 9 respektive 7 mikrofoner med samma dimensioner som visat i figurerna ovan. . . . .	XIX
A.2	MSL och HPBW av UCA, URA och UHA ickestyrda och styrda till $\varphi = 30^\circ$ , $\theta = 0^\circ$ med 16, 16 respektive 19 mikrofoner med samma dimensioner som visat i figurerna ovan. . . . .	XX
A.3	MSL och HPBW av UCA, URA och UHA ickestyrda och styrda till $\varphi = 30^\circ$ , $\theta = 0^\circ$ med 36, 36 respektive 37 mikrofoner med samma dimensioner som visat i figurerna ovan. . . . .	XXI
B.1	MSL och HPBW av UMA16 v2 ickestyrd och styrd till $\varphi = 30^\circ$ , $\theta = 0^\circ$ .	XXIV

# 1

## Introduktion

En akustisk kamera kombinerar optiska kameratekniker med ljudsignalerna från en mikrofonuppsättning som signalbehandlas för att mäta av ljudtrycksnivåer från olika vinklar. Detta kallas för strålskanning. Därefter superponeras det resulterande färgdiagrammet över videon från den optiska kameran vilket möjliggör snabb och exakt lokalisering av ljudkällor[3].

Genom att analysera ljudmönstren från akustiska kameror kan forskare och ingenjörer studera information av dem akustiska egenskaperna hos ljudkällorna. Denna förmåga är av särskild betydelse för att både bättre förstå och behandla problem som till exempel säkerhet, felsökning av maskineri och trafik bullermätning [2].

Akustiska kameror har traditionellt varit väldigt dyra eftersom det stora antalet mikrofoner som behöver parallellt spela in ljud samtidigt för att sedan genomgå beräkningstung signalbehandling och därefter superponeras över video. Prestandakraven för att uppnå detta i realtid resulterade i väldigt dyr och klumpig hårdvara. Men efter uppkomsten av billigare hårdvara har tröskeln sänkts avsevärt och är nu realistiskt att konstruera en akustisk kamera från enkla komponenter till ett rimligt pris [10].

### 1.1 Syfte och mål

Syftet med projektet är att konstruera en effektiv akustisk kamera från enkla komponenter. Därutöver skall en optimal geometri av mikrofonuppsättningen motiveras med hjälp av simuleringar.

Målet för den akustiska kameran är att möjliggöra följande.

- Realtidslokalisering av ljud med acceptabel latens: Huvudmålet är att utveckla en akustisk kamera som kan detektera och lokalisera ljudkällor i realtid. Detta innebär att systemet kräver adekvat latens.
- Smidig transport av kameran: Funktionalitetskraven inkluderar även att skapa en akustisk kamera som är lätt att transportera. Detta innebär att kameran bör vara portabel, enkel att installera och använda i olika miljöer och situationer.
- Enkla komponenter: Målet med arbetet är att designa en akustisk kamera från enkla komponenter som man enkelt och billigt kan återskapa för eget bruk.

- Säkerställa hög prestanda och tillförlitlighet: Detta innefattar att säkerställa att kameran kan hantera olika ljudfrekvenser och miljöer samtidigt som den bibehåller noggrannheten i ljudlokaliseringen.

### 1.2 Avgränsingar

Arbetet begränsas till en tvådimensionell akustisk kamera som använder sig av enkel fjärrfältlobstyrning. Den optimala geometrin av mikrofonuppsättningen motiverades genom simulationer och tillverkningsbegränsingar men realiserades aldrig. Istället konstruerades den akustiska kameran från en kommersiell mikrofonuppsättning.

# 2

## Teori och teknisk bakgrund

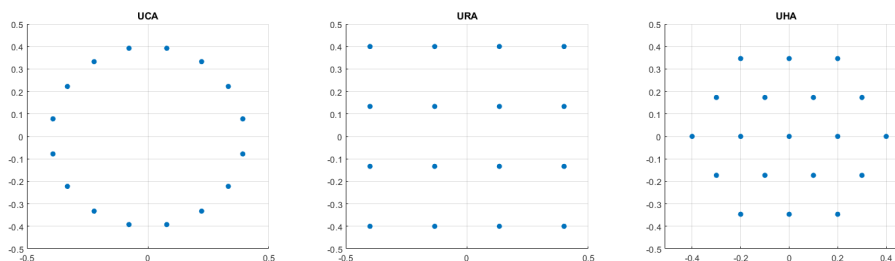
I detta avsnitt förklaras all nödvändig signalbehandlingsteori som mjukvaran och simuleringen grundar sig i samt alla hårdvarukomponenter och deras respektive uppgifter

### 2.1 Mikrofonuppsättning

En mikrofonuppsättning består av ett nätverk av mikrofoner som är distribuerade för att samla in ljudsignaler från olika riktningar[6].

Under de senaste tiden har mikrofonuppsättningar fått ökad uppmärksamhet och utvecklats för att möjliggöra separation av avancerad ljudbehandling[2]. Den ökande tillgängligheten och minskade kostnaden för mikrofonuppsättningar har också öppnat upp nya möjligheter inom områden som känsliga applikationer.

Mikrofonuppsättningen hos akustiska kameror är avgörande för deras prestanda och användbarhet. Genom att välja rätt geometri optimeras riktningen och upplösningen av ljudinspelningen. Vanliga geometrier inkluderar i figur 2.1, var och en med sina fördelar beroende på tillämpningen. Plan geometri erbjuder enkelhet och precision vid ljudfokusering, medan sfäriska geometrier passar bra för att fånga ljud från alla riktningar. Cylindriska former är fördelaktiga för att mäta ljud längs en linje eller i ett specifikt område. Genom att förstå och välja lämplig geometri kan akustiska kameror effektivt användas för olika ändamål, från ljudlokalisering och bildbehandling till rumskaraktisering och kvalitetskontroll.



**Figur 2.1:** Vanliga geometrier på mikrofonuppsättningar som kommersiella akustiska kameror använder sig utav. Från vänster till höger: UCA, URA, UHA.

De olika uppsättningarna, vilket kommer att tydliggöras senare i rapporten, visar sig vara mer lämpliga för olika ändamål, såsom olika mätningssmiljöer och frekvensom-

råden. Mer avancerade och komplexa geometrier är betydligt svårare att konstruera, och dessutom kräver de mer avancerade programmerings- och simuleringsmetoder.

## 2.2 Signalteori

I detta avsnitt förklaras signalteori som mjukvaran behandlar för att kunna räkna ut ljudstyrkan i särskilda riktningar

### 2.2.1 Direktivitet

En uppsättning av mikrofoner som alla tar in ljudsignaler samtidigt och summeras ihop kan ha destruktiv eller konstruktiv interferens beroende på var ljudkällan kommer ifrån [8]. Mikrofonuppsättningens *direktivitet* åt en viss riktning är måttet av dess känslighet till ljud som radierar från den riktningen och ges av

$$D = 4\pi \frac{U_{rad}(\omega, \Theta, \varphi)}{P_{total}} [dBi] \quad (2.1)$$

där  $U_{rad}(\Theta, \varphi)$  är strålningsintensiteten av ljudkällan i riktningen  $(\Theta, \varphi)$  och  $P_{total}$  är den totala effekten avgett från ljudkällan [13].

För att simulera direktiviteten av en mikrofonuppsättning antas följande [3]

- Alla mikrofoner är identiska
- Alla mikrofoner är isotropa
- Alla inkommande ljudvågor kan betraktas som planvågor

Dem inkommande ljudvågorna antas, i och med avgränsningen, vara planvågor som kan representeras i formen

$$e^{j\mathbf{k}^T \mathbf{r}} \quad (2.2)$$

där  $\mathbf{k}$  är den tredimensionella vågtalet som beskriver vågberedningen

$$\mathbf{k} = \frac{\omega}{c} \begin{bmatrix} \sin \Theta \cos \varphi \\ \sin \Theta \sin \varphi \\ \cos \Theta \end{bmatrix} \quad (2.3)$$

och  $\mathbf{r}$  är dem rektangulära koordinaterna där vågen observeras

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.4)$$

Det bildade interferensmönstret som fås av att addera ihop uppsättningens alla utsignaler multiplicerat med mikrofonspecifika koefficienter,  $a_n$ , kallas för dess *receiving array factor*, RAF i kortform, vilket för en uppsättning av  $N$  stycken isotropiska mikrofoner placerade vid  $\vec{r}_n$  ges av [4]

$$RAF(\hat{r}) = \sum_{n=1}^N a_n e^{-jk\hat{r} \cdot \vec{r}_n} \quad (2.5)$$

## 2. Teori och teknisk bakgrund

Detta arbetet behandlar sfäriska koordinater och utan någon styrning sätts  $a_n$  uniformt till  $1/N$  vilket, uttryckt i matrisformen i ekvation 2.2, ger den generiska RAF:n

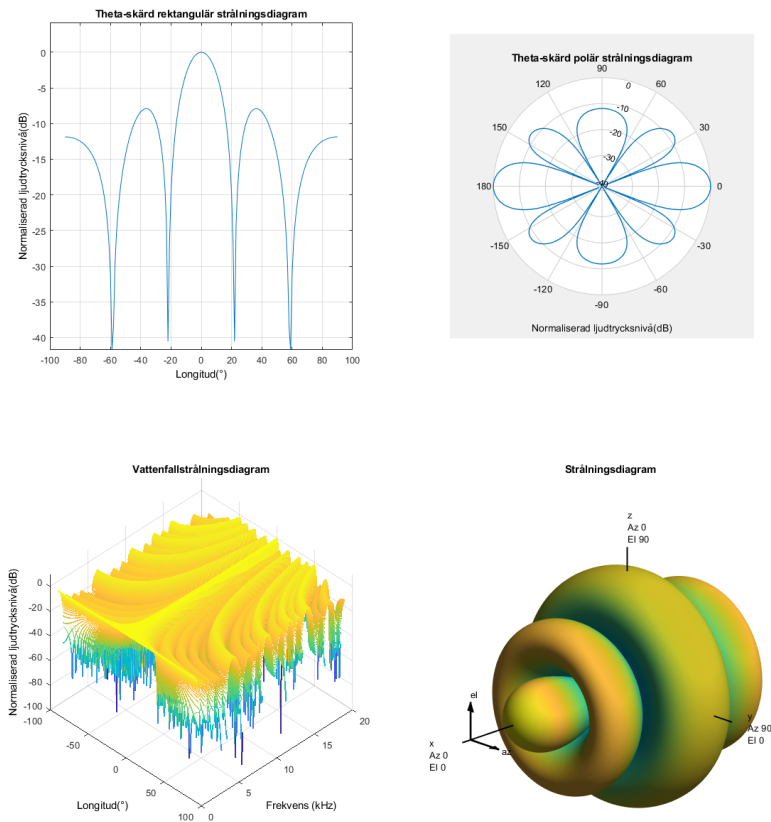
$$RAF(\omega, \theta, \varphi) = \frac{1}{N} \sum_{n=1}^N e^{-jk^T \cdot \mathbf{r}_n} \quad (2.6)$$

Enligt [4] är en antennauppsättnings array factor, AF i kortform, lika med komplexkonjugatet av dess RAF vilket gör simuleringen av båda likgiltiga. Strålningsintensiteten från ekvation 2.1 är en funktion av AF

$$U_{rad}(\omega, \theta, \varphi) = |AF(\omega, \theta, \varphi)|^2 [dB] \quad (2.7)$$

För en emitterande sensoruppsättning kan  $U_{rad}$  betraktas som förmågan att rikta strålning åt ett visst håll medans en mottagande sensoruppsättning kan betrakta det som känsligheten för strålningskällor från en viss riktning.

### 2.2.1.1 Strålningsdiagram



**Figur 2.2:** Exempel på strålningsdiagram som används för att visualisera direktiviteten hos sensoruppsättningar.

*Strålningsdiagram* används för att visualisera egenskaperna och beteenden hos sensoruppsättningar genom att rita ut direktiviteten, strålningsintensiteten eller AF:n som en funktion av riktning eller frekvens [8]. Det finns flera sätt att visualisera dessa men den vanligaste för 2D-uppsättningar är att skära vid en bestämd polvinkel (ofta  $\theta = 0$ ) samt frekvens och representera strålningsintensiteten som en funktion av longitud. Direktiviten tar i det här fallet formen av lober där den i mätriktningen man är intresserad av kallas för huvudloben och alla andra, förhoppningsvis mindre, kallas för sidloben. Dessa kan sedan ritas i både rektangulärt och polärt format och ofta normaliserat så att huvudloben antar värdet 0 dB. Ett annat sätt är att rita ut strålningsintensiteten som en funktion av både polär- och longitudvinkeln i 3D. Det sista vanliga sättet är att skära vid en bestämd polvinkel och rita ut strålningsintensiteten som en funktion av longitud och frekvens i form av ett så kallat vattenfallplot.

### 2.2.1.2 Lobbredd

*Lobbredd*, även känd som *half power beam width* eller *HPBW* i kortform på engelska, är vinkeln mellan huvudlobens två halveringspunkter [9]. För ett normaliserat strålningsdiagram är halveringspunkterna -3dB och kallas därför ibland också för *3 dB beamwidth* på engelska [3]. Ju snävare lobbredd desto bättre direktivitet och därmed lokaliseringsförmåga har mikrofonuppsättningen.

### 2.2.1.3 Maximala relativa sidlobsnivån

Den maximala relativa sidlobsnivån, även kallad *maximum relative side lobe level* på engelska eller *MSL* i kortform, är måttet på hur stor den kraftigaste sidloben är i förhållande till huvudloben [10]. Dessa är av särskild intresse då vinklar med kraftiga sidloben plockas upp av mikrofonuppsättningen beroende kan resultera i kraftig missvisande resultat.

### 2.2.1.4 Vikning

Som sett från ekvation 2.7 är direktiviteten beroende av ljudfrekvensen som sett från k-termen och mikrofonernas placering som sett från r-termen. Detta beroende kan introducera vad som kallas för *spatial aliasing* eller *vikning* på svenska [10]. Otillräckligt avstånd mellan mikrofonerna leder till låga frekvenser inte plockas upp som hög och låg av dem olika mikrofonerna samtidigt vilket leder till att summan av ljudsignalerna ser väldigt likt ut från alla infallsvinklar. För långt avstånd mellan mikrofonerna leder till att högfrekventa ljudsignaler fasförskjuts en hel våglängd för vissa mikrofoner från vissa infallsvinklar vilket leder till kraftiga sidloben vid just dem vinklarna. För att motverka båda av dessa krävs stora uppsättningar med stort maximalt avstånd mellan mikrofonerna för att kunna läsa av höga och låga instanser av lågfrekventa ljudsignaler samtidigt. Likaväl så krävs tillräckligt många mikrofoner för att säkerställa att ointressanta infallsvinklar som fasförskjuter ljudsignalerna från vissa mikrofoner med en hel våglängd tas ut destruktivt och på så sätt inte ger upphov till kraftiga sidloben.

För att undvika vikning vid uppmätning av högfrekvent ljud krävs det att ele-

mentavståndet,  $d_{element}$ , är mindre än

$$d_{element} < \frac{c}{2\lambda_{min}} \quad (2.8)$$

och för att korrekt mäta upp lågfrekvent ljud bör maxavståndet mellan två mikrofoner,  $d_{max}$ , vara större än

$$d_{max} > \frac{c}{\lambda_{max}} \quad (2.9)$$

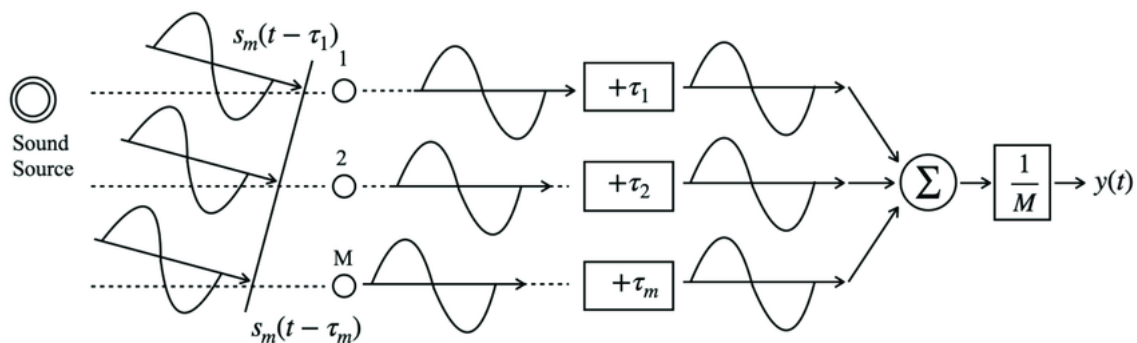
[10]. Variabeln  $c$  är i det här fallet ljudets hastighet i luft och  $\lambda$  ljudvåglängd.

## 2.2.2 Lobstyrning

Lobstyrning är ett signalbehandlingsknep som används för att skicka och ta emot signaler i särskilda riktningar. Den utgår på att, som namnet antyder, styra loberna hos sensoruppsättningar[8]. Huvudloben styrs åt en viss riktning och sidloberna antingen dämpas i alla riktningar eller också selektivt stärks dem i andra önskevärda riktningar man vill skicka eller ta emot signaler.

### 2.2.2.1 Delay and Sum

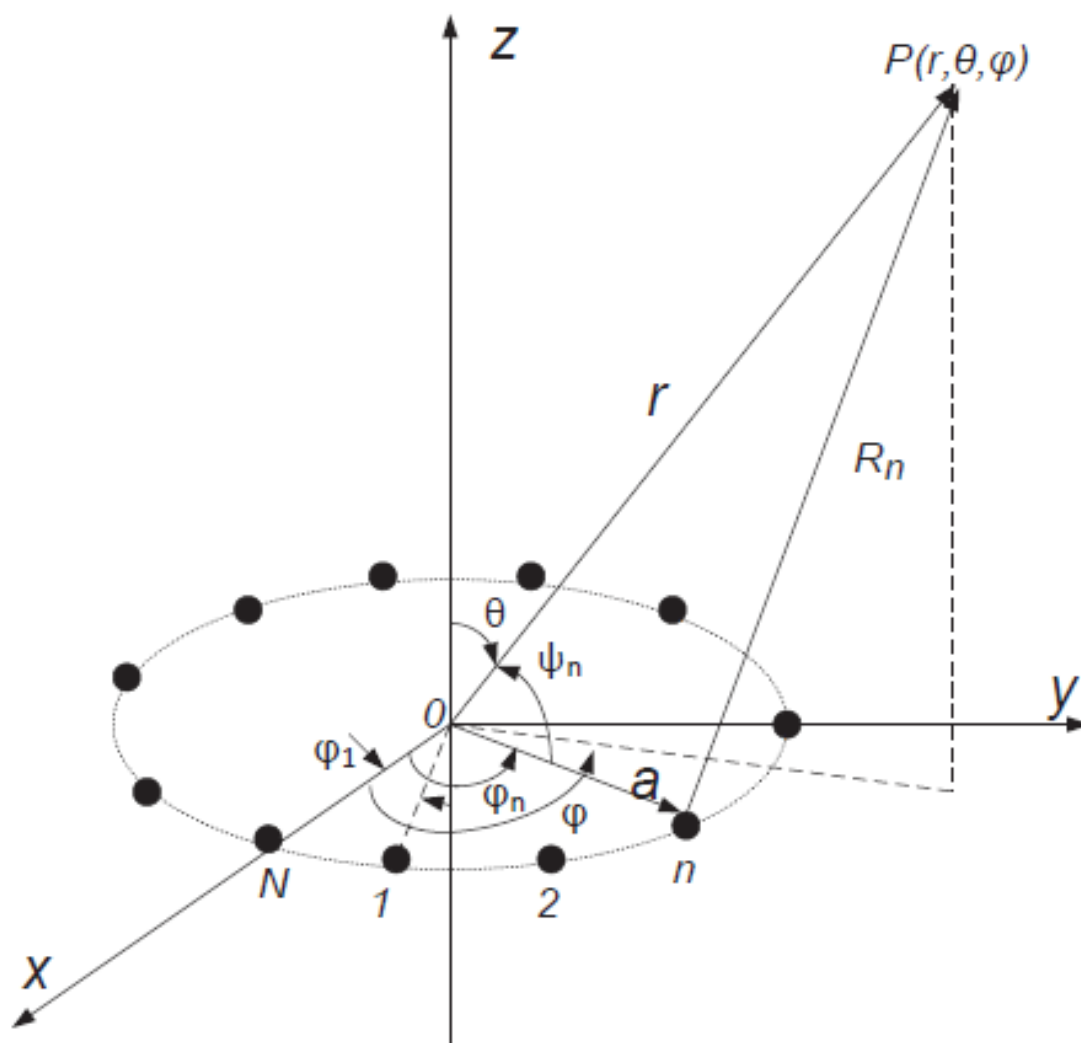
En form av lobstyrning är den så kallade "Delay and sum"-algoritmen, DAS i kortform, och fungerar genom att räkna på tidskillnaden som det tar för ljud att nå dem enskilda mikrofonerna och den destruktiva interferensen som summan av dessa ljudsignaler har på varandra [1]. Riktningstyrkan fås genom att variera tidsfördrjningen av ljudsignalerna innan summeringen av dem och dividera med antalet ljudsignaler (se figur 2.3 nedan). Bidraget till ljudstyrkan vid mätriktningen försvagas då i alla riktningar förutom mätriktningen optimalt med flera decibel.



