

הפקולטה להנדסה
המעבדה לעיבוד אותות

מיקוד גלים אקוסטיים על ידי מערך רמקולים

רביד אברהם
לירוי דוכניאן

פרויקט שנה ד' לקראת תואר ראשון בהנדסה

מנחים: ד"ר עופר שוורץ ומר פיני טנדייטניק
מנחה אקדמי: פרופ' שרון גנות

אוקטובר 2020

3	1. תקציר
4	2. תודות
5	3. הצגת הבעיה
6	4. רקע עיוני
6	4.1 הגדרת משתנים ומידול הבעיה
6	4.2 הצגת הפתרון
6	4.2.1 בניית $G(k)$ (מטריצת המרחב) והגדרת חזית הגל
7	4.2.2 שיטת ריבועים מינימליים
7-9	4.2.3 סינון ה DATA בשיטת STFT
10	5. שיטות לפתרון בעיית ה-Beamforming
10	5.1 הסבר על מבנה ופעולת הרמקול
	5.2 שיטה חלופית - הגישה החומרית לפתרון בעיית ההשתקה
10-11	המרחבית
12-13	5.3 הגישה התכנותית -אלגוריתמית
13	6. סימולציה ותוצאות
13	6.1 מבנה מעגלי
13	6.1.1 מבנה בעל 16
17	6.1.2 מבנה בעל 32 רמקולים
20	6.1.3 בחינת השפעת רדיוס המעגל
20	6.2 מבנה לינארי
23	7. סימולציית חדר באמצעות (RIR) room impulse responses
24	7.1 תוצאות סימולציית rir
26	7.2 ניתוח תוצאות סימולציית rir
27	8. מסקנות
28	9. סיכום
29	10. נספח – אלגוריתם
42	11. מקורות

1. תקציר

Beamformer הוא מעבד לביצוע סינון מרחבי. מטרתו היא העברת הסיגנל בכיוונים מסוימים, השתקת אחרים והשתקת הרעש הנלווה תוך שמירה על הכיווניות של האותות. שימושים ל Beamforming קיימים בכל אפליקציה או מודל שבו ישנו יתרון ומשמעות לריכוז האנרגיה בכיוון אחד מסוים על פני פיזוריה בכלל המרחב, דוגמא לכך הינה למשל בשידור סלולרי של גלי RF אשר צריכים להגיע למקלט שעל גבי מכשיר הפלאפון בצורה מדויקת ככל הניתן. מאנטנת MIMO למשל יהיה ניתן לבצע Beamforming דיגיטלי ולאחריו לשר את האות האנלוגי תוך כדי יצירת diversity של השידור והגדלת קיבוליות הערוץ ויציבות הסיגנל.

מטרת פרויקט זה היא תכנון והוכחת היתכנות למערך רמקולים אשר יאפשר לרכז אלומת שמע לכיוון רצוי ולהדעיק את השמע בכיוונים לא רצויים. במהלך הפרויקט השקענו זמן רב בללמוד על הפרמטרים המשפיעים במערכת שכזו ועל כן האלגוריתם שיצרנו נתן לנו מושג על מידת ההשפעה של כל פרמטר וכך הוביל לחקר ביצועים מעמיק לשם אופטימיזציה של המערכת. לכתובת האלגוריתם השתמשנו בידע וכלים מעולם עיבוד האותות שניתן לנו במהלך מספר קורסים בתואר הראשון וכמובן במעבדה המעשית, מלבד אלה השתמשנו רבות במאמרים נוספים בתחום ובפרוייקטים דומים שנעשו בעולם האקדמי, על זאת ועוד על פעולת האלגוריתם נרחיב בהמשך. בפרק הניסויים התמקדנו בהצגת היתכנות של הכוונת אלומת שמע ממערך רמקולים תוך שימוש בסימולציה ממוחשבת. מגוון מקרים אשר ממדלים את העולם האמיתי ככל שניתן נבדקו כדי להגביר את רמת האמינות של הסימולציה שלנו.

2. תודות

ברצוננו להביע את תודתנו לכל העוזרים והמסייעים במהלך הכנת הפרויקט. לעופר שורץ על ההנחיה הצמודה, הדרבון, השאפתנות, הישיבה ביחד כתף אל כתף ועל כל הליווי במהלך הפרויקט בכיף ובשמחה.

לפרופ' שרון גנות על הליווי האקדמי, על העצות המחכימות, על ההכוונה, על הניסיון הרב, ועל שתמיד קיבלת אותנו במאור פנים ובחיוך.

לפיני טנדייטניק על הסיוע הרב – בהארת הדרך, בהסברים המלומדים, בהתעמקות ועל כל התמיכה לאורך הפרויקט.

תודה מיוחדת נתונה למשפחותינו על התמיכה לאורך כל התואר, על הפרגון והאכפתיות, על ההקשבה, ההבנה והסיוע בכל הדרוש.

3. הצגת הבעיה

פרויקט זה עוסק בבעיה של הכוונת אותות שמע בעזרת טכניקת עיצוב אלומה (Beamforming) דרך מערך רמקולים.

Beamforming או Spatial Filtering (עיצוב אלומה) הינה טכניקה מוכרת בתחום עיבוד האותות המשמשת לשידור או קליטה זוויתית. התופעה הזו מושגת על ידי שילוב אלמנטים במערך שבו כל אחד מהאלמנטים בעל מופע (פאזה) שונה, בצורה זו אותות בזוויות מסוימות יחוו התאבכות בונה ואותות בזוויות אחרות יחוו התאבכות הורסת.

בשנים האחרונות מודלים של בקרה על כיווניות של אותות אומצו ע"י אפליקציות ומערכות שמע רבות.

כאשר עוברים לעסוק באותות קול, טכניקת עיצוב האלומה יעילה בחילוץ והפרדה של מקורות קול בחדר, למשל הפרדת מספר דוברים בבעיית מסיבת הקוקטייל המוכרת (cocktail party problem). מיקום הרמקולים מראש.

לעומת אותות תקשורת מבוססי גל נושא, באותות שמע יש טווח רחב של תדרים שעשויים להופיע לכן יש יתרון בהפרדת התדרים לפי פסי תדר מראש, לפני שלב עיצוב אלומת אות השמע. זאת מכיוון שלתדרים שונים יש ' מסנן מעצב אלומה אופטימלי' (optimal beamform filter) שונה. לכן הגישה המוצעת היא לטפל בכל פס תדר בנפרד באופן אופטימלי ולאחר מכן לבצע סינתזה ולחבר את כלל המוצאים מכל מסנן.

כיום, כאשר מפיקים אלומת שמע מרמקולים בחדר או בחלל מסוים, האלומה מתפשטת בכל המרחב ללא בקרה או שליטה שלנו. בעזרת השיטה שלנו ניתן לשלוט על כיוון ההתפשטות של האלומה.

האלגוריתם שיצרנו לוקח בחשבון מספר משתנים של הבעיה:

- מספר רמקולים
- אופן סידור הרמקולים במרחב
- מאפייני הרמקול
- מאפיינים של החלל בו המערכת נמצאת (מקדם הדהוד, החזרה וכו')

במהלך הפרויקט חקרנו כיצד כל אחד מהמאפיינים הללו משפיע על הכוונת אלומת השמע ויצירת אזורים מושתקים ואזורים לא מושתקים.

4.2.2 שיטת ריבועים מינימליים

לצורך הדוגמה נניח $f(k)$ אידיאלי כלהלן –

$$f(k) = \begin{pmatrix} \cdot \\ \cdot \\ 0 \\ 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \end{pmatrix}$$

נפתור בשיטת ריבועים מינימליים על ידי הדרישה הבאה –

$$\text{minimize } \|G(k)h(k) - f(k)\|^2$$

כלומר, אנחנו מחפשים $h(k)$ כך שמוצא המערכת שלנו יהיה קרוב ככל שניתן ל- $f(k)$ האידיאלי, דהיינו שגיאה מינימלית במובן ריבועים מינימליים. הנ"ל מגדיר לנו פתרון סגור לבעיית הריבועים מינימליים הנתון על ידי:

$$h = (\bar{G}^T \bar{G})^{-1} \bar{G}^T f$$

4.2.3 סינון ה DATA בשיטת STFT (short time Fourier transform)

מטרת ה-STFT היא לבצע אנליזה ספקטרלית לאות בעל הרכב תדרי משתנה בזמן באמצעות חלוקה של האות למקטעים והתמרת כל מקטע בנפרד.