**Figur 2.3:** DAS-algoritmen demonstrerad i tidsdomänen. Bild hämtad ifrån [18]

DAS-algoritmen är den mest välkända och enklaste algoritmen att implementera och presterar bra i dem flesta situationerna. Den låga komplexiteten gör den även passande för realtidssystem som till exempel akustiska kameror[1].

## 2.2.2.2 Fjärrfältslobstyrning av UCA i tidsdomänen



**Figur 2.4:** Geometrin av UCA bestående av  $N$  sensorer med radie  $a$  och normal åt  $z$ -axeln. Bild hämtad från [12].

Mikrofonernas positioner i  $xy$ -planet definieras av deras longitudvinkel

$$\phi_n = 2\pi \left( \frac{n}{N} \right), n = 1, 2, \dots, N \quad (2.10)$$

Avståndet mellan ljudkällan  $P(r, \theta, \varphi)$  och  $n$ :te mikrofonen  $R_n$  ges av

$$R_n = (r^2 + a^2 - 2ar \cos \psi_n)^{1/2} \quad (2.11)$$

vilket med antagelsen att ljudkällan är väldigt långt bort kan förenklas till

$$R_n \approx r - a \cos \psi_n = r - a \sin \theta \cos(\phi - \phi_n) \quad (2.12)$$

Tidsfördröjningen av varje mikrofon för att styra huvudloben åt  $(\theta_0, \phi_0)$  ges till slut av [12]

$$\tau_n(\theta_0, \phi_0) = \frac{r - R_n}{c} = (a/c) \sin \theta_0 \cos(\phi_0 - \phi_n) \quad (2.13)$$

där  $c$  är i detta fallet ljudets hastighet. Detta ger UCA-upsättningen den  $\theta$ -skärda strålningsintensiten [8].

$$|AF(\omega, \varphi, \phi_0)| = \left| \frac{1}{N} \sum_{n=1}^N e^{j \frac{\omega}{c} a (\cos(\phi_0 - \phi_n) - \cos(\varphi - \phi_n))} \right|^2 \quad (2.14)$$

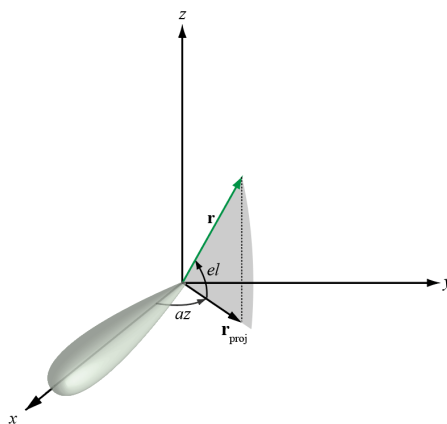
## 2.3 Mjukvara

Det här avsnittet går igenom lite bakgrundinformation inför mjukvaruutvecklingssektionen senare in i metodavsnittet.

### 2.3.1 MATLAB

MATLAB, en förkortning av *Matrix Laboratory*, syftar på både själva datorprogrammet och dess inbyggda programmeringsspråk. Det är gjort för att lätt och snabbt kunna programmera matematiska och tekniska beräkningar samt sömlöst visualisera dessa. Man åstadkommer detta via enkel programmeringssyntax och viktigast av allt deras inbyggda verktyg och funktioner som de kallar för toolboxer vilket innebär att man sällan behöver implementera algoritmer, visualiseringsverktyg och så vidare i kod själv. Fördelen av att vara snabb och enkel att programmera har dock nackdelen av att inte vara den snabbaste lösningen. Därför används MATLAB ofta flitigt av ingenjörer och forskare för prototypframtagning och simulering innan man går över till själva implementeringen av vad man nu vill utveckla [19].

#### 2.3.1.1 Sfäriska koordinater i MATLAB



**Figur 2.5:** Geometrin av UCA bestående av  $N$  sensorer med radie  $a$  och normal åt  $z$ -axeln. Bild hämtad från [15].

I MATLAB definieras latitud- och longitudvinkeln i förhållande till x-axeln. Azimuth,  $az$  i kortform, motsvarar longitudvinkeln och betecknas med  $\varphi$  och elavation,  $el$  i kortform, motsvarar polvinkeln och betecknas med  $\theta$ [15]. Av denna anledning placeras dem simulerade uppsättningarnas sensorer inuti yz-planet.

### 2.3.1.2 Systemobjekt

Något som utmärker MATLAB för simulering och prototypframtagning är deras inbyggda så kallade *systemobjekt*. De är specialiserade programmeringsobjekt som man antingen kan kalla metoder på eller ta som argument i andra systemobjekt och funktioner. Man kan till exempel skapa ett UCA-systemobjekt för att tas in som argument i funktioner för att få ut strålningsdiagram eller lobbredd. Alternativt kan systemobjektet användas som argument i skapandet av ett nytt lobstyrningssystemobjekt som lobstyr ljudsignalerna från en mikrofonuppsättning enligt geometrin specificerat i UCA-systemobjektet. En annan fördel av många systemobjekt är kodgeneration i andra språk som C/C++ eller till och med HDL för implementering i FPGA[16].

### 2.3.1.3 Phased Array Systems Toolbox

Phased Array Systems toolboxen möjliggör enkel simulering och analysering av sensoruppsättningar men också lobstyrning. Den möjliggör i kontexten av det här arbetet lobstyrning i tidsdomänen som plockas upp av en mikrofonuppsättning med viss geometri som sedan strålskannas och superponeras som färdigram över webbkameravideo[17].

### 2.3.1.4 Sensor array analyzer

Sensor array analyzer är en inbyggd app som använder sig utav "Control System"-, "DSP System"-, "Phased Array System"- och "Signal Processing"-toolboxarna. Appen låter dig analysera en rad av olika sensoruppsättningar genom att enkelt variera geometrin av uppsättningen och deras tillhörande parametrar samt signalhastighet och frekvens via dess grafiska användargränssnitt. Genom att sätta dessa parametrar beräknar appen automatiskt ut uppsättningens egenskaper som till exempel direktivitet, AF som kan visualieras i 3D samt 2D i polära såväl som rektangulära koordinater. Appen låter dig även exportera all simulation till MATLAB-kod med hjälp av deras "Generate MATLAB-script"-funktionalitet. Detta är väldigt smidigt för att generera uppsättningar utan någon dedikerad systemobjekt som till exempel UHA [17].

# 3

## Metoder och implementation

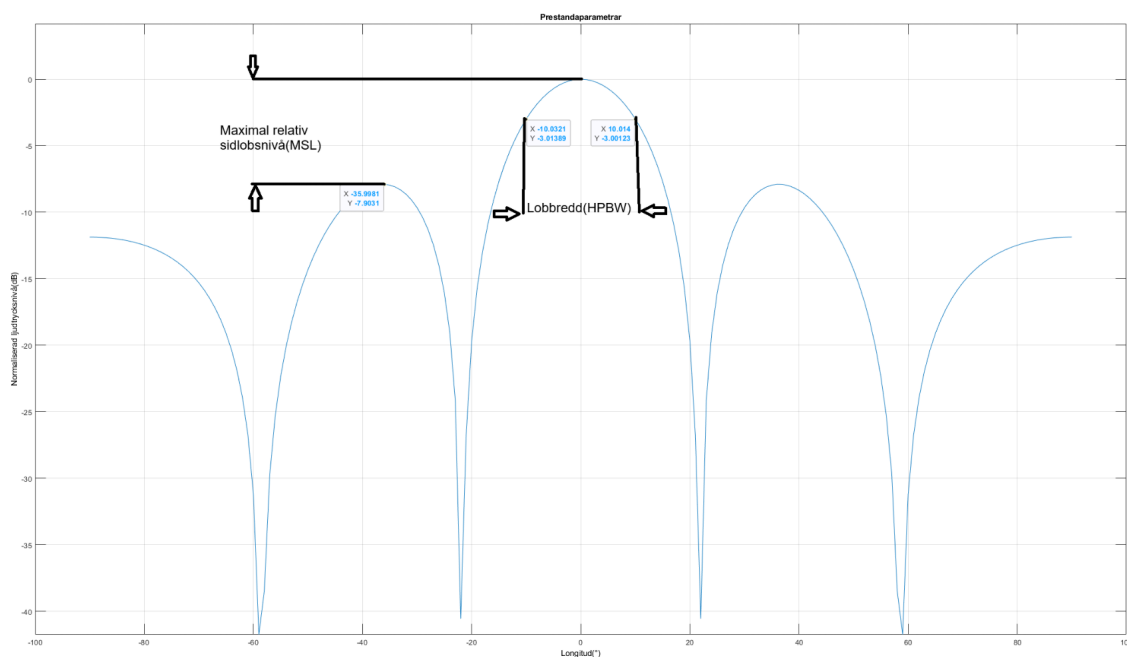
### 3.1 Bestämningen av optimal uppsättningsgeometri

Den optimala geometrin på mikrofonuppsättningen motiverades från sammanvägningar av resultat från simuleringar, tillverknings- och hårdvarubegränsningar samt kostnadsberäkningar.

#### 3.1.1 Tillverkningsbegränsningar

Dem flesta ansedda tillverkare begränsar diametern av sina icke-specialbeställda PCB:er till 40 cm [10]. Med rum för marginal begränsades därför diametern av mikrofonuppsättningarna till 35 cm. Eftersom 35 cm redan resulterar i väldigt hög lägsta vinkningsfri uppmättningsfrekvens som sett från ekvation 2.8 och 2.9. Därför valdes diametern till den maximala begränsningen utan vidare experimenterande.

#### 3.1.2 Simulering av uppsättningsgeometrier



**Figur 3.1:** Visualisering av prestandaparametrarna HPBW och MSL som användes för att motivera mikrofonuppsättningens geometri.

Framtagning av den optimala geometrin togs fram genom att först experimentera med 2D-uppsättningarna som medföljs i phased array systems nämligen uniform rectangular(URA), circular(UCA) och hexagonal(UHA) array.

Prestandaparametrarna som togs i åtanke för motiveringen av geometrin var

- Lobbredd(HPBW)
- Den maximala relativa sidlobsnivån(MSL)

Dem maximala och minsta vinkningsfria uppmättningsfrekvenserna togs även i åtanke för att få ett hum om uppsättningarnas frekvensomfång.

#### 3.1.2.1 Skapa UCA-systemobjekt

URA-uppsättningen simulerades med hjälp av det inbyggda UCA-systemobjektet som definieras av antalet sensorer, N, och radien, r. Som tidigare nämnt i avsnitt 2.3.1.1 så placeras sensorerna i yz-planet genom att ansätta 'ArrayNormal' till 'x'. Detta görs också för dem två resterande uppsättningarna.

```
function [UCA] = createUCA(N, r)
UCA = phased.UCA('NumElements',N,...
    'ArrayNormal','x','Taper',1);
UCA.Radius = r;
end
```

#### 3.1.2.2 Skapa URA-systemobjekt

URA-uppsättningen simulerades med hjälp av det inbyggda URA-systemobjektet som definieras av elementärvståndet, diametern och antalet rader respektive kolonner av sensorer.

```
function [Array] = createURA(row, col, diameter)
Array = phased.URA('Size',[row col],...
    'Lattice','Rectangular','ArrayNormal','x');
Array.ElementSpacing = [diameter/(row-1) diameter/(col-1)
    ];
end
```

#### 3.1.2.3 Skapa UHA-systemobjekt

Eftersom UHA inte har ett eget systemobjekt användes istället ett ConformalArray-systemobjekt som låter en modellera uppsättningar av valfri geometri. Matlabkoden genererades med hjälp av "Sensor Array Analyzer"-appens "Generate MATLAB-script"-funktionalitet.

```
function [Array] = createUHA(N)
switch N
case 2
    Nside = 2;
    delta = 0.175;
```

```

    case 3
        Nside = 3;
        delta = 0.0875;
    case 4
        Nside = 4;
        delta = 0.05833333333333333;
    otherwise
        warning('Please pick 2,3 or 4 sides')
end
rows = [1:Nside Nside-1:-1:1];
Radius = delta * (Nside - 1);
pos = zeros(3,1);
count = 0;
for idx = 1:length(rows)
    y = -Radius/2 - (rows(idx)-1)*delta*0.5 : delta : ...
        Radius/2 + (rows(idx)-1)*delta*0.5;
    pos(2, count+1:count+length(y)) = y;
    pos(3, count+1:count+length(y)) = sqrt(3)/2*Radius -
        ...
        (idx-1)*delta*sind(60);
    count = count+length(y);
end
Array = phased.ConformalArray('ElementPosition', pos, ...
    'ElementNormal', zeros(2, size(pos,2)), 'Taper', 1)
;
end

```

#### 3.1.2.4 Skapa styrningsvikter

För att kunna lobstyra krävs det styrningsvikter vilket i sin tur generas med hjälp av SteeringVector-systemobjektet. Matlabkoden genererades också med hjälp av "Sensor Array Analyzer"-appens "Generate MATLAB-script"-funktionalitet.

```

function [w] = createWeights(Array, frequencies,
    PropagationSpeed, SteeringAngles)
w = zeros(getNumElements(Array), length(frequencies));
for idx = 1:length(frequencies)
    SteerVector = phased.SteeringVector('SensorArray',
        Array, ...
        'PropagationSpeed', PropagationSpeed);
    w(:, idx) = step(SteerVector, frequencies(idx),
        SteeringAngles);
end
end

```

### 3.1.2.5 Uträkning av dem maximala och minsta vinkningsfria uppmättningsfrekvenserna

I uträkning av dem maximala och minsta vinkningsfria uppmättningsfrekvenserna nyttjades Muellers handskrivna matlabfunktion[10]. Den fungerar genom att först räkna ut alla möjliga Euklidiska avstånd med hjälp av pdist-funktionen som finns inbyggd i MATLABs "Statistics and Machine Learning"-toolbox. Därefter väljes det minsta respektive största avståndet bland dessa som senare senare in i ekvation 2.8 och 2.9 för att få ut fmax och fmin.

```
function [fmin,fmax] = CalcFrequencies(points)
% calculate fmin and fmax for input points .
c = 343.2; % speed of sound

distances = pdist(points); % calculate all distances in
    point vector
dmin = min(distances); % smallest distance in array
dmax = max(distances); % largest distance in array

fmax = c/(2*dmin); % fmax follows shannon / nyquist -> at
    least two sampling points per wavelenght
fmin = c/dmax; % the wavelenght of fmin ist the largest
    distance in an array
end
```

### 3.1.2.6 Ritning av strålningsdiagram

Ritningen sker via MATLABs "phased array systems"-toolbox inbyggda "pattern"-funktion[17]. För att skära vid någon vinkel anges någon av variablerna *AZ* eller *EL* till en skalär. För att rita ut rektangulärt anges variabeln *coordinateSystem* till 'rectangular' och för polärt anges den till 'polar'. Variabeln *plotType* kan varieras för att rita ut strålningsintensitet angett normalt eller angett i decibel alternativt direktivet angett i dBi(se ekvation 2.1) genom att ändra plotType till 'PowerDB', 'Power' eller 'Directivity' respektivt. Variabeln *plotStyle* anges normalt till 'Overlay' men kan ändras till 'Waterfall' ifall man vill rita ut vattenfallstrålningsdiagram förutsatt att *coordinateSystem* sätts till 'rectangular' och frequencies är ett spann av frekvenser.

```
pattern(Array, frequencies, AZ, EL, 'PropagationSpeed',
    PropagationSpeed, 'CoordinateSystem', coordinateSystem
    , 'weights', w, 'Normalize', true, 'Type', plotType, '
    PlotStyle', plotStyle);
```

### 3.1.2.7 Uträkningen av lobbredd

HPBW-uträkningen skeddes genom MATLABs inbyggda "beamwidth"-funktion som finns i deras antenna-toolbox. Den tar in ett phased.array-systemobjekt och tillhörande viktvektor samt frekvens och räknar ut HPBW:n för den inmatade frekvensen.

```
function [bw] = calcBW(array, frequency, wts)
[bw, angles] = beamwidth(array, frequency, 'dbDown',3, '
    PropagationSpeed', 340, 'Weights', wts);
end
```

### 3.1.2.8 Uträkningen av den maximala relativa sidlobsnivån

Den maximala relativa sidlobsnivån kan fås av att studera strålningsdiagrammen men för en noggrannare och mer metodisk uträkning av den användes Muellers listade handskrivna matlabfunktion "*Calculate the Beam- and Frequency-Pattern*" [10].

```
function [MSL] = calcMSL (pattern)
% Calculate the MSL for each frequency of input pattern
% find positions and values of all maxima in pattern
[maxima,maxpos] = max(pattern,[],'omitnan');
% if maxima = NaN ( empty ), the position is also NaN
for f = 1: size ( pattern ,2)
    if isnan ( maxima ( f ) )
        maxpos ( f ) = NaN ;
    end
end
% find frequencys which are not NaN
frequencys = find (~ isnan ( maxpos ) ) ;
% number of frequencys for loop
Nfreq = size ( frequencys ,2) ;
% create MSL and fill it with NaN
MSL = NaN (1 , max ( frequencys ) ) ;
% for all frequencys
for f = 1: Nfreq
    % find peaks in beampattern for current f
    pks = findpeaks ( pattern (: , f ) ) ;
    % sort peaks descending
    pks = sort ( pks , 'descend' ) ;
    % if there is more than one peak in pattern
    if size ( pks ,1) > 1
        % MSL equals highest peak ( main lobe ) - second
        highest peak
        MSL ( frequencys ( f ) ) = pks (1) - pks (2) ;
    % else ( only one or no peak )
    else
        % MSL is empty
        MSL ( frequencys ( f ) ) = NaN ;
    end
end
end
end
```

### 3.1.3 Evaluering av geometriernas prestandaparametrar

URA erbjuder bäst HPBW som sett från bilaga A i tabellerna A.1 till A.3 men extremt dåligt mätbart frekvensband som sett från MSL-grafen i bilaga A figurerna A.7 till A.12. UHA erbjuder däremot i särklass bäst MSL och bredast frekvensband som sett från MSL-graferna i bilaga A figurerna A.13 till A.17 men sämst HPBW som sett från bilaga A tabellerna A.1 till A.3. UCA är unik i avseendet att den erbjuder sämst MSL vid dess optimala frekvensband men upplever inte någon tydlig brytpunkt där MSL går ändå till noll som sett från MSL-graferna i bilaga A figurerna A.1 till A.6. Istället erhåller den märkbar mättningsförmåga i form av några få dB MSL ända upp till toppskiktet av det hörbara frekvensbandet. Bland HPBW landar UCA dessutom mellan URA, som är bäst, och UHA, som är sämst som sett från bilaga A tabellerna A.1 till A.3.

Utifrån simuleringarna konstaterades det därför att URA med dess överlägsna HPBW och största avstånd mellan två enskilda mikrofoner är bäst för precis lokalisering av lågfrekvent ljud. UHA med dess breda men oförlåtande frekvensband är bäst vid kontrollerade miljöer om man håller sig inom frekvensbandet. Dess överlägsna MSL med flera dB gör den även optimal vid uppmätningen av flera ljudkällor samtidigt. UCA är den mest mångsidiga utav dem som erbjuder god HPBW på bekostnad av sämre MSL men med väldigt förlåtande frekvensband vilket gör det bästa alternativet för det mesta utanför nischfall.

Eftersom mångsidigheten utan några direkt större nackdelar ansågs önskevård konstaterades därför UCA att vara den optimala geometrin på mikrofonuppsättningen.

#### 3.1.3.1 Val av mikrofonantal

UCA simulerades med 8, 16 respektive 36 mikrofoner. HPBW:n såg ingen förändring vid förändring av mikrofonantal som sett från bilaga A tabellerna A.1 till A.3. Däremot observerades en stor förbättring av frekvensband från drygt 1000 Hz - 3000 Hz ickestyrd och 1000 Hz - 2000 Hz styrd till 1000 Hz - 5000 Hz ickestyrd och 1000 Hz - 3000 Hz styrd vid fördubbling av mikrofonerna från 8 till 16 som sett från MSL-graferna i bilaga A figurerna A.1 till A.6. Därutöver ökades det mindre optimala frekvensområdet med en MSL över 3 dB med hela 8000 Hz till som sett från bilaga A tabellerna A.1 och A.2 och MSL-grafen i bilaga A figur A.3.

Vid upptrappningen från 16 till 36 mikrofoner observerades en MSL på runt 8 dB ända upp till toppskiktet av det hörbara frekvensbandet ickestyrd som trappas ned till det en mindre optimal MSL på 5 dB efter styrning som sett från bilaga A figurerna A.5 och A.6. Eftersom frekvensbandet av UCA:n med endast 8 mikrofoner ansågs för begränsat, speciellt efter styrning, och en mycket stor andel av ljud faller inom frekvensbandet för UCA:n med 16 mikrofoner, styrning medräknat, nöjdes mikrofonantalet till 16.

Valet av 16 istället för 32 drar inte bara ned på kostnaden av den akustiska kameran men även komplexiteten av eventuell PCB samt resulterar i lägre latens.