התמרת פורייה לזמן קצר מוגדרת על ידי:

$$X_{STFT}(e^{j\omega}, n) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega m}$$

$x[n]$ - הוא אות אינסופי בזמן בדיד.

$w[n]$ - היא פונקציית חלון כלשהיא (הנקראת **חלון אנליזה**, לאו דווקא חלון מלבני) בעלת אורך L_h .

המשמעות של ההתמרה היא שבכל פעם לוקחים חלק מהאות (שנוצר על ידי ההכפלה בחלון)

ומבצעים לו DTFT. ככל שנקדם את n , נקבל חלק חדש של האות.

ככל שהחלון גדול יותר - מקבלים רזולוציית תדר טובה יותר של האות, משום שההתמרה של החלון

קרובה יותר להלם (ומקבלים DTFT מלא של האות). ככל שהחלון יותר קטן הרזולוציה נמוכה יותר,

אך נוכל לזהות שינויים בהרכב התדרי של האות לאורך הזמן.

לא נעבוד כאמור עם התמרת ה DTFT הרציפה אלא עם דגימות שלה במרווחים של $\frac{2\pi}{M}k$, ונקבל התמרת DFT עבור כל חלק של האות שמוכפל בחלון אנליזה $w[n]$:

$$X_{STFT}[n, k] = X_{STFT}(e^{j\omega}, n) \Big|_{\omega=\frac{2\pi}{M}k} = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\frac{2\pi}{M}kn} \quad k = 0, 1, \dots, M-1$$

שלבי חישוב ה STFT :

1. $x[n]$ - אות בדיד שנרצה לבצע עליו את ה STFT.

2. $h[n]$ - חלון האנליזה של ה STFT.

3. L_h - אורך חלון האנליזה.

4. M - אורך ה DFT.

5. R - גודל החפיפה בין המקטעים.

6. **שלב א:** עבור כל אינדקס n מכפילים את האות $x[n+m]$ בחלון $h[-m]$ הכפולה איבר איבר .

$$x[n] \rightarrow \begin{array}{|c|c|c|c|} \hline x[n-(L_h-1)] & \dots & x[n-1] & x[n] \\ \hline h[L_h-1] & \dots & h[1] & h[0] \\ \hline \end{array}$$

נשים לב שעלינו להפוך את החלון. האות $x[n+m]$ נכנס דגימה אחרי דגימה. מקבלים חוצץ באורך L_h .

שלב ב: נניח כי $L_h > M$ וכפולה שלמה שלו (אחרת יש להוסיף אפסים). על מנת לחשב את

$s_\ell[n]$ עלינו לקפל את החוצץ על עצמו $\frac{L_h}{M}$ פעמים (המספרים מציינים אינדקסים):

$$s_\ell[n] = \sum_{r=-\infty}^{\infty} x[n+(rM+\ell)]h[-(rM+\ell)] \quad \ell = 0, 1, \dots, M-1$$