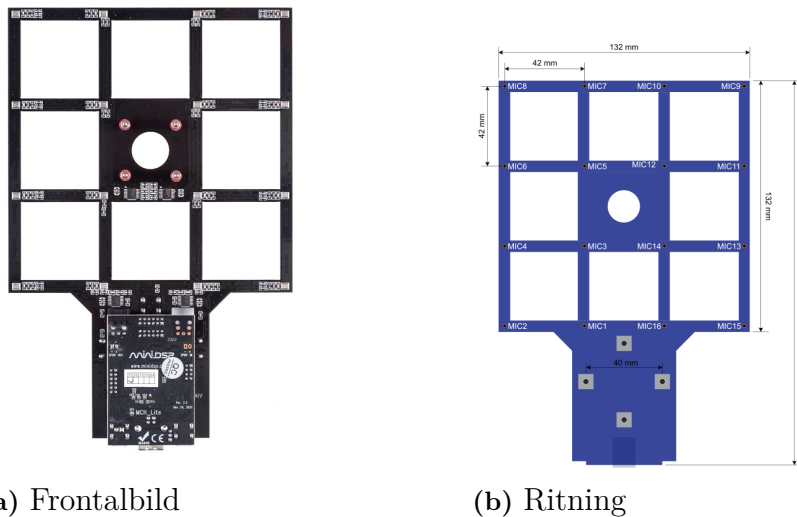
## 3.2 Hårdvaruutveckling

I denna sektion utforskas hårdvaruutvecklingen. Genom att granska den tekniska utvecklingen av produkten ges en djupare inblick i skapandet av dess struktur.

### 3.2.1 PCB

Ritningen av PCB:en påbörjades men hamnade utanför projektets tidsram och behövde därför överges för en kommersiell lösning. Detta innebar också att den simulerade optimala uppsättningsgeometrin givet dagens tillverkningsbegränsningar inte kunde testas i praktiken.

### 3.2.2 UMA-16 v2 USB mic array



**Figur 3.2:** Frontalbild av UMA-16 v2 USB till vänster och ritning till höger. Bilder hämtade från [14].

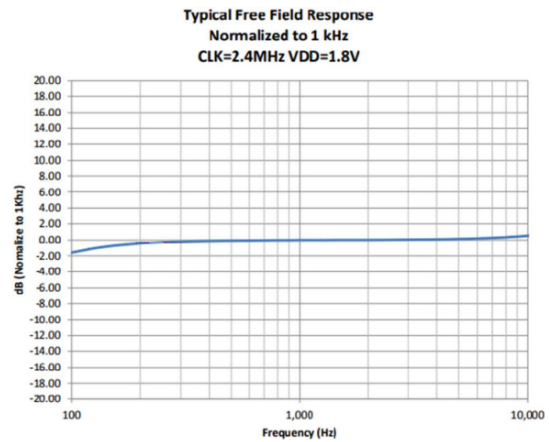
För den kommersiella lösningen valdes "UMA-16 v2 USB mic array"-plattformen. Det är en 4x4 URA-mikrofonuppsättning med ett elementavstånd på 42 mm (se figur 3.2b). Den kopplas via USB-B till USB-A och matar ut 16-ljudkanaler, en för varje mikrofon, med 24-bitars bitdjup och upp till 48 kHz samplingsfrekvens direkt in i datorn via ASIO [14].

### 3.2.2.1 Tillhörande mikrofoner

TEST CONDITIONS:  $f_{\text{clock}} = 2.4 \text{ MHz}$ ,  $V_{\text{DD}} = 1.8 \text{ V}$ , unless otherwise indicated

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Supply Current <sup>1,2,3</sup>	$I_{\text{DD}}$		-	626	700	$\mu\text{A}$
Sensitivity <sup>1</sup>	S	94 dB SPL @ 1 kHz	-30	-29	-28	dBFS
Signal to Noise Ratio	SNR	94 dB SPL @ 1 kHz, A-weighted, $f_{\text{clock}} = 3.072 \text{ MHz}$	-	65.5	-	dB(A)
Total Harmonic Distortion	THD	94 dB SPL @ 1 kHz	-	0.1	-	%
		120 dB SPL @ 1 kHz	-	1.6	-	
Acoustic Overload Point	AOP	10% THD @ 1 kHz	-	122	-	dB SPL
Power Supply Rejection Ratio	PSRR	200 mVpp sinewave @ 1 kHz	-	64	-	dBV/FS
Power Supply Rejection	PSR+N	100 mVpp 1/8 duty cycle rectangular waveform @ 217 Hz, A-weighted	-	-91	-	dBFS(A)

(a) Frontalbild



(b) Ritning

**Figur 3.3:** Givna prestandaparametrar för mikrofonerna i uppsättningen. Bilder hämtade ifrån [14]

Uppsättningen är utrustad med mikrofonerna Knowles SPH1668LM4H med de angivna prestandaparameterna i figur 3.3 [14]. Sammanfattningsvis får man fram att dess sensitivitet ligger typiskt mellan 94 dB<sub>SPL</sub> och 123 dB<sub>SPL</sub> samt att dess självbrus är 29 dB<sub>A</sub>.

## 3.3 Mjukvaruutveckling

Denna sektion går igenom mjukvaruutvecklingen av den påbörjade FPGA-lösningen samt den fastställda MATLAB-implementationen.

### 3.3.1 FPGA-lösning

Mjukvaran hade först tänkt implementeras i FPGA men eftersom PCB:n aldrig hann bli färdig övergavs den, som tidigare sagt, för UMA-16 v2 implementerad i MATLAB.

### 3.3.2 MATLAB-implementation

Denna sektion går igenom alla beståndsdelar av MATLAB-implementation med en snabb genomgång om hur koden fungerar.

#### 3.3.2.1 Inläsning av ljudkort

Ljudkortet är listat som 'miniDSP ASIO Driver' och läses in via audioDeviceReader-systemobjektet.

```
fs = 48000;
audioFrameLength = 1024/4;
deviceReader = audioDeviceReader(...
    'Device', 'miniDSP ASIO Driver',...
    'Driver', 'ASIO', ...
    'SampleRate', fs, ...
    'NumChannels', 16 ,...
    'OutputDataType', 'double',...
    'SamplesPerFrame', audioFrameLength);
```

#### 3.3.2.2 Inläsning av webbkamera

Inläsningen av webcams sker via "MATLAB Support Package for USB Webcams add-on". Först skapar man en webcamlistobjekt som läser in alla inkopplade USB-webbkameror. Därefter skapar man ett webcamobjekt och specificerar vilken från listan. Sist väljer man hur många bilder per sekund man vill läsa av kameran och i vilken upplösning via 'Resolution' respektive 'FrameRate' egenskaperna genom punktnotation.

```
camList = webcamlist;
cam = webcam('Integrated Camera');
cam.Resolution = "1280x720";
```

#### 3.3.2.3 Superponering över webbkameravideo

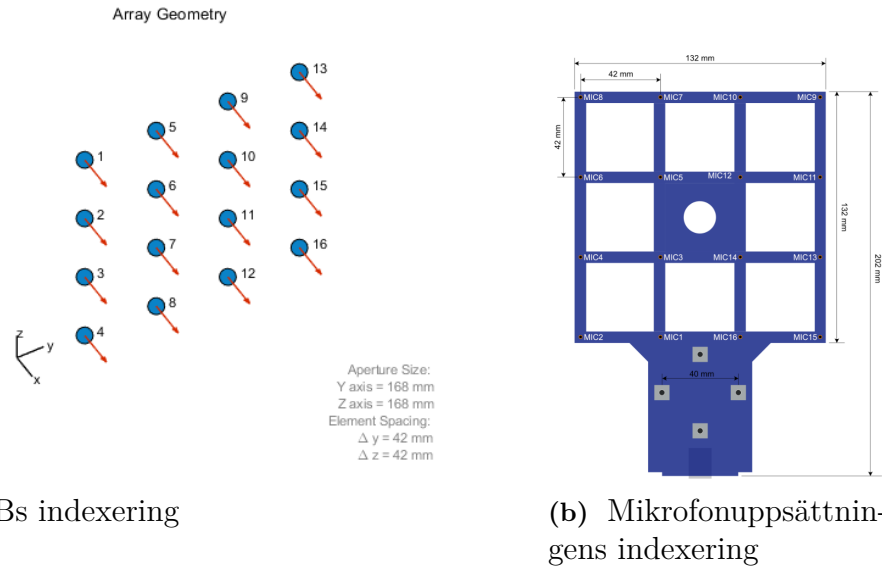
Superponering sker via "MATLAB Support Package for USB Webcams add-on". Först skapar man matrisen av ljudtrycksnivåerna av alla mätvinklar via strålskanningsfunktionen. Därefter tar man en stillbild från webbkameran uppföljt av en "hold on" för att kunna rita över den. Man omvandlar sedan matrisen av ljudtrycksnivåer till ett transparent färgdiagram och superponerar den över stillbilden. Sist ritas stillbilden tillsammans med det superponerade färgdiagrammet ut.

```
while true
    beamscan = beamscan_realtime_mex(rx, reswidth,
        resheight, n, hfov, vfov);
    B = imresize(beamscan, n, "bilinear");
    imshow(snapshot(cam));
    hold on;
    im = imagesc(B);
    im.AlphaData = 0.5;
    hold off;
    drawnow
end
```

#### 3.3.2.4 Indexering av ljudkanalerna

Eftersom phased.TimeDelayBeamformer-systemobjektet antar att kolonnerna i PCM-strömmatrisen, rx, är sorterade indexvis, det vill säga första kolonnen korrespon-

derar till PCM-strömmen från sensorn med index 1 och så vidare, behövdes det indexeras om. Detta gjordes genom att först visualisera URA-systemobjektet med hjälp viewArray-funktionen med 'ShowIndex'-argumentet sätt till 'All' vilket senare jämfördes med indexering av mikrofonuppsättningen vilket framgår från dess ritning(se figur 3.4).



**Figure 3.4:** Indexering. Bild hämtad ifrån [14].

Därefter skapades ett index i form av en radvektor där mikrofonindexen sett från figur 3.4b mappas elementvis för att matcha indexet som sett från figur 3.4a. Kolonnerna från PCM-strömmatrisen, sound, kopieras sedan indexvist för att skapa en ny PCM-strömmatris, rx, med rätt indexering.

```
index = [8 6 4 2 7 5 3 1 10 12 14 16 9 11 13 15];
for i=1:length(index)
    rx(:,i) = sound(:,index(i));
end
```

#### 3.3.2.5 Strålskanning i realtid

Slutfunktionen med alla dessa beståndsdelar blev till slut följande.

```
function [] = realTime()
cam = webcam('Integrated Camera');
cam.Resolution = "1920x1080";

fs = 48000;
audioFrameLength = 1024/4;
deviceReader = audioDeviceReader(...
    'Device', 'miniDSP ASIO Driver',...
    'Driver', 'ASIO', ...
```

```
'SampleRate', fs, ...
'NumChannels', 16 ,...
'OutputDataType', 'double',...
'SamplesPerFrame', audioFrameLength);
setup(deviceReader);

reswidth = 1920;
resheight = 1080;
n = 40;
hfov = 70.42;
vfov = 43.2;

rx = zeros(audioFrameLength, 16);

while true
    tic;
    sound = deviceReader();
    index = [8 6 4 2 7 5 3 1 10 12 14 16 9 11 13 15];
    for i=1:length(index)
        rx(:,i) = sound(:,index(i));
    end
    beamscan = beamscan_realtime_mex(rx,reswidth,
        resheight,n, hfov, vfov);
    B = imresize(beamscan,n, "bilinear");
    imshow(snapshot(cam));
    hold on;
    im = imagesc(B);
    im.AlphaData = 0.5;
    hold off;
    drawnow
    disp(1/toc)
end
end
```

För att kunna genomgå kodgenerering så måste variabler och systemobjekt som används som argument i lobstyrningsobjektet initieras om i varje iteration av parfor-loopen. Mätvinklarna av självla strålskanningen beror på upplösningen *reswidth* och *resheight* samt horisontella respektive vertikala synfältet *hfov* och *vfov* av kameran. Istället för en mätvinkel per pixel kan antalet mätvinklar i varje led minskas med hjälp av *n*. Denna minskning är kvadratisk så en minskning på 40 i varje led, vilket valdes för prestandamätningen, leder till en 1600 gånger minskning av antalet mätvinklar i förhållandet till antalet pixlar. Detta skalas sedan bilinjärt upp med hjälp av *imresize*-funktionen till webbkamerans upplösning.

```
function [beamscan] = beamscan_realtime(rx, reswidth,
    resheight, n, hfov, vfov)
    beamscan=zeros(resheight/n,reswidth/n);
    parfor row = 1:resheight/n
```

```
angAzimuth = linspace(-hfov/2, hfov/2, reswidth/n
);
angElavation = linspace(-vfov/2, vfov/2,
    resheight/n);
array = phased.URA('Size',[4 4],'Lattice','
    Rectangular','ArrayNormal','x');
array.ElementSpacing = [42/1000 42/1000];
beamformer = phased.TimeDelayBeamformer('
    SensorArray',array,...
'SampleRate',48e3,'PropagationSpeed',340,'
    DirectionSource','Input port');
beamscan_column = zeros(reswidth/n,1);
for column = 1:reswidth/n
    angSteer = [angAzimuth(column); angElavation(
        row)];
    beam = step(beamformer, rx(:,,:), angSteer);
    % beamscan_column(column) = logtal .* log10(
        rms(beam)) + maxdB;
    beamscan_column(column) = rms(beam);
end
beamscan(row,:) = beamscan_column;
end
end
```

### 3.3.2.6 Efterbehandlad strålskanning

Eftersom efterbehandling kan vara önskevärt för datorer med långsam hårdvara eller ifall man enkelt vill kunna spara mätningar modifierades realtidfunktionen för att kunna göra just detta. Som argument tar funktionen in hur många sekunder och i hur många bilder per sekund man vill mäta. PCM-strömmatriserna sparas som en tredimensionell matris och stillbilderna från webbkameran sparas som en 'movie frame' som sedan efterbehandlas för att strålskannas och superponeras för att bilda en uppsättning av 'movie frames' som kan efterspelas med hjälp av movie-funktionen.

```
function [M] = postProcessing(seconds, fps)
cam = webcam('Integrated Camera');
cam.Resolution = "1920x1080";

fs = 48000;
audioFrameLength = 1024/4;
deviceReader = audioDeviceReader(...
    'Device','miniDSP ASIO Driver',...
    'Driver','ASIO', ...
    'SampleRate', fs, ...
    'NumChannels', 16 ,...
    'OutputDataType','double',...
);
```

```
'SamplesPerFrame', audioFrameLength);
setup(deviceReader);

reswidth = 1920;
resheight = 1080;
n = 40;
hfov = 70.42;
vfov = 43.2;

figure;
frames = seconds*fps;
M(frames) = struct('cdata',[],'colormap',[]);
video(frames) = struct('cdata',[],'colormap',[]);
rx = zeros(audioFrameLength,16, frames);
for k = 1:frames
    sound = deviceReader();
    index = [8 6 4 2 7 5 3 1 10 12 14 16 9 11 13 15];
    for i=1:length(index)
        rx(:,i,k) = sound(:,index(i));
    end
    rgbImage = snapshot(cam);
    imshow(rgbImage);
    drawnow
    video(k) = getframe;
    pause(1/fps);
    disp(1/toc)
end
close all; figure;
tic
parfor j=1:frames
    beamscan = btest(rx,j,reswidth,resheight,n, hfov,
        vfov);
    B = imresize(beamscan,n, "bilinear");
    webcamframe = frame2im(video(j));
    imshow(webcamframe);
    hold on;
    im = imagesc(B);
    im.AlphaData = 0.5;
    hold off;
    drawnow nocallbacks
    M(j) = getframe;
    disp(j)
end
toc
close all;
end
```

Strålskanningsfunktionen modifierades för att undvika ha en parfor-loop inuti en for-loop och för att kunna läsa av dem lagrade PCM-strömmatriserna.

```
function [beamscan] = btest(rx, j, reswidth, resheight, n,
    hfov, vfov)
    beamscan=zeros(resheight/n,reswidth/n);
    angAzimuth = linspace(-hfov/2, hfov/2, reswidth/n);
    angElavation = linspace(-vfov/2, vfov/2, resheight/n)
    ;
    array = phased.URA('Size',[4 4], 'Lattice', '
        Rectangular', 'ArrayNormal', 'x');
    array.ElementSpacing = [42/1000 42/1000];
    beamformer = phased.TimeDelayBeamformer('SensorArray'
        ,array, 'SampleRate',48e3, 'PropagationSpeed',340, '
        DirectionSource', 'Input port');
    for row = 1:resheight/n
        beamscan_column = zeros(reswidth/n,1);
        for column = 1:reswidth/n
            angSteer = [angAzimuth(column); angElavation(
                row)];
            beam = step(beamformer, rx(:,:,j), angSteer);
            beamscan_column(column) = rms(beam);
        end
        beamscan(row,:) = beamscan_column;
    end
end
```

Den returnerade uppsättningen av 'movie frames' kan sedan spelas tillbaks 'n' antal gånger i 'fps' bilder per sekund.

```
movie(M,n,fps)
```

För bäst återskapning bör 'fps' väljas så nära som dem utmatade bilderna per sekund under inspelningen.

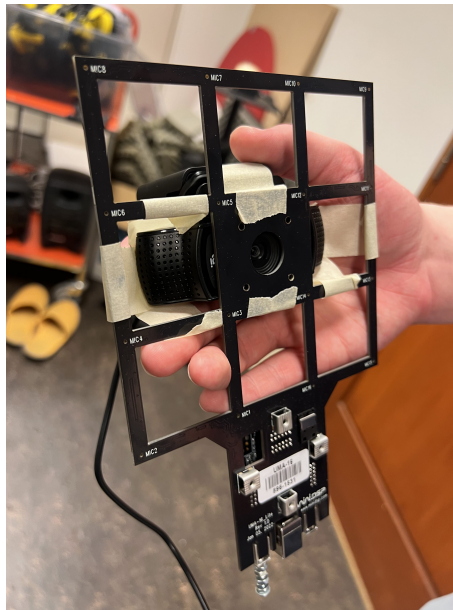
## 3.4 Konstruktion

Denna sektion ger en djupgående analys av konstruktions- och monteringsprocessen för den akustiska kameran samt utforskar de olika stegen som användes för att montera enheten.

### 3.4.1 Webbkameramontering

Valet av kamera föll på en modell från Logitech HD Pro Webcam C920 med 1920x1080 upplösning och ett 78° diagonalt synfält[20]. Detta korresponderar till en horisontell respektive vertikal synfält på 70.42° och 43°.

Monteringen var en enkel "gör-det-själv"-process. Tejp användes för att säkra kameran på mikrofonuppsättningen, vilket illustreras i figur 3.5 .



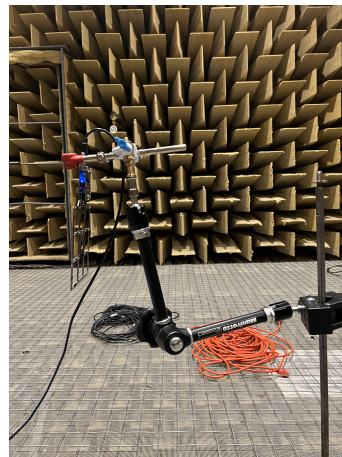
**Figur 3.5:** Bild på hur kameran monterades fast på mikrofonuppsättningen med hjälp av tejp.

#### 3.4.2 Stativ

Handledaren gav gruppen ett Manfrotto-stativ med tre justerbara punkter. En klämme på ena änden användes för att fästa stativet vid ett bord eller liknande objekt, medan den andra änden hade en inbyggd skruv för att fästa arrayen.



(a) Montering av mikrofonuppsättningen.



(b) Fullständig akustisk kamera.

**Figur 3.6:** Mikrofonuppsättningens montering till stativet a) och den fullständiga akustiska kameran b).

En utmaning uppstod när inbyggda skruven på det ursprungliga stativet (Manfrotto) visade sig vara för stor för att passa i skruvfästet för mikrofonuppsättningen. För

att lösa detta problem användes ett mindre stativ med klämmor på båda ändarna. En skruv fästes på mikrofonuppsättningen, med den ena änden av det nya stativet fäst vid den inbyggda skruven, medan den andra änden säkrades på mikrofonuppsättningens skruv. Dock var skruven på mikrofonuppsättningen för liten för att klämman skulle få tillräckligt grepp. För att hantera detta problem placerades sju muttrar på skruven för att ge klämman bättre fäste som illustreras på figur 3.6(a).

#### **3.4.3 Kalibrering**

Det troddes först behövas kalibreringen av mätvinklarna men webbkameratillverkarnas angivna synfält som matades in i funktionerna observerades stämma väldigt bra överens efter några testmätningar.

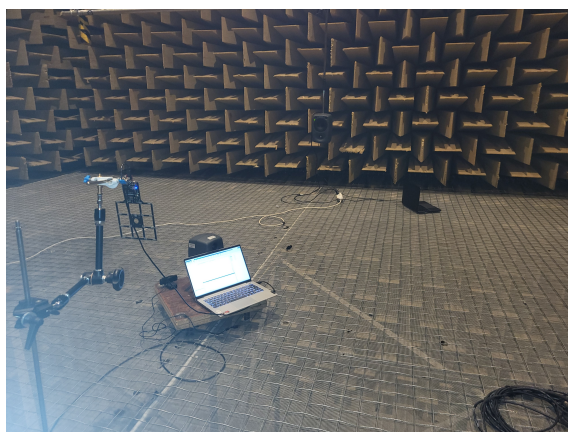
# 4

## Resultat

I detta avsnitt presenteras mätresultet och prestandan hos den den färdiga produkten.

### 4.1 Prestandamätningar

Prestandamätningarna utfördes i akustiklabbet hos avdelningen för teknisk akustik, som är utrustat med faciliteter för ljudmätningar. Särskilt framträdande är deras ekofria rum, en miljö som är noga konstruerad för att minimera reflektioner och externa störningar, vilket säkerställer högsta möjliga precision vid mätningarna.



**Figur 4.1:** En bild av ekofria rummet med den akustiska kameran placerad 3 m från en utav studiohögtalarna

### 4.2 Resultat från prestandamätningar

I detta avsnitt presenteras frekvensomfånget, dynamikomfånget och urskiljningsförmågan av den färdiga produkten.

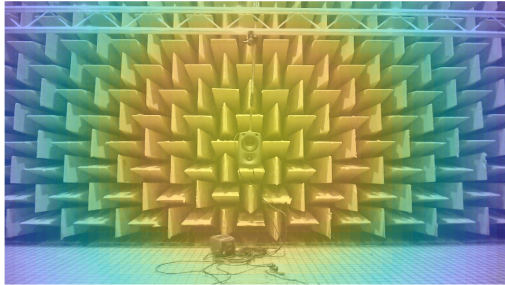
#### 4.2.1 Frekvensomfång

En studiohögtalare placerades 3 meter från mikrofonuppsättningen, enligt figur 4.1. Därefter påbörjades frekvensmätningarna genom mata ut en sinuston med en startfrekvens på 1 kHz. Frekvensen trappades stegvis upp med en 1 kHz åt gången tills

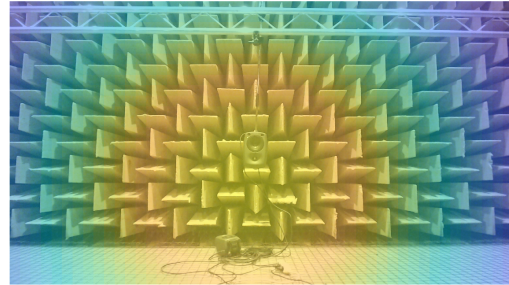
## 4. Resultat

---

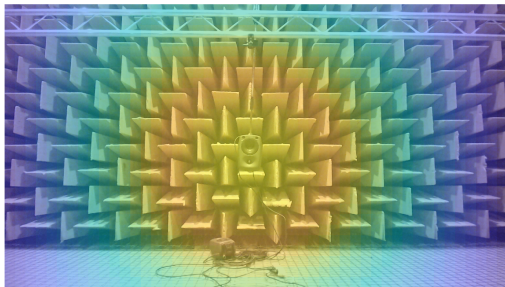
ytterligare punkter i form av märkbar vikning uppträdde på bilden. Detta fenomen observerades vid 12 kHz enligt figur 4.2.



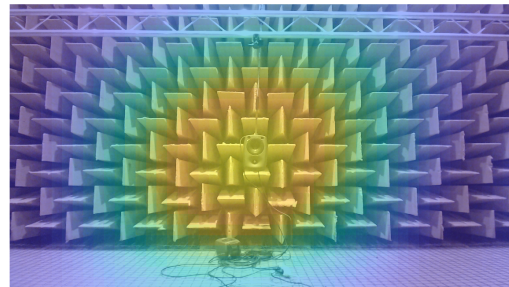
(a) 1 kHz



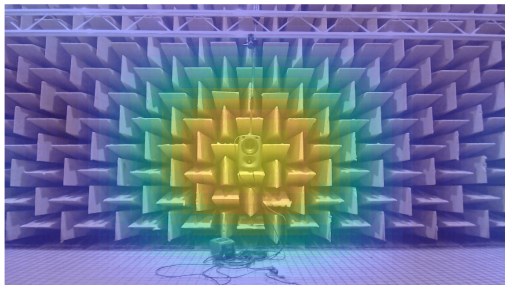
(b) 2 kHz



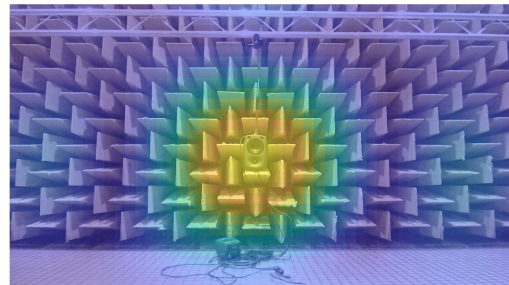
(c) 3 kHz



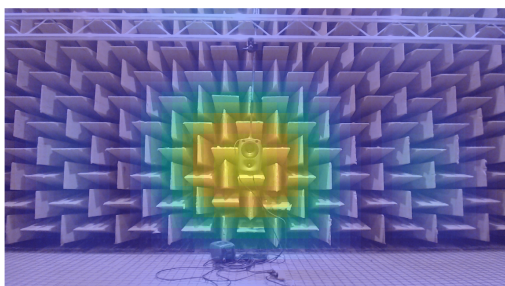
(d) 4 kHz



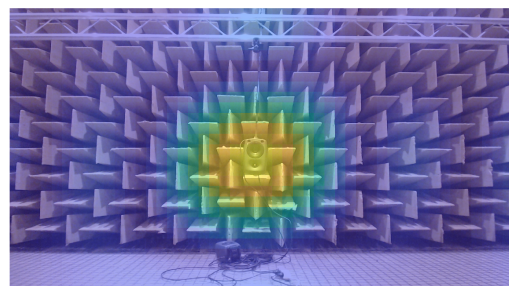
(e) 5 kHz



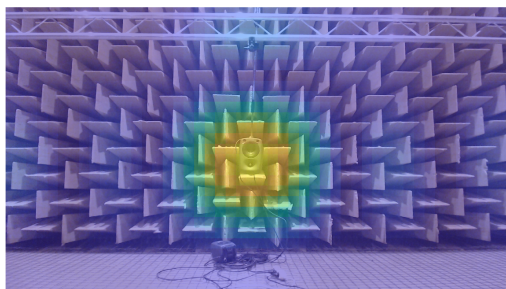
(f) 6 kHz



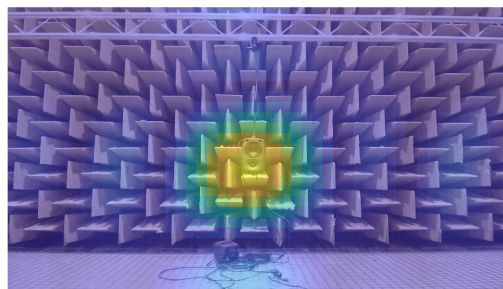
(g) 7 kHz



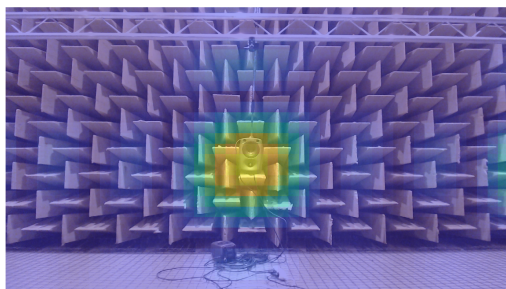
(h) 8 kHz



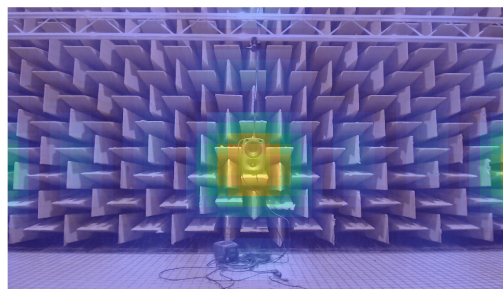
(i) 9 kHz



(j) 10 kHz



(k) 11 kHz



(l) 12 kHz

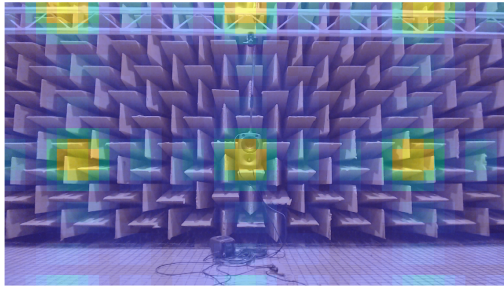
**Figur 4.2:** Videoutmatning från den akustiska kameran vid utmatning av sinus-ton av varierande frekvenser från en studiohögtalare placerad 3m ifrån kameran. Frekvensintervallet går från 1 kHz till 12 kHz.

Vikning observerades längst åt höger redan i figur 4.2 k) men utvecklades aldrig till gult innan 4.2l) så därför stannades mätningar just där.

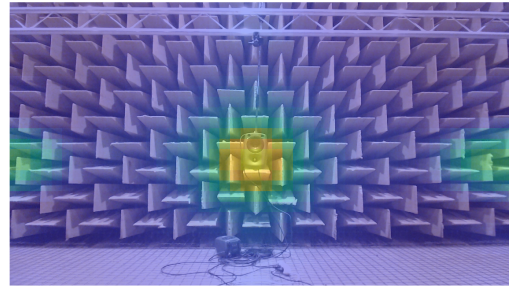
## 4. Resultat

---

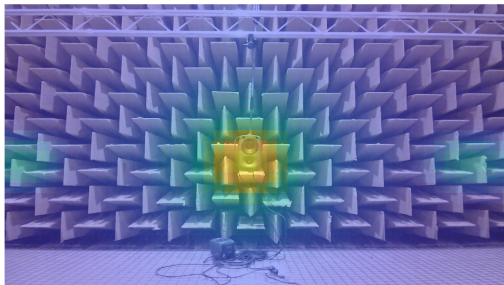
För att tydliggöra avvikelserna i det akustiska sämsta möjliga fallet utfördes ytterligare prestandamätningar vid 20 kHz samt mätningar av icke-rena toner i form av vitt, rosa och brunt brus (se figur 4.3 nedan).



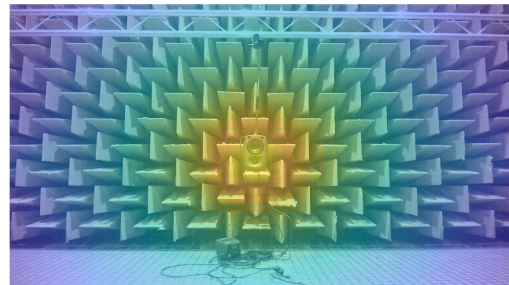
(a) 20 kHz



(b) Vitt brus



(c) Rosa brus



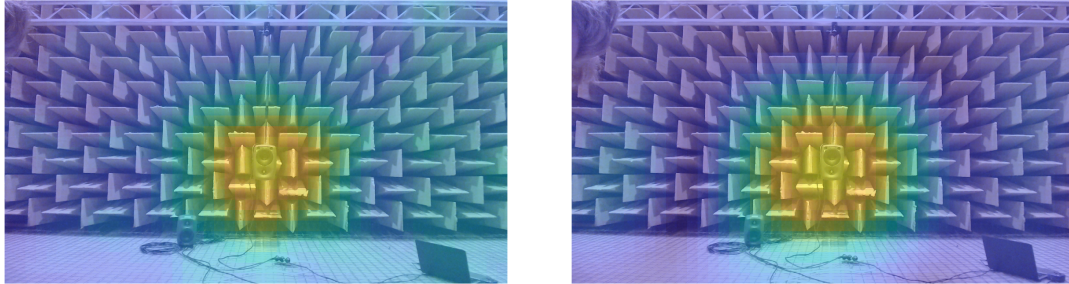
(d) Brunt brus

**Figur 4.3:** Videoutmatning från den akustiska kameran vid utmatning av sinuston på a) 20 kHz samt b) vitt c) rosa och d) brunt brus från en studiohögtalare vid  $70 \text{ dB}_A$  placerad 3m ifrån kameran

Exponeringen av det vita bruset resulterade i tydligast vikning uppföljt väldigt nära av det rosa bruset. Det bruna bruset upplevde ingen vikning men betydligt sämre lokaliseringsförmåga i form av lobredd.

### 4.2.2 Dynamikomfång

Dynamikomfånget, skillnaden mellan den svagaste mätbara ljudsignalen och den starkaste mätbara ljudsignalen, för den akustiska kameran presenteras i figur 4.4 nedan.



(a) 30 dB<sub>A</sub>

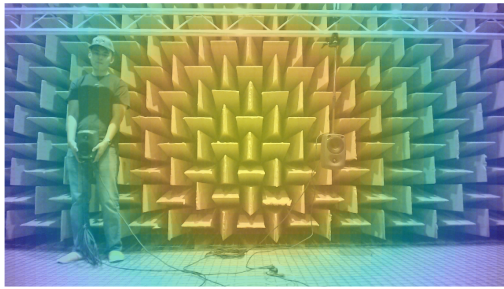
(b) 80 dB<sub>A</sub>

**Figur 4.4:** Videoutmatning från den akustiska kameran vid utmatning av sinuston på 4 kHz från en studiohögtalare vid a) 30 dB<sub>A</sub> och b) 80 dB<sub>A</sub> placerad 3m ifrån kameran.

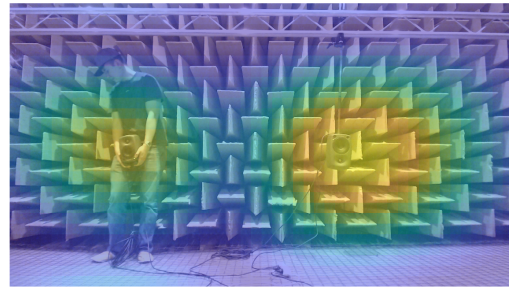
Ljudstyrkenivån varierades tills då brandgult bara utstrålades från högtalaren vilket observerades ske vid 30 dB<sub>A</sub>. Som "sida vid sida"-jämförelse testades även samma 4 kHz sinuston att spelas vid 80 dB<sub>A</sub>. Detta resulterade i att all kvarlevande brus, som man något tydligt kan se till höger om högtalaren i figur 4.4b), eliminerats och att bakgrunden är tydligt mycket djupare blå. Den starkaste mätbara signalen ljudsignalen testades aldrig men framgår från mikrofonernas givna prestandaparametrar(se sektion 3.2.2.1) vara 124 dB<sub>A</sub>. Alltså landar dynamikomfånget på runt 30 dB<sub>A</sub>-124 dB<sub>A</sub>.

### 4.2.3 Ljudurskiljningsförmåga

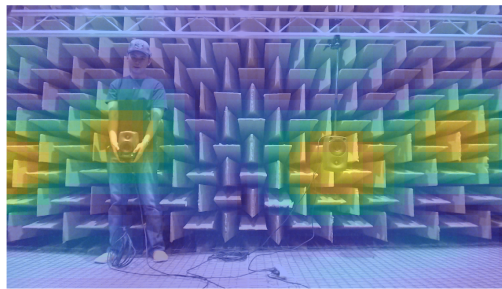
För att evaluera den akustiska kamerans förmåga att lokalisera och urskilja ljud utfördes mätningar med två studiohögtalare i stereokonfiguration 3 m ifrån den akustiska kameran med varierande mellanrum vid utmatning av en sinuston med frekvenserna 2 kHz, 6 kHz och 10 kHz.



(a) 2 kHz

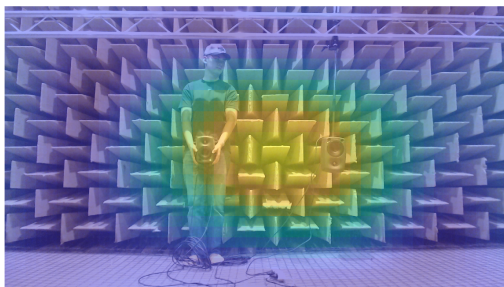


(b) 6 kHz

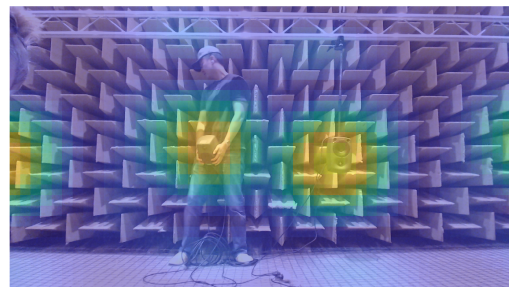


(c) 10 kHz

**Figur 4.5:** Videoutmatning från den akustiska kameran vid utmatning av sinus-ton med varierande frekvens från studiehögtalare i stereokonfiguration med 175 cm mellanrum placerad 3m ifrån kameran.

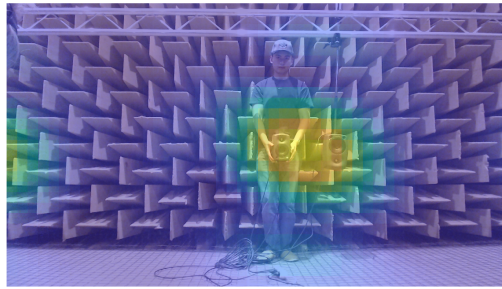


(a) 6 kHz



(b) 10 kHz

**Figur 4.6:** Videoutmatning från den akustiska kameran vid utmatning av sinus-ton med varierande frekvens från studiehögtalare i stereokonfiguration med 85 cm mellanrum placerad 3m ifrån kameran.



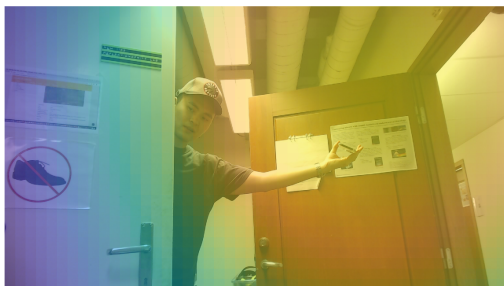
(a) 10 kHz

**Figur 4.7:** Videoutmatning från den akustiska kameran vid utmatning av sinuston med varierande frekvens från studiohögtalare i stereokonfiguration med 30 cm mellanrum placerad 3m ifrån kameran.

Som sett från figur 4.5, 4.6 och 4.7 ovan så varierades avståndet tills studiohögtalarna som spelar den lägst frekventa sinustonen blir urskiljbara. Detta skedde vid 175 cm för 2 kHz, 85 cm för 6 kHz och tillslut 30 cm för 10 kHz. Väldigt tydlig vickning observerades vid stereoutmatning av sinustonen på 10 kHz från alla tre mellanrum men inte hos någon av dem andra två frekvenserna.

### 4.3 Vardagliga mätningar

I figuren 4.8 nedan presenteras två testmätningar för att kontrollera användbarheten utanför det ekofria rummet.



(a) Mobil som spelar musik



(b) Laptophögtalare som spelar en sinuston på 8 kHz

**Figur 4.8:** Videoutmatning från den akustiska kameran vid mätningar av vardagliga objekt i ett vanligt rum.

# 5

## Diskussion och slutsats

Detta avsnitt behandlar sammanfattningen av slutprodukten samt en diskussion om dess begränsningar och eventuella användsområden.

### 5.1 Mätresultat

I följande sektion diskuteras mätresultaten från prestandamätningarna.

#### 5.1.1 Frekvensomfång

Frekvensomfånget stämde bra överens med simuleringarna av mikrofonuppsättningen (se bilaga B). Den presterade förvånansvärt bra vid exponering av alla testade typer av brus även dem mer lågfrekventa rosa och brunt. Dock ser man på mätningarna från figur 4.2 att det vore relevant med en bandpassfilter i och med den extremt breda lobbredden vid lägre frekvenser och MSL vid dem allra högsta frekvenserna.

#### 5.1.2 Dynamikomfång

Mikrofonerna angivna prestandaparametrar förutspådde resultaten mycket bra (se sektion 3.2.2.1). Eftersom självbruset av mikrofonerna typiskt låg på runt  $29 \text{ dB}_A$  är det mycket rimligt att mätbara resultat först dyker upp vid runt  $30 \text{ dB}_A$ .

#### 5.1.3 Ljudurskiljningsförmåga

Förmågan att urskilja två ljudkällor är högst beroende på lobbredden. Dessutom ställs det krav på MSL för att inte förvränga resultaten som observerades för 10 kHz i figur 4.6. Det blir enklare att urskilja högre frekvenser i och med snävare lobbredd men man löper samtidigt större risk för sidlobber att förvränga resultaten. Därför är det viktigt att inte mäta för många högfrekventa ljudkällor samtidigt för inte få vilseledande mätresultat. Inte heller bör för lågfrekventa ljudkällor ty hur dålig lobbredd mikrofonuppsättningen har som sett från frekvensomfångsmätningarna och bilaga B.

## 5.2 Kända buggar

För att få MATLAB att ladda in webbkameran måste man först gå till "hardware manager" under "appar" och klicka på "acquire live preview". Detta öppnar ett livescript som läser in webbkameran och sparar det som ett objekt bland alla andra variabler. Därför bör "clear all"-kommandot undvikas för att slippa göra det flera gånger om.

## 5.3 Användbarhet

Med en genomsnittlig uppdateringsfrekvens på runt 5 på en någorlunda modern åttakärnig bärbar dator så är den användbar men ganska begränsad i vardagliga miljöer. Snabbt rörande fordon vore till exempel jobbigt att mäta. MATLAB-implementation som vi gick för tvingar en även att släpa runt en bärbar dator vilket förvärrar smidigheten ganska avsevärt. Mätningar utanför det ekofria rummet gick (se figur 4.8) men behövdes spelas betydligt högre för att kompensera för all eko och störande ljud från omgivningen.

## 5.4 Potentiella förbättringar

Prestandan kan förbättras med hjälp av en FPGA-implementation. DAS-algoritmen som användes för arbetet är egentligen idealt för detta på grund av hur enkel och parallelerbar den är. För den kommersiella lösningen skulle FPGA:n kommunicera fram och tillbaks via USB vilket förvärrar latensen av en sådan lösningen något. I en optimal FPGA-lösning med mikrofonerna direktkopplade via PCB skulle superponering kunna ske direkt i FPGA:n.