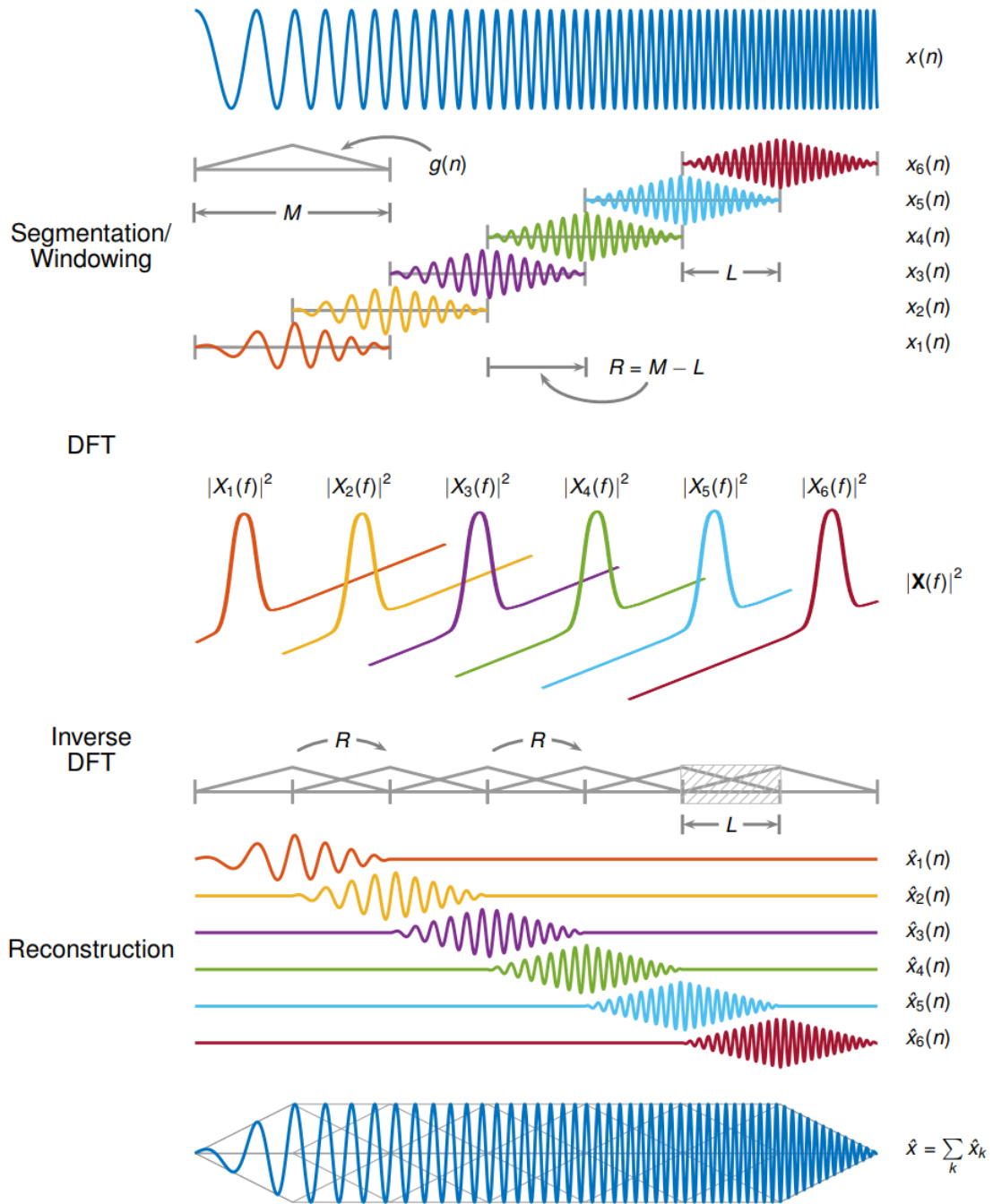
$$= \begin{array}{|c|c|c|c|} \hline M-1 & \dots & 1 & 0 \\ \hline 2M-1 & \dots & M+1 & M \\ \hline \vdots & & & \\ \hline L_h-1 & \dots & (\frac{L_h}{M}-1)M+1 & (\frac{L_h}{M}-1)M \\ \hline \end{array} \begin{array}{l} + \\ + \\ + \\ + \end{array} = \begin{array}{|c|c|c|c|} \hline s_\ell[M-1] & \dots & s_\ell[1] & s_\ell[0] \\ \hline \end{array}$$

שלב ג: נבצע DFT באורך M לדגימות $s_\ell[n]$ שקיבלנו מהקיפול, לקבלת $Z_k[n]$.

שלב ד: נכפיל את הדגימות באקספוננט, $e^{-j\frac{2\pi}{M}kn}$ על מנת לבצע מעבר לפס בסיס.

בשלב הסינתזה נבצע את התהליך ההפוך.

נמחיש את תהליך ה-STFT באיור :



5. שיטות לפתרון בעיית ה-BeamForming

לפתרון הבעיה אותה אנו מציגים בפרוייקט זה, יש מספר גישות, המכנה המשותף של הגישות הוא ההסתכלות על המערכת בתור אפקט פיזיקלי, מידע עובר עיבוד מסויים ואז משודר ע"י חומרה מסויימת. כל גישה לוקחת חלק אחר מחלקי המערכת המוזכרים ומשתמשת באופן הפעולה שלו על מנת ליצור אלומה מוכוונת.

ישנן גישות מתמטיות שמבוססות