## 5.5 Jämförelse mellan den uträknade ideala uppsättningsgeometrin

En mikrofonuppsättning med den optimala geometrin förutsås ha mycket bättre frekvensomfång förutspått av dess styrda och icke-styrda MSL-grafter i bilaga A figur A.3 och A.4. Dessutom förutspås den ha avsevärt bättre lobbredd vid alla frekvenser tack vare dess större diameter vilket skulle ge den bättre lokalisings- och urskilningsförmåga speciellt vid dem lägre frekvenserna (se bilaga A tabell A.2 och bilaga B tabell B.1).

## 5.6 Etiska aspekter

Akustiska kameror väcker flera etiska komplikationer i framtida implementationer. Tekniken har lyft upp som lösning till tidigare väldigt svårgenomdrivbara lagar som till exempel bullernivåer i städer. En sådan tillsynes oskyldig tillämpning av tekniken kan dock öppna upp en hel Pandaros ask för vidare tillämpningar i syftet

att massövervaka samhället. Akustiska kameror kan i framtiden användas för att fånga upp och lokalisera konversationer på allmänna platser. Detta skulle ge känslan av konstant övervakande för många och leda till att man inte vågar uttala sig. Massadoptionen av övervakningskameror är en integritetskränkning i sig själv men en teknik som kan videoövervaka och, på det, noggrant fånga upp och lokalisera ljud är potentiellt det största hotet mot vår personliga integritet som samhället kommer behöva stå emot inom den överskådliga framtiden.

### 5.7 Slutsats

Sammanfattningsvis uppfyllde slutprodukten alla mål och presterade väldigt förutsägbart från simuleringarna. Med detta som utgångspunkt kan det konstateras att en akustisk kamera med den optimala geometrin skulle prestera mycket bättre. Det finns flera förbättringsområden utöver geometrin såsom smidighet och latens vilket gör det här arbetet relevant för att bygga vidare på i ett fortsättningsprojekt.

Därutöver kan projektet även ses som en genomförbarhetsstudie för DAS- fjärrfältlobstyrning i tidsdomänen som höll upp riktigt bra.

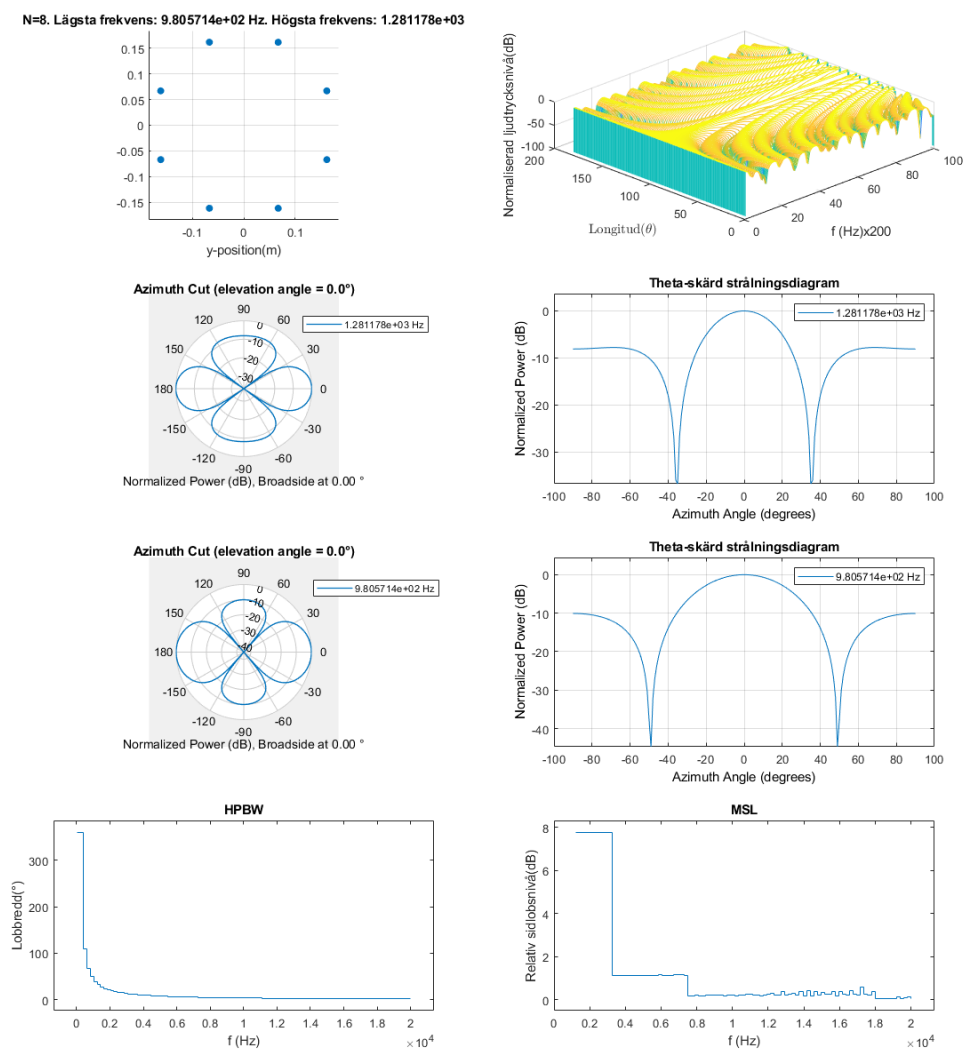
# Bibliografi

- [1] Perrot et al. "So you think you can DAS? A viewpoint on delay-and-sum beamforming," *Ultrasonics*, vol. 111, 106309, Mar. 2021. [Online] Tillgänglig: <https://www.sciencedirect.com/science/article/pii/S0041624X20302444>
- [2] Sorama. "WHAT IS AN ACOUSTIC CAMERA AND HOW DOES IT WORK?", <https://sorama.eu/what-is-an-acoustic-camera>
- [3] M. Carré, "Sound power estimation with an acoustic camera", M.S thesis, Luleå University of Technology, Luleå, Sverige, 2010.
- [4] H. Frid, "Analysis and Optimization of Installed Antenna Performance," Ph.D, KTH, Skolan för elektroteknik och datavetenskap, Stockholm, Sverige, 2020. [Online] Tillgänglig: <https://kth.diva-portal.org/smash/get/diva2:1392934/FULLTEXT01.pdf>
- [5] Virtual Reconstructions, Visualization, Auralization and Interaction within a Cultural Heritage buildings and Sites - Scientific Figure on ResearchGate. [Online] Tillgänglig: [https://www.researchgate.net/figure/Basic-principles-of-the-delay-and-sum-beamformer-in-the-time-and-frequency-domains\\_fig1\\_258837777](https://www.researchgate.net/figure/Basic-principles-of-the-delay-and-sum-beamformer-in-the-time-and-frequency-domains_fig1_258837777)
- [6] Khemapat Tontiwattanakul, Jatupon Hongweing, Peerawit Trakulsatjawat, and Pataweekan Noimai, "Design and build of a planar acoustic camera using digital microphones," Jul. 2019, doi: <https://doi.org/10.1109/iceast.2019.8802537>.
- [7] Näsholm, S. P., Iranpour, K., Wuestefeld, A., Dando, B. D. E., Baird, A. F., & Oye, V. (2022). Array signal processing on distributed acoustic sensing data: Directivity effects in slowness space. *Journal of Geophysical Research: Solid Earth*, 127, e2021JB023587. <https://doi.org/10.1029/2021JB023587>
- [8] Pessentheiner, Hannes & Petrik, Stefan & Romsdorfer, Harald. (2012). Beamforming Using Uniform Circular Arrays for Distant Speech Recognition in Reverberant Environments and Double-Talk Scenarios. 2. 1366-1369.
- [9] J. Tornberg, "MPD-vågformer för helikopterradar", Dissertation, Institutionen för systemteknik, 2003.
- [10] Müller, Leon. (2019). Development of a Real-Time, Low-Budget Acoustic Camera on an FPGA. 10.13140/RG.2.2.20758.68163.
- [11] P. Ioannides and C. A. Balanis, "Uniform circular and rectangular arrays for adaptive beamforming applications," in *IEEE Antennas and Wireless Propagation Letters*, vol. 4, pp. 351-354, 2005, doi: 10.1109/LAWP.2005.857039.
- [12] M. G. M. Hussain, "Beamforming with Uniform-Circular-Array and Ultrawideband-Throb Signal," 2022 IEEE International Symposium on Phased

- Array Systems & Technology (PAST), Waltham, MA, USA, 2022, pp. 1-8, doi: 10.1109/PAST49659.2022.9975053.
- [13] The MathWorks, Inc. “Element and Array Radiation and Response Patterns” mathworks.com. Hämtad: Maj 05, 2024. [Online]. Tillgänglig: <https://se.mathworks.com/help/phased/ug/element-and-array-radiation-patterns-and-responses.html>
- [14] UMA-16 v2 User Manual, V1.2, miniDSP Ltd, Hong Kong, Folkrepubliken Kina, 6 Jun. 2022. Hämtad: Maj 05, 2024. [Online]. Tillgänglig: <https://www.minidsp.com/images/documents/UMA-16%20v2%20User%20Manual.pdf>
- [15] The MathWorks, Inc. “Spherical Coordinates” mathworks.com. Hämtad: Maj 07, 2024. [Online]. Tillgänglig: <https://se.mathworks.com/help/phased/ug/spherical-coordinates.html>
- [16] The MathWorks, Inc. “What Are System Objects?” mathworks.com. Hämtad: Maj 07, 2024. [Online]. Tillgänglig: [https://se.mathworks.com/help/matlab/matlab\\_prog/what-are-system-objects.html](https://se.mathworks.com/help/matlab/matlab_prog/what-are-system-objects.html)
- [17] The MathWorks, Inc. (2022). Phased Array System Toolbox version: 9.14.0.2489007 (R2023a) Update 6. Hämtad: Maj 07, 2024. Available: <https://se.mathworks.com/products/phased-array.html>
- [18] Hirokazu Madokoro et al., “Mallard Detection Using Microphone Arrays Combined with Delay-and-Sum Beamforming for Smart and Remote Rice–Duck Farming,” *Applied sciences*, vol. 12, no. 1, pp. 108–108, Dec. 2021, doi: <https://doi.org/10.3390/app12010108>.
- [19] The MathWorks, Inc. “What Is MATLAB?” mathworks.com. Hämtad: Maj 07, 2024. [Online]. Tillgänglig: <https://se.mathworks.com/discovery/what-is-matlab.html>
- [20] Logitech, Webbkameror/C920 HD PRO WEBCAM, <https://www.logitech.com/sv-se/products/webcams/c920-pro-hd-webcam.960-001055.html>

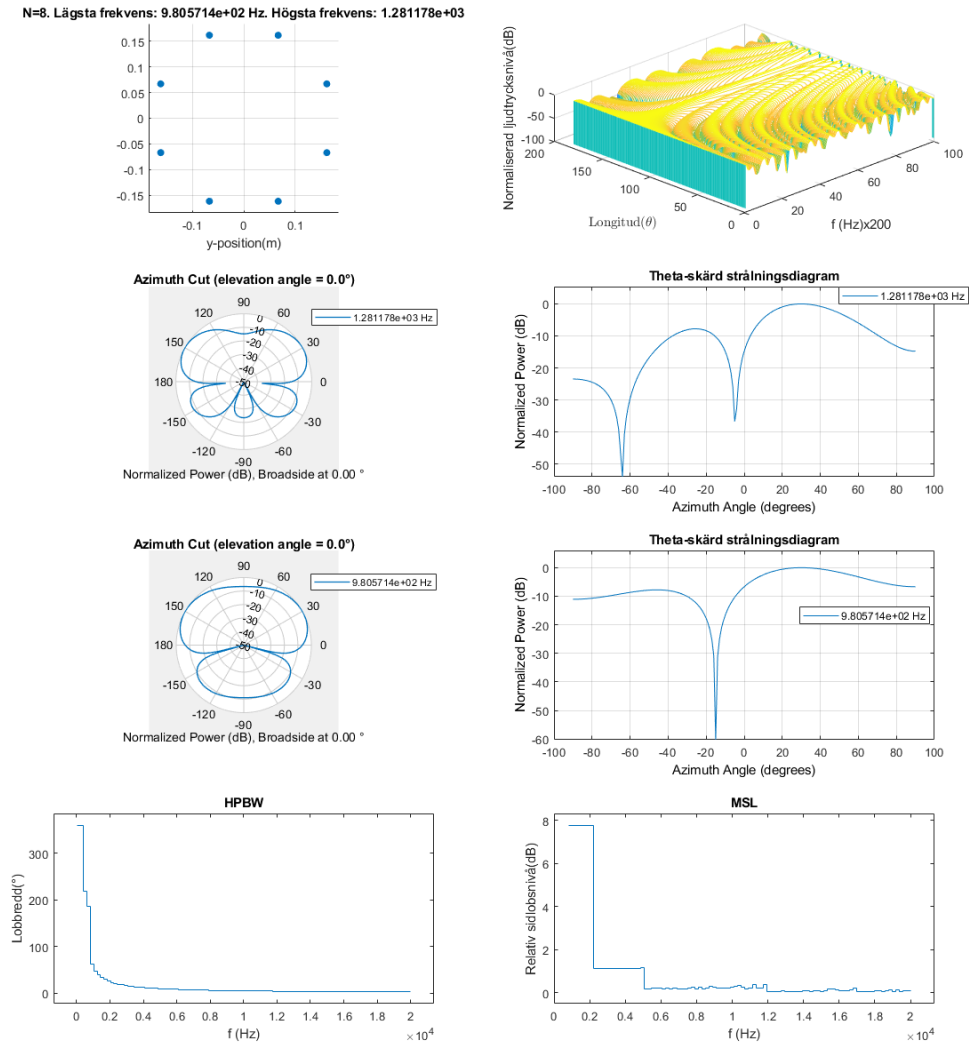
# A

## Simulerade prestandaparametrar



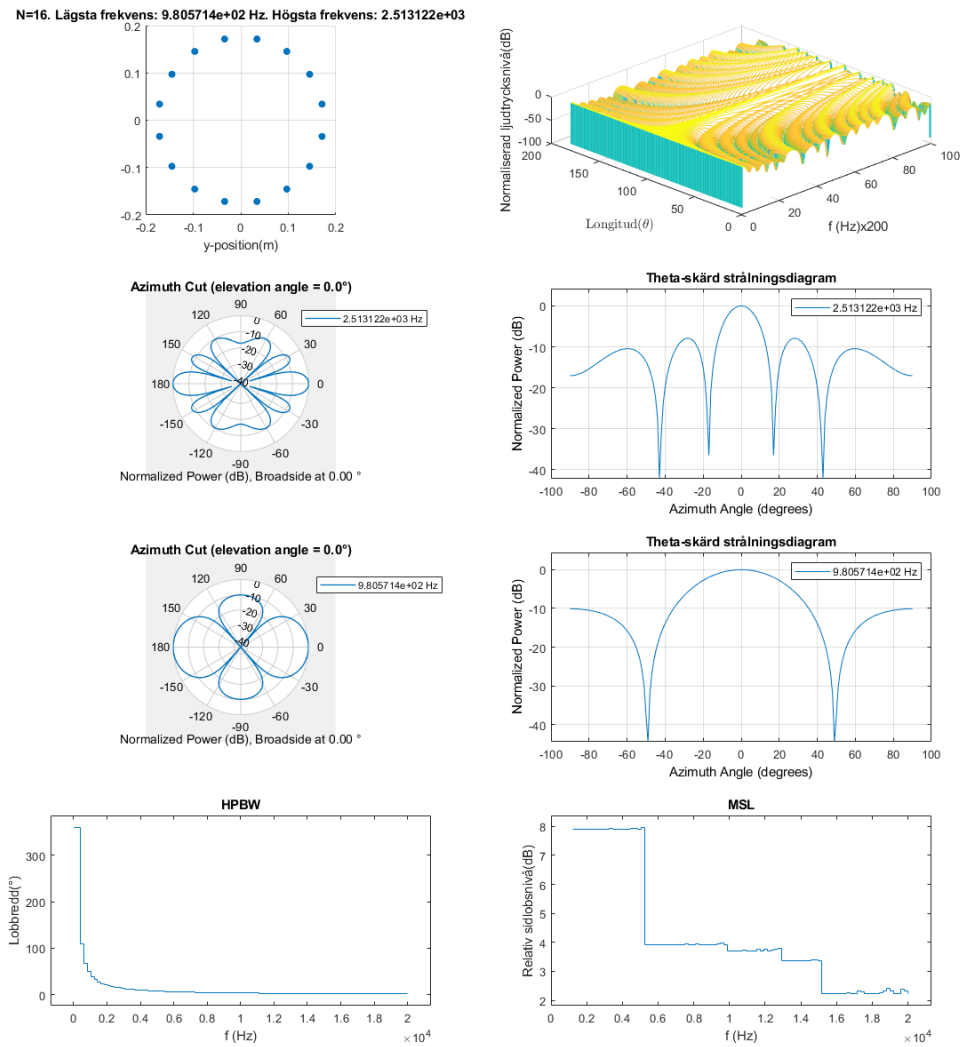
Figur A.1: Prestandaparametrarna av UCA med N=8 mikrofoner och radius  $r=0.175\text{m}$ . Ickestyrd.

## A. Simulerade prestandaparametrar



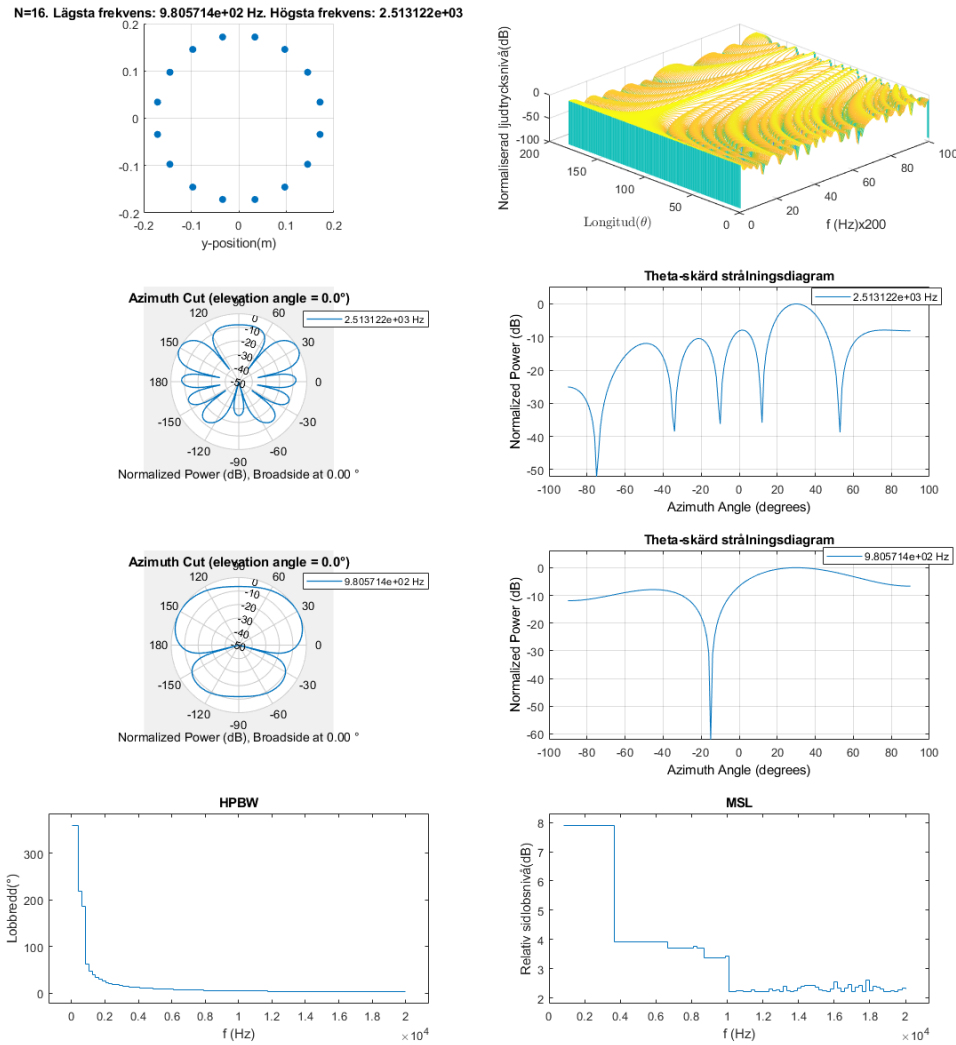
**Figur A.2:** Prestandaparametrarna av UCA med N=8 mikrofoner och radius  $r=0.175\text{m}$ . Styrd till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$

## A. Simulerade prestandaparametrar



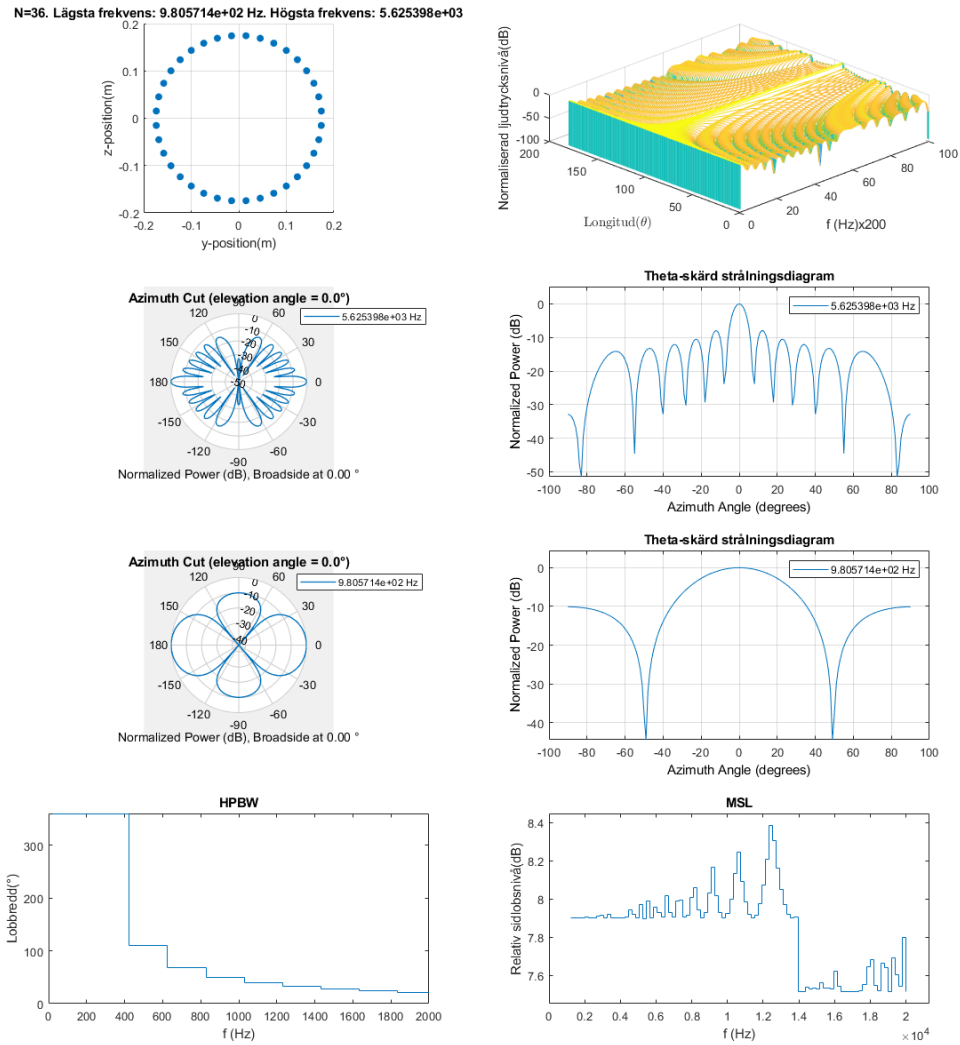
**Figur A.3:** Prestandaparametrarna av UCA med N=16 mikrofoner och radius  $r=0.175\text{m}$ . Ickestyrd.

## A. Simulerade prestandaparametrar



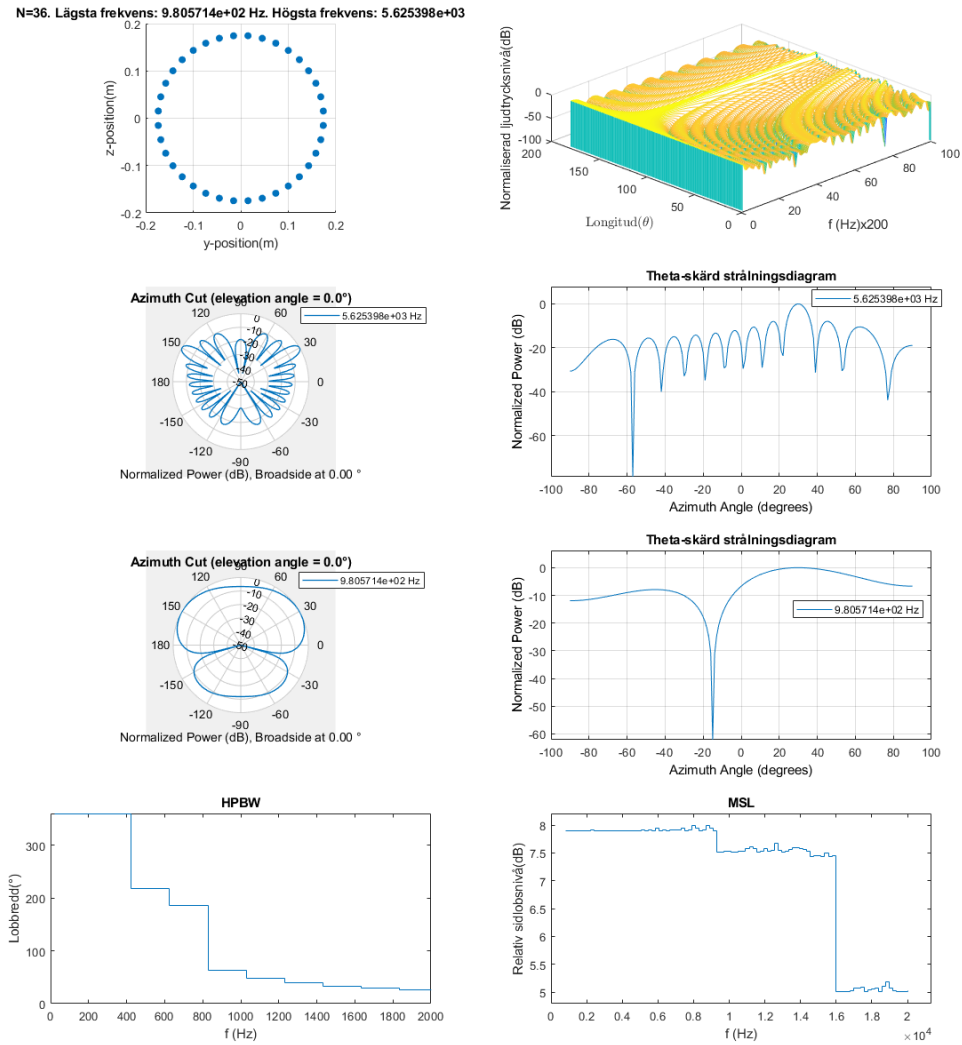
**Figur A.4:** Prestandaparametrarna av UCA med N=16 mikrofoner och radius  $r=0.175\text{m}$ . Styrd till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$

## A. Simulerade prestandaparametrar



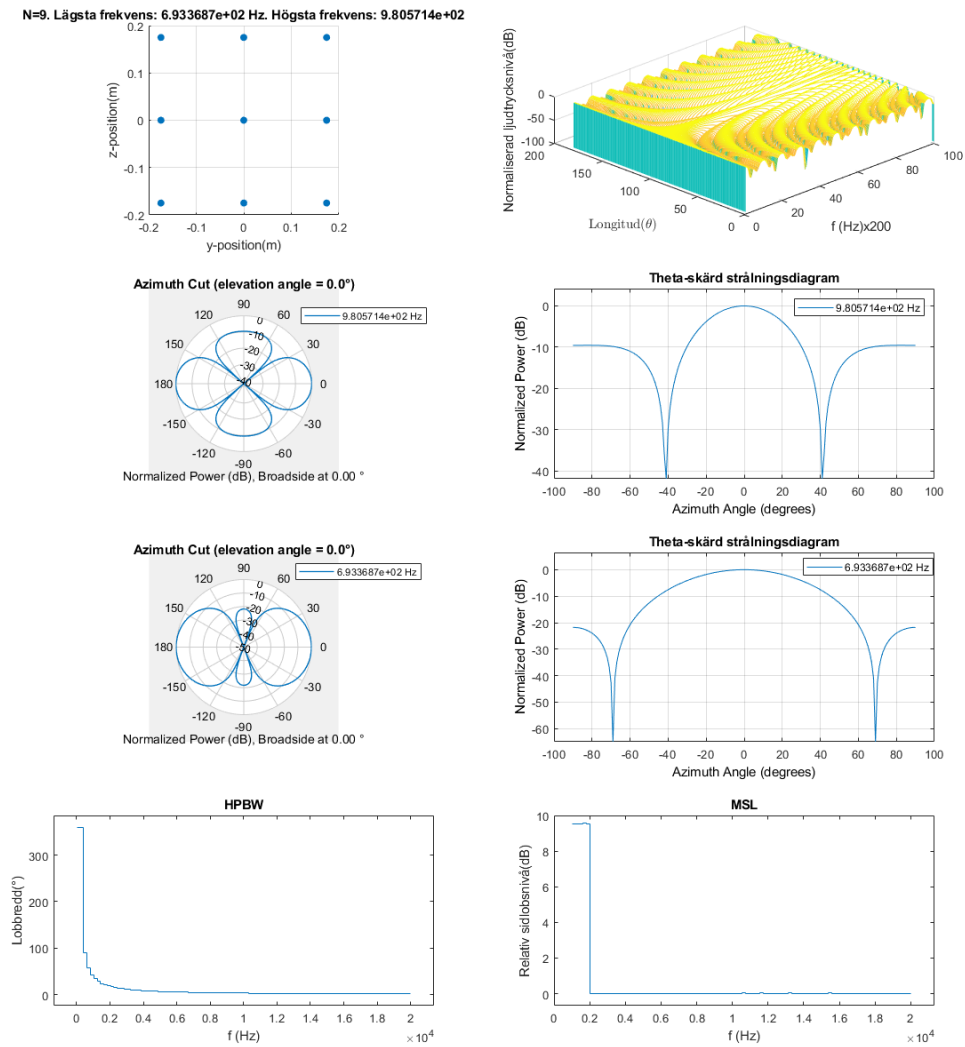
**Figur A.5:** Prestandaparametrarna av UCA med N=32 mikrofoner och radius  $r=0.175\text{m}$ . Ickestyrd.

## A. Simulerade prestandaparametrar



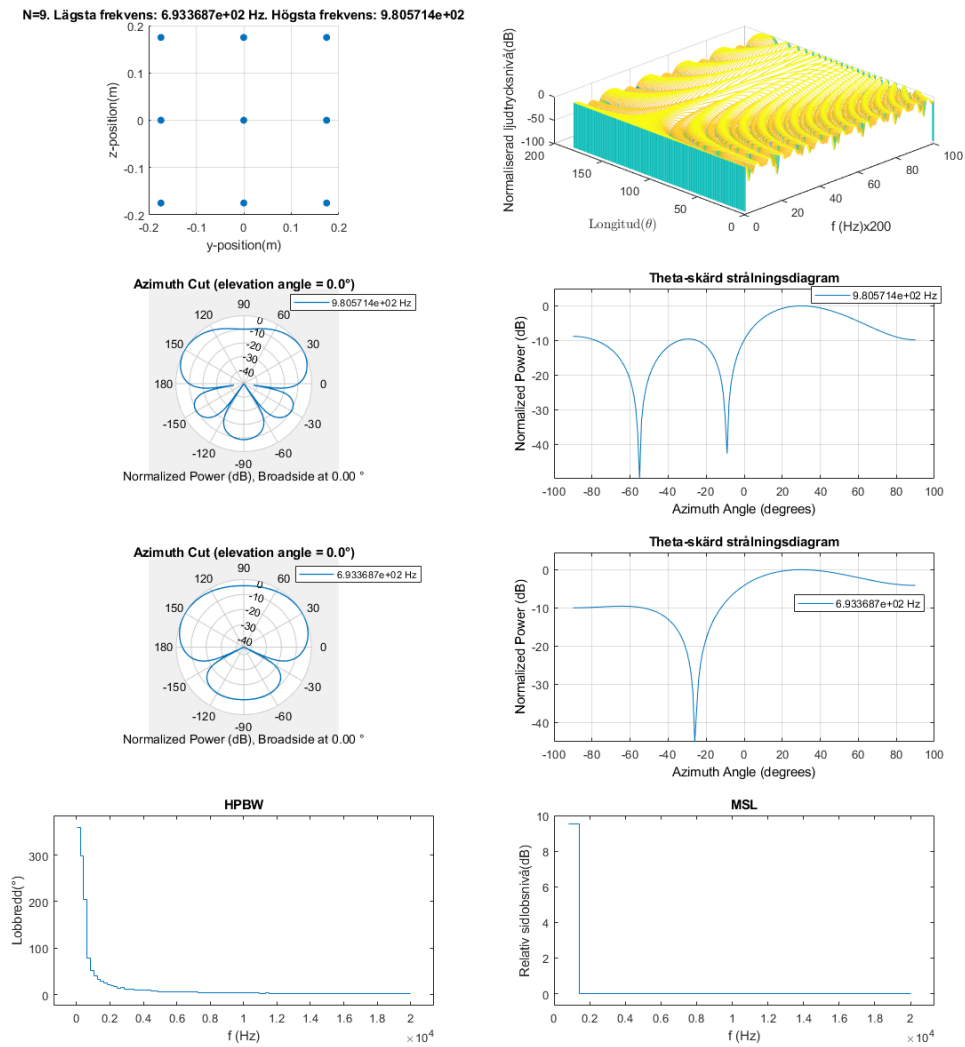
**Figur A.6:** Prestandaparametrarna av UCA med  $N=32$  mikrofoner och radius  $r=0.175\text{m}$ . Styrd till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$

## A. Simulerade prestandaparametrar



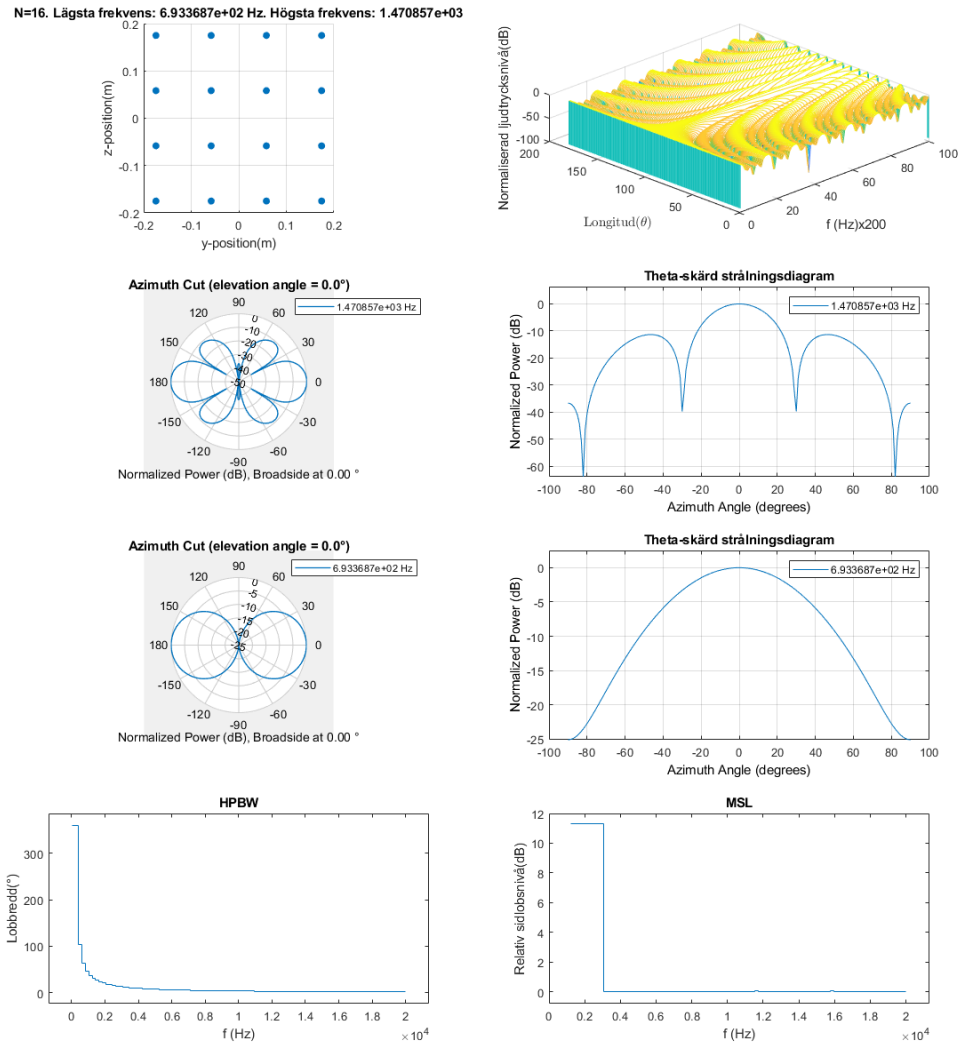
**Figur A.7:** Prestandaparametrarna av 3x3 URA med N=9 mikrofoner och elementavstånd  $d=0.35/2=0.175\text{m}$ . Ickestyrd.

## A. Simulerade prestandaparametrar



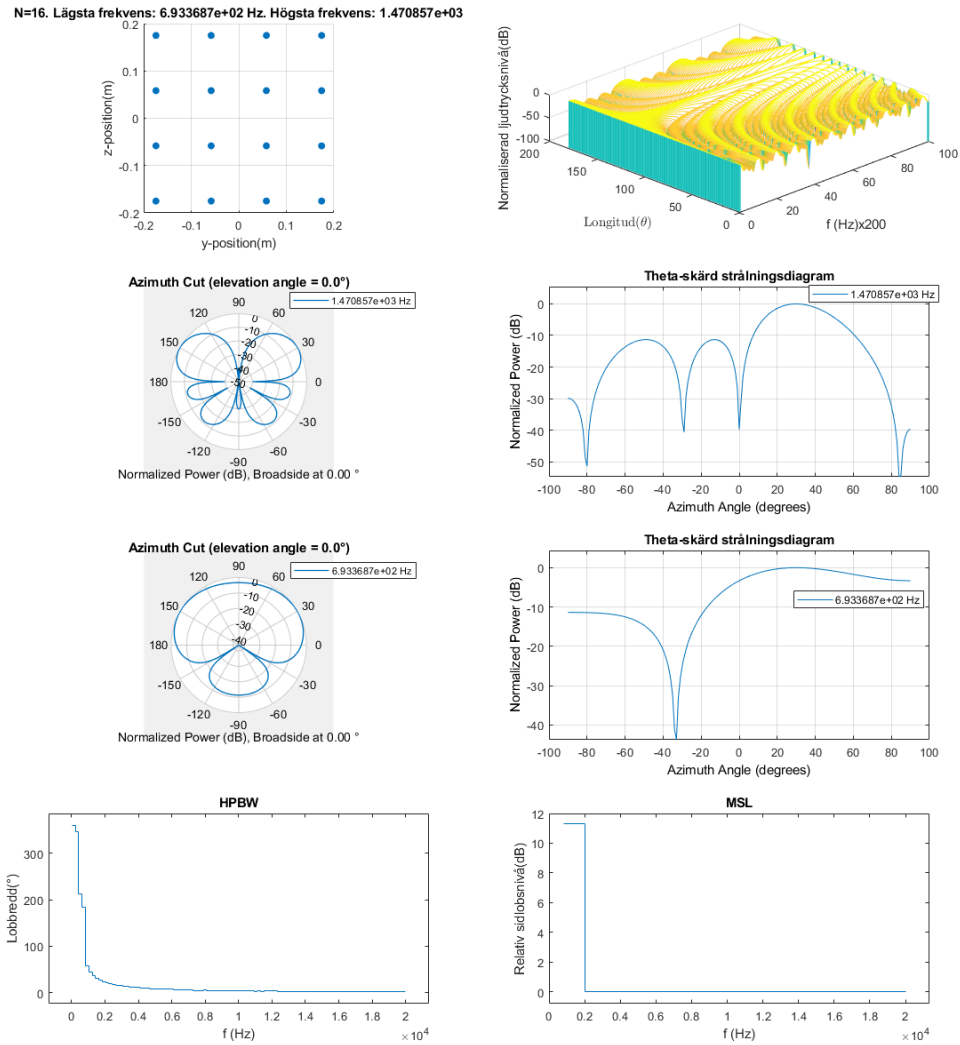
**Figur A.8:** Prestandaparametrarna av 3x3 URA med N=9 mikrofoner och elementavstånd  $d=0.35/2=0.175\text{m}$ . Styrda till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$ .

## A. Simulerade prestandaparametrar

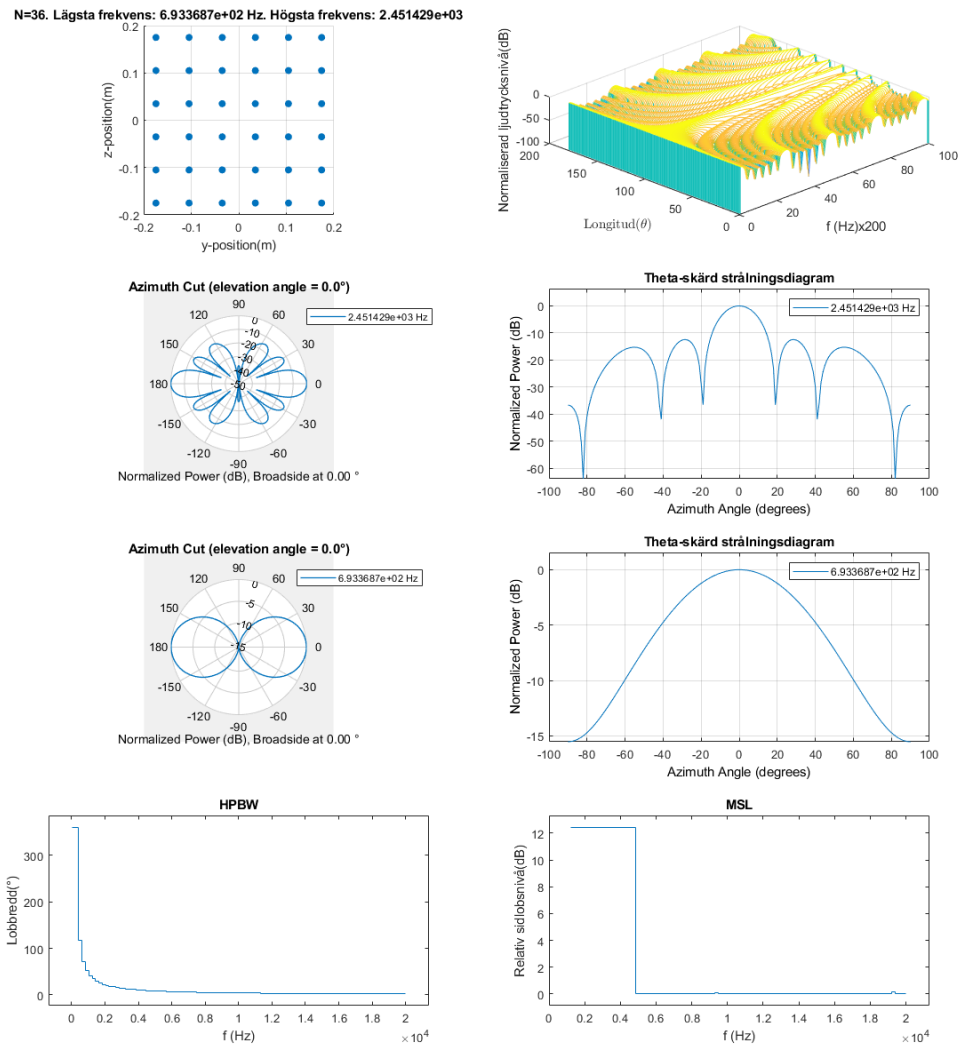


**Figur A.9:** Prestandaparametrarna av 4x4 URA med N=16 mikrofoner och elementavstånd  $d=0.35/3=0.116$  m. Ickestyrd.

## A. Simulerade prestandaparametrar

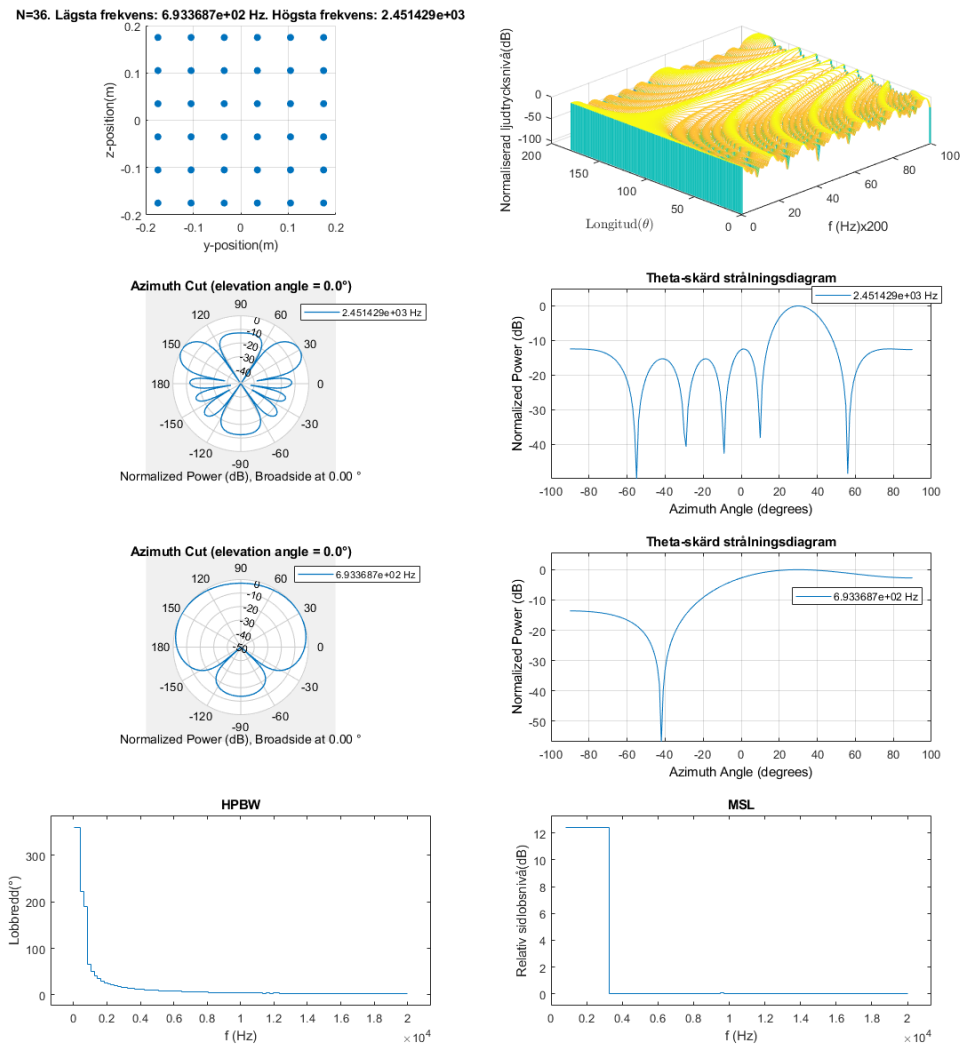


**Figur A.10:** Prestandaparametrarna av 4x4 URA med N=16 mikrofoner och elementavstånd  $d=0.35/3=0.116\text{m}$ . Styrd till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$ .



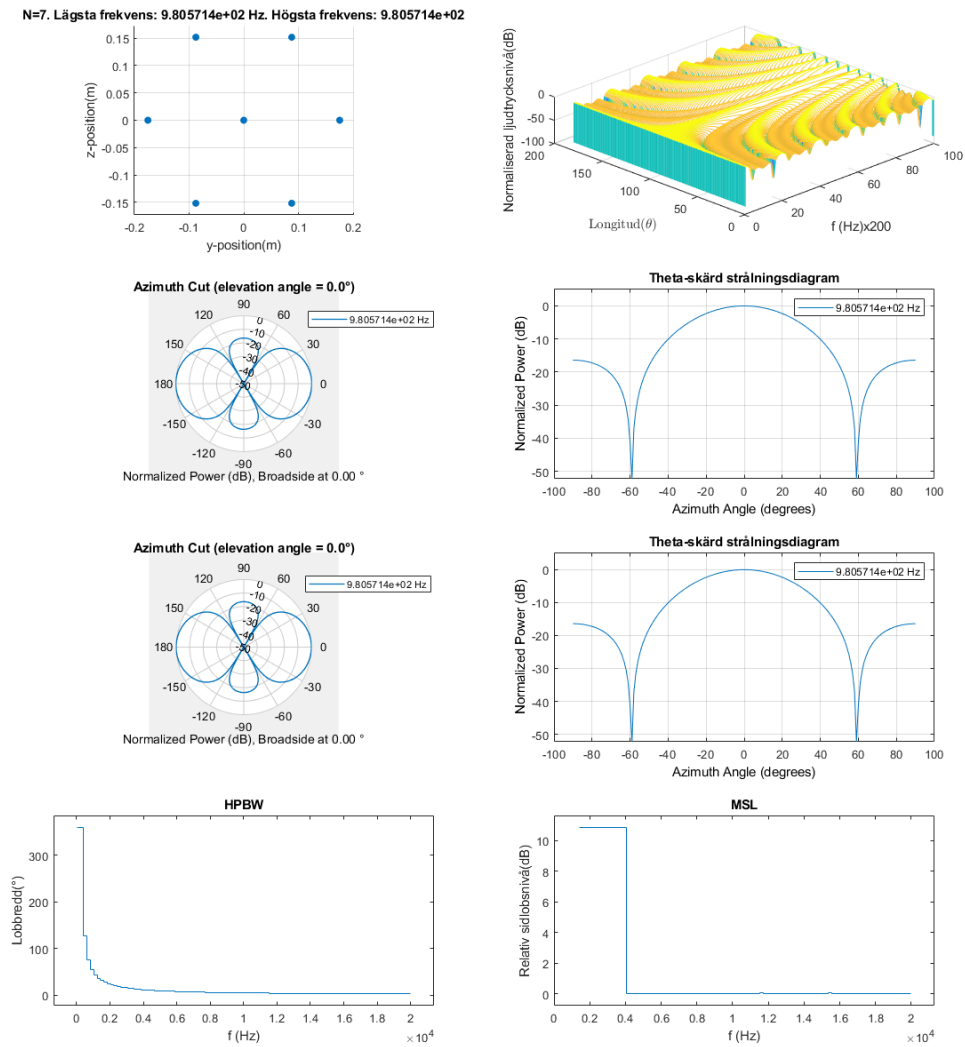
Figur A.11: Prestandaparametrarna av 6x6 URA med N=36 mikrofoner och elementavstånd  $d=0.35/5=0.07$  m. Ickestyrd.

## A. Simulerade prestandaparametrar



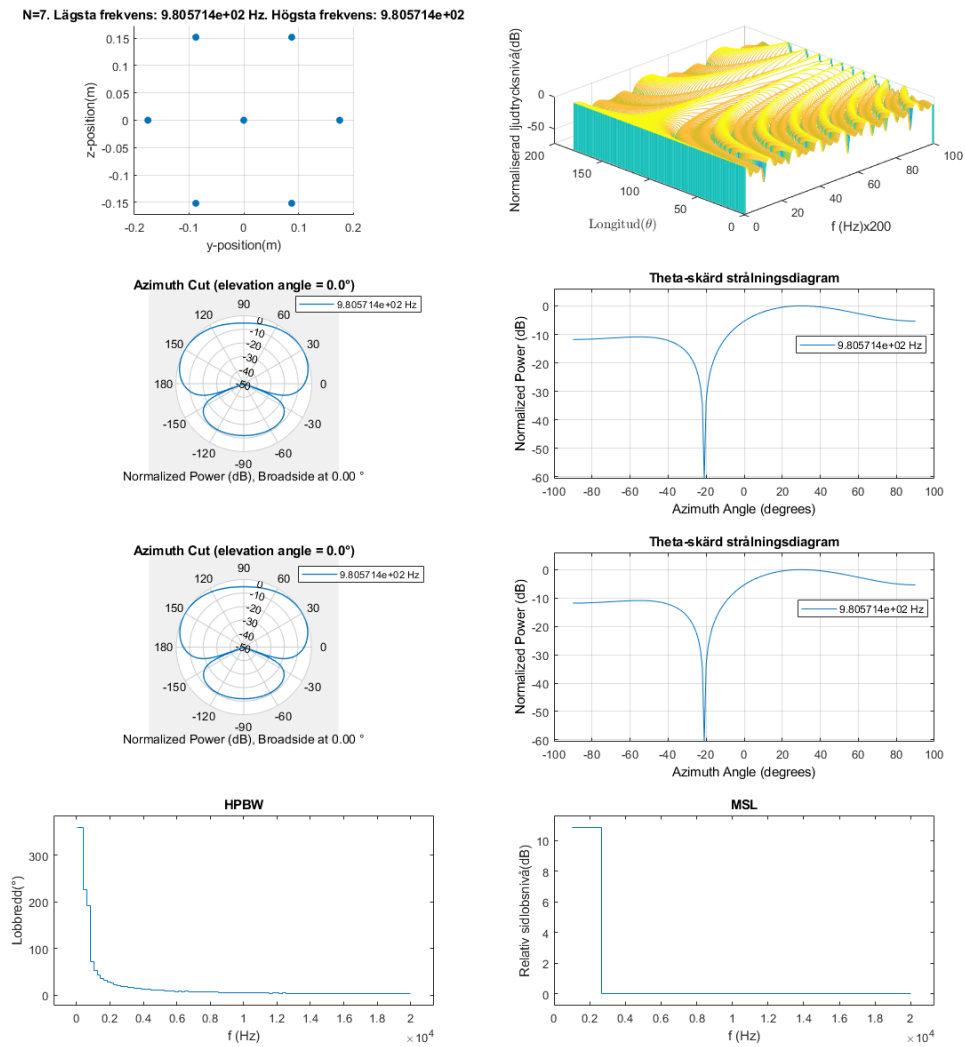
**Figur A.12:** Prestandaparametrarna av 6x6 URA med N=36 mikrofoner och elementavstånd  $d=0.35/5=0.07\text{m}$ . Styrd till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$ .

## A. Simulerade prestandaparametrar



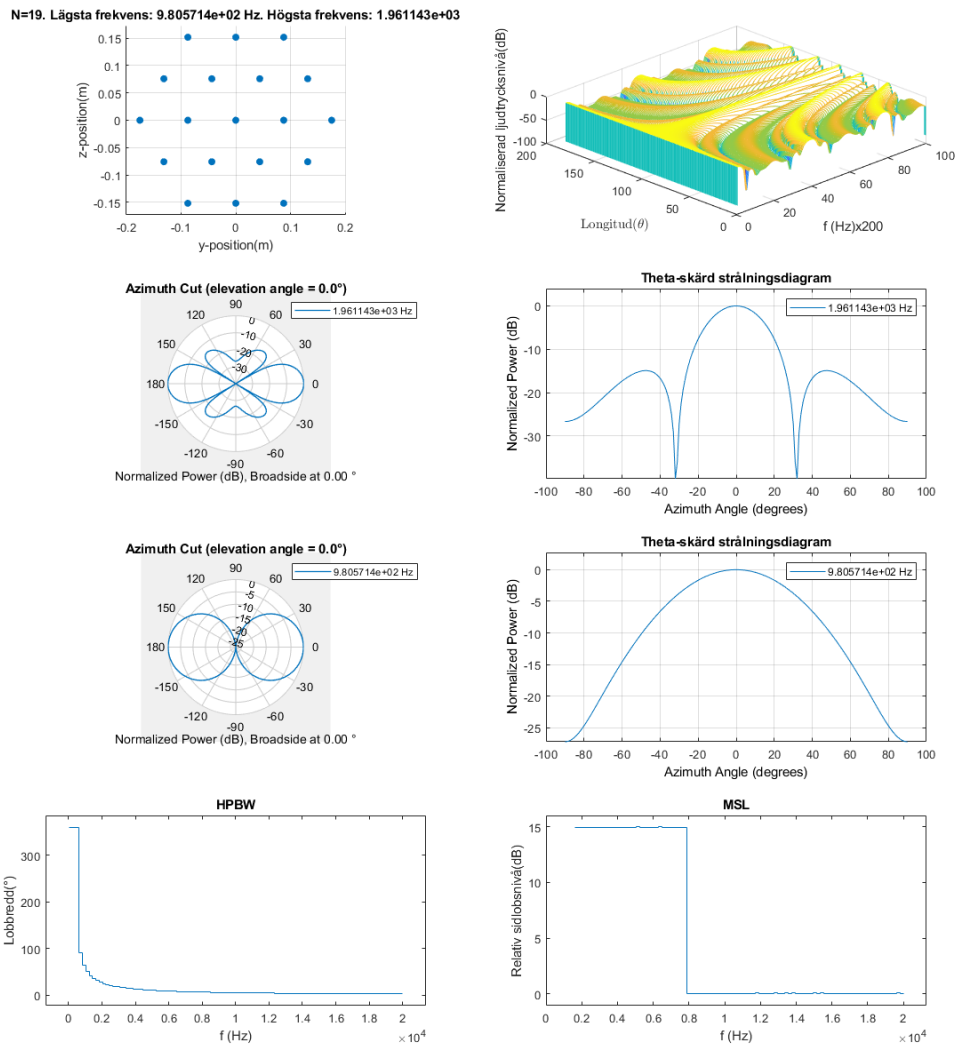
**Figur A.13:** Prestandaparametrarna av 2-sidig UHA med N=7 mikrofoner och radius  $r=0.175\text{m}$ . Ickestyrd.

## A. Simulerade prestandaparametrar



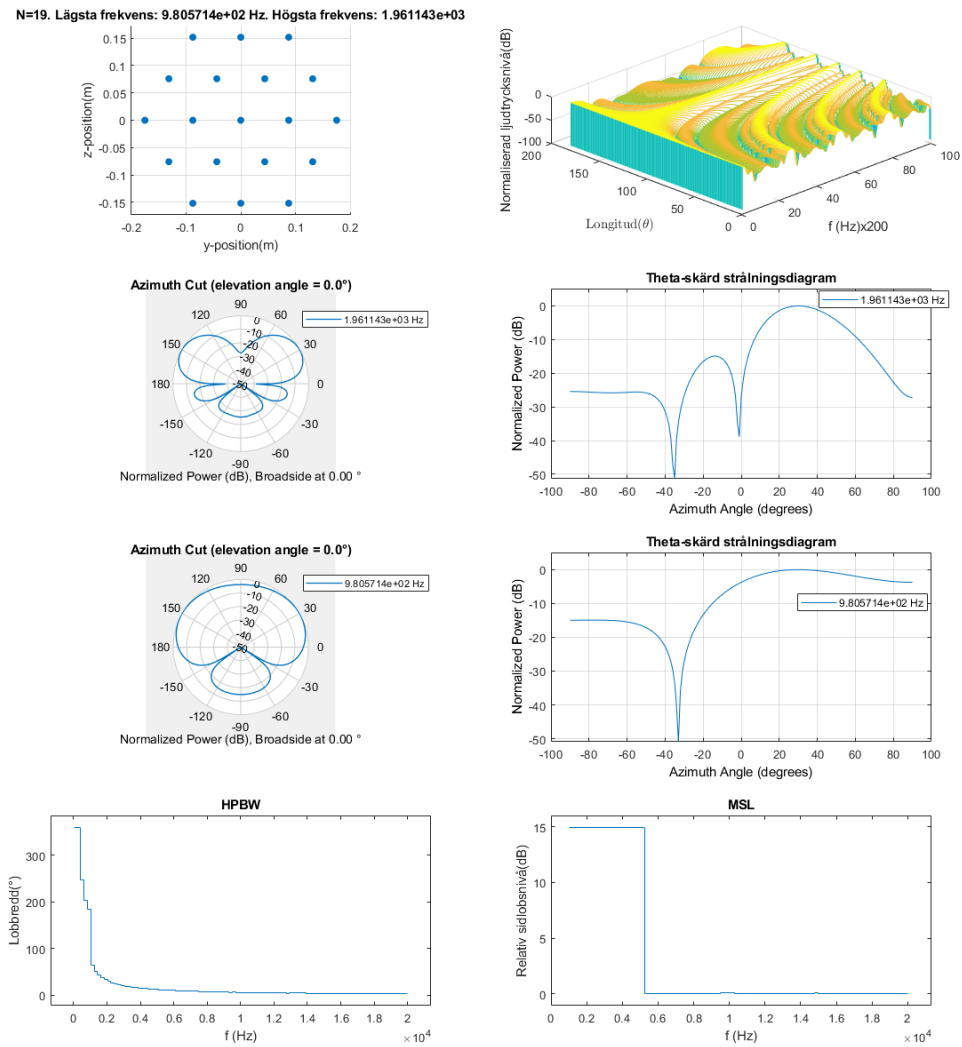
**Figur A.14:** Prestandaparametrarna av 2-sidig UHA med N=7 mikrofoner och radius  $r=0.175\text{m}$ . Styrd  $\varphi = 30^\circ$  och  $\theta = 0^\circ$ .

## A. Simulerade prestandaparametrar

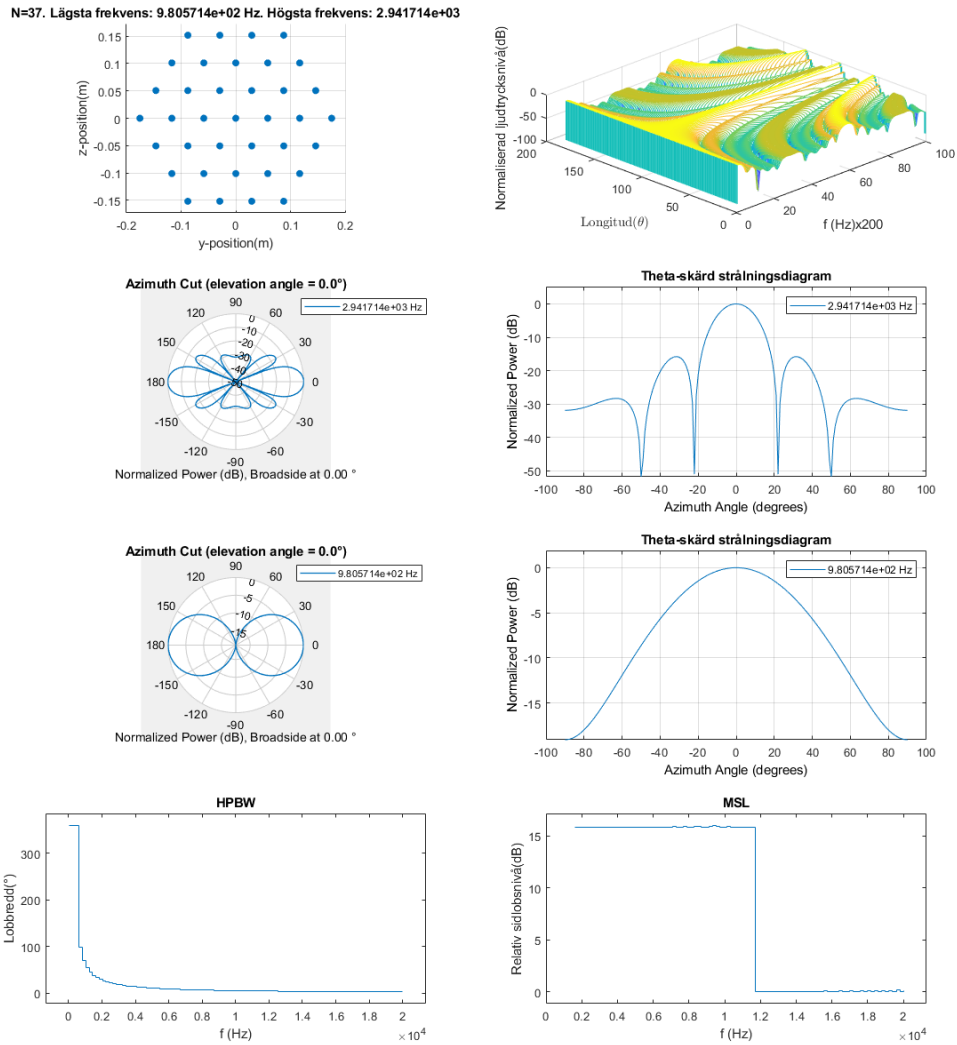


**Figur A.15:** Prestandaparametrarna av 3-sidig UHA med N=19 mikrofoner och radius  $r=0.175\text{m}$ . Ickestyrd.

## A. Simulerade prestandaparametrar

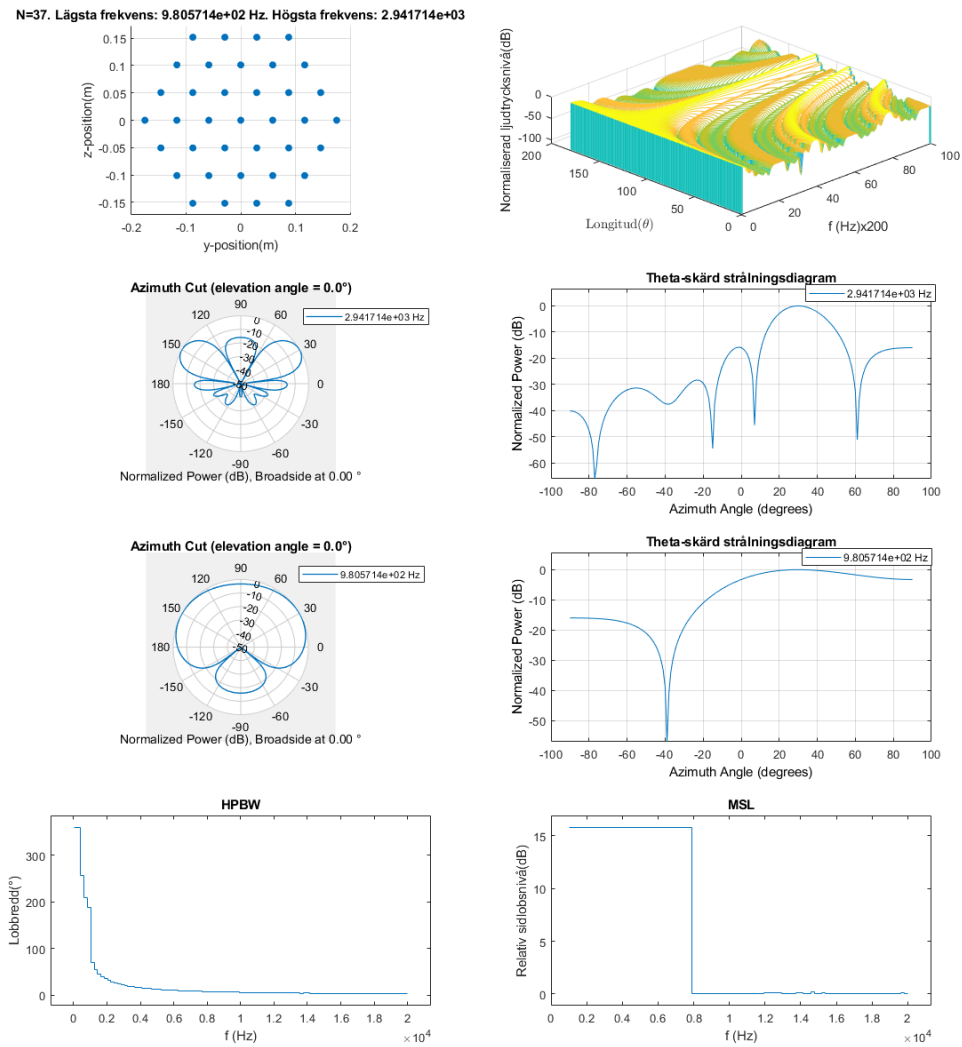


**Figur A.16:** Prestandaparametrarna av 3-sidig UHA med N=19 mikrofoner och radius  $r=0.175\text{m}$ . Styrd till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$ .



Figur A.17: Prestandaparametrarna av 4-sidig UHA med N=37 mikrofoner och radius  $r=0.175\text{m}$ . Ickestyrd.

## A. Simulerade prestandaparametrar



**Figur A.18:** Prestandaparametrarna av 4-sidig UHA med N=37 mikrofoner och radius  $r=0.175\text{m}$ . Styrd till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$ .

## A. Simulerade prestandaparametrar

f(Hz)	HPBW(°)	MSL(dB)
500	88.1400	NaN
1000	40.7000	NaN
1500	26.8200	7.7694
2000	20.0400	7.7728
2500	16.0000	7.7827
3000	13.3200	7.7885
4000	9.9800	1.1123
5000	7.9800	1.1140
6000	6.6600	1.1230
8000	5.0000	0.1951
10000	4.0000	0.2125
12000	3.3400	0.1878

(a) UCA ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	202.5600	NaN
1000	49.2200	7.7687
1500	31.5000	7.7698
2000	23.3400	7.7785
2500	18.5800	1.1140
3000	15.4400	1.1225
4000	11.5500	1.1391
5000	9.2300	0.1868
6000	7.6900	0.1877
8000	5.7700	0.1919
10000	4.6100	0.1968
12000	3.8400	0.0329

(b) UCA styrd

f(Hz)	HPBW(°)	MSL(dB)
500	74.1000	NaN
1000	35.0600	9.5425
1500	23.1800	9.5472
2000	17.3400	0.0002
2500	13.8400	0.0000
3000	11.5400	0.0064
4000	8.6400	0.0004
5000	6.9200	0.0000
6000	5.7600	0.0014
8000	4.3200	0.0016
10000	3.4600	0.0000
12000	2.8800	0.0011

(c) URA ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	191.7600	NaN
1000	41.7900	9.5428
1500	27.1000	0.0008
2000	20.1400	0.0005
2500	14.4200	0.0004
3000	11.6700	0.0031
4000	10.0000	0.0003
5000	7.2000	0.0017
6000	6.6600	0.0042
8000	5.0000	0.0002
10000	3.6000	0.0004
12000	3.3300	0.0011

(d) URA styrd

f(Hz)	HPBW(°)	MSL(dB)
500	98.3400	NaN
1000	44.4600	NaN
1500	29.2200	10.8819
2000	21.8200	10.8846
2500	17.4200	10.8834
3000	14.5000	10.8833
4000	10.8600	0.0004
5000	8.6800	0.0000
6000	7.2400	0.0164
8000	5.4400	0.0010
10000	4.3400	0.0000
12000	3.6200	0.0036

(e) UHA ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	209.7400	NaN
1000	54.4500	10.8818
1500	34.4500	10.8834
2000	25.4600	10.8829
2500	20.2500	10.8814
3000	16.8200	0.0020
4000	12.5800	0.0012
5000	9.0400	0.0011
6000	7.3200	0.0081
8000	6.2800	0.0009
10000	5.0200	0.0044
12000	4.1800	0.0107

(f) UHA styrd

**Tabell A.1:** MSL och HPBW av UCA, URA och UHA ickestyrda och styrda till  $\varphi = 30^\circ$ ,  $\theta = 0^\circ$  med 8, 9 respektive 7 mikrofoner med samma dimensioner som visat i figurerna ovan.

## A. Simulerade prestandaparametrar

f(Hz)	HPBW(°)	MSL(dB)
500	88.1400	NaN
1000	40.7000	NaN
1500	26.8200	7.8994
2000	20.0400	7.9030
2500	16.0000	7.9047
3000	13.3200	7.9064
4000	9.9800	7.9098
5000	7.9800	7.9268
6000	6.6600	3.9047
8000	5.0000	3.9047
10000	4.0000	3.7062
12000	3.3400	3.7078

(a) UCA ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	202.5600	NaN
1000	49.2200	7.8994
1500	31.5000	7.8997
2000	23.3400	7.9041
2500	18.5800	7.8991
3000	15.4400	7.8992
4000	11.5500	3.9077
5000	9.2300	3.9160
6000	7.6900	3.9048
8000	5.7700	3.7265
10000	4.6100	3.4074
12000	3.8400	2.2620

(b) UCA styrd

f(Hz)	HPBW(°)	MSL(dB)
500	83.0000	NaN
1000	38.7000	NaN
1500	25.5200	11.3064
2000	19.0800	11.3063
2500	15.2400	11.3117
3000	12.6800	0.0003
4000	9.5200	0.0031
5000	7.6000	0.0153
6000	6.3400	0.0007
8000	4.7600	0.0125
10000	3.8000	0.0003
12000	3.1800	0.0030

(c) URA ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	198.7200	NaN
1000	46.5200	11.3066
1500	29.9300	11.3033
2000	22.2100	0.0001
2500	17.6800	0.0018
3000	14.7000	0.0009
4000	11.0000	0.0003
5000	8.7900	0.0074
6000	7.3300	0.0006
8000	4.8000	0.0011
10000	4.1100	0.0003
12000	3.6700	0.0003

(d) URA styrd

f(Hz)	HPBW(°)	MSL(dB)
500	124.7800	NaN
1000	52.6000	NaN
1500	34.3600	14.9223
2000	25.6000	14.9249
2500	20.4200	14.9291
3000	17.0000	14.9245
4000	12.7200	14.9296
5000	10.1800	14.9314
6000	8.4800	14.9231
8000	6.3600	0.0012
10000	5.0800	0.0000
12000	4.2400	0.0483

(e) UHA ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	225.4200	NaN
1000	67.3100	14.9222
1500	40.8900	14.9244
2000	30.0200	14.9225
2500	23.8000	14.9403
3000	19.7500	14.9222
4000	14.7500	14.9227
5000	11.7700	14.9235
6000	9.8000	0.0060
8000	7.3500	0.0035
10000	5.3000	0.0033
12000	4.9000	0.0239

(f) UHA styrd

**Tabell A.2:** MSL och HPBW av UCA, URA och UHA ickestyrda och styrda till  $\varphi = 30^\circ$ ,  $\theta = 0^\circ$  med 16, 16 respektive 19 mikrofoner med samma dimensioner som visat i figurerna ovan.

## A. Simulerade prestandaparametrar

f(Hz)	HPBW(°)	MSL(dB)
500	88.1400	NaN
1000	40.7000	NaN
1500	26.8200	7.8994
2000	20.0400	7.9030
2500	16.0000	7.9047
3000	13.3200	7.9064
4000	9.9800	7.9098
5000	7.9800	7.9268
6000	6.6600	7.9727
8000	5.0000	8.0988
10000	4.0000	7.9508
12000	3.3400	8.1163

(a) UCA ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	202.5600	NaN
1000	49.2200	7.8994
1500	31.5000	7.8997
2000	23.3400	7.9041
2500	18.5800	7.8991
3000	15.4400	7.8992
4000	11.5500	7.9034
5000	9.2300	7.9192
6000	7.6900	7.9121
8000	5.7700	7.9825
10000	4.6100	7.5224
12000	3.8400	7.5221

(b) UCA styrd

f(Hz)	HPBW(°)	MSL(dB)
500	92.9000	NaN
1000	42.5000	NaN
1500	27.9800	12.4263
2000	20.8800	12.4281
2500	16.6800	12.4268
3000	13.8800	12.4261
4000	10.4000	12.4255
5000	8.3200	0.0007
6000	6.9400	0.0002
8000	5.2000	0.0385
10000	4.1600	0.0017
12000	3.4800	0.0008

(c) URA ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	205.9800	NaN
1000	51.6800	12.4255
1500	32.9000	12.4277
2000	24.3500	12.4272
2500	19.3800	12.4258
3000	16.1000	12.4265
4000	12.0400	0.0087
5000	9.4400	0.0020
6000	7.3000	0.0002
8000	5.2300	0.0093
10000	4.8000	0.0015
12000	3.6500	0.0008

(d) URA styrd

f(Hz)	HPBW(°)	MSL(dB)
500	141.2800	NaN
1000	56.3000	NaN
1500	36.6600	NaN
2000	27.3000	15.7938
2500	21.7600	15.7933
3000	18.1000	15.7959
4000	13.5600	15.8197
5000	10.8400	15.7959
6000	9.0200	15.8034
8000	6.7800	15.7969
10000	5.4200	15.8226
12000	4.5200	0.0023

(e) UHA ickestyrd

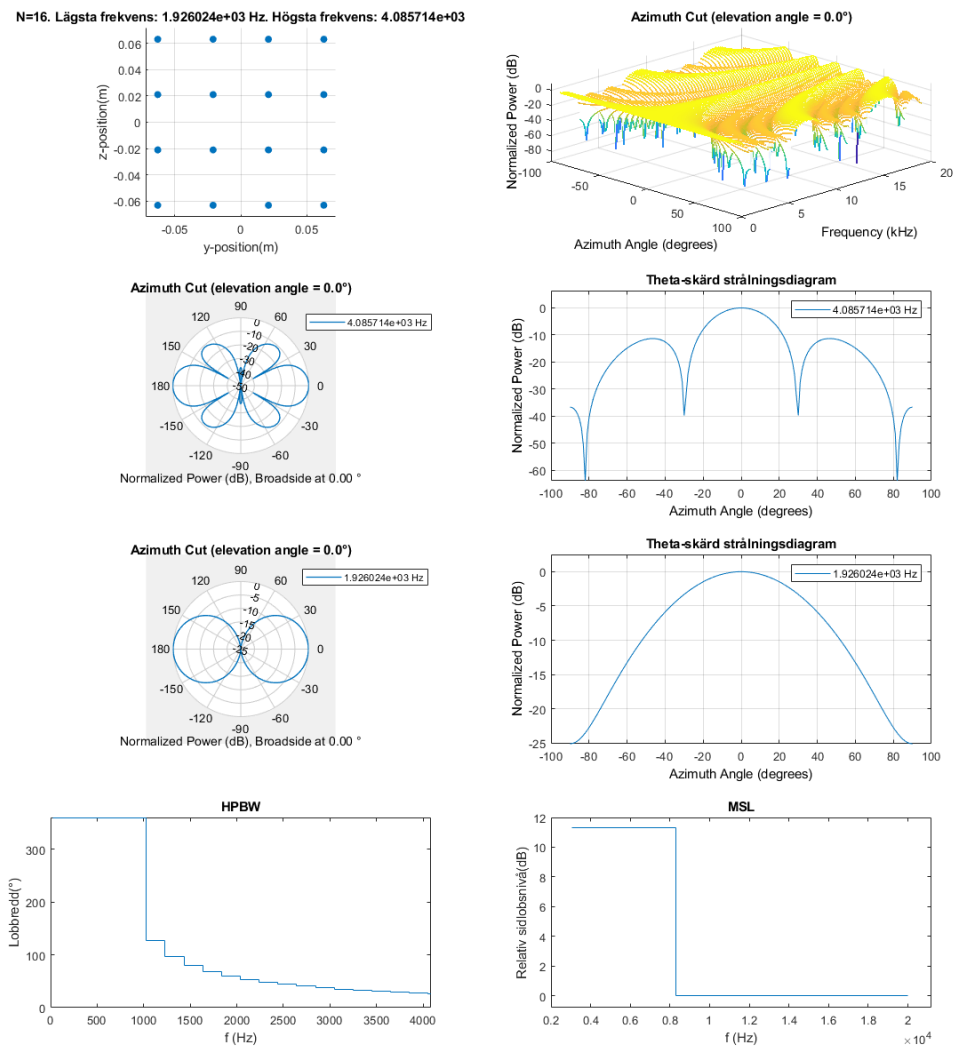
f(Hz)	HPBW(°)	MSL(dB)
500	232.6600	NaN
1000	74.7200	NaN
1500	43.8500	15.7958
2000	32.0700	15.8014
2500	25.4000	15.8057
3000	21.0500	15.8027
4000	15.7100	15.7939
5000	12.5400	15.7959
6000	10.4400	15.7956
8000	7.8200	0.0006
10000	6.2600	0.0148
12000	5.2100	0.0070

(f) UHA styrd

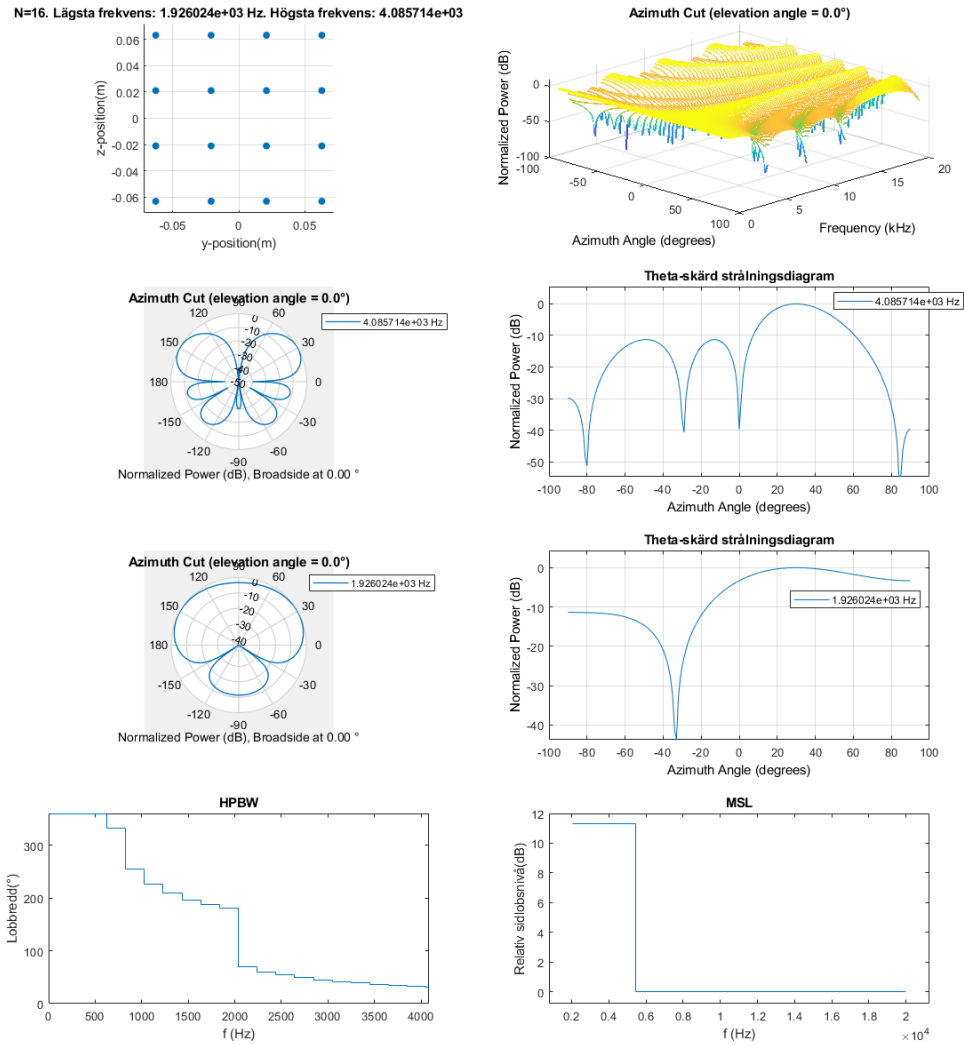
**Tabell A.3:** MSL och HPBW av UCA, URA och UHA ickestyrda och styrda till  $\varphi = 30^\circ$ ,  $\theta = 0^\circ$  med 36, 36 respektive 37 mikrofoner med samma dimensioner som visat i figurerna ovan.

# B

## Simulering av UMA-16 v2 USB



**Figur B.1:** Prestandaparametrarna av 4x4 URA med N=16 mikrofoner och elementavstånd  $r=44$  mm samma som UMA16 v2. Ickestyrd.



Figur B.2: Prestandaparametrarna av 4x4 URA med N=16 mikrofoner och elementavstånd  $r=44$  mm samma som UMA16 v2. Styrdd till  $\varphi = 30^\circ$  och  $\theta = 0^\circ$ .

## B. Simulering av UMA-16 v2 USB

---

f(Hz)	HPBW(°)	MSL(dB)
500	360.0000	NaN
1000	133.9200	NaN
1500	75.6800	NaN
2000	54.8000	NaN
2500	43.2000	NaN
3000	35.7400	11.3033
4000	26.6000	11.3041
5000	21.2200	11.3095
6000	17.6600	11.3047
8000	13.2200	11.3045
10000	10.5600	0.0001
12000	8.8000	0.0141

(a) UMA16 v2 ickestyrd

f(Hz)	HPBW(°)	MSL(dB)
500	360.0000	NaN
1000	229.7000	NaN
1500	193.0400	NaN
2000	71.4800	11.3033
2500	52.6600	11.3041
3000	42.6400	11.3042
4000	31.2400	11.3035
5000	24.7500	11.3066
6000	20.5200	0.0002
8000	15.3200	0.0021
10000	11.1200	0.0001
12000	10.1800	0.0004

(b) UMA16 v2 styrd

**Tabell B.1:** MSL och HPBW av UMA16 v2 ickestyrd och styrd till  $\varphi = 30^\circ$ ,  $\theta = 0^\circ$ .

INSTITUTIONEN FÖR ARKITEKTUR OCH SAMHÄLLSBYGGNADSTEKNIK  
CHALMERS TEKNISKA HÖGSKOLA

Göteborg+

[www.chalmers.se](http://www.chalmers.se)



**CHALMERS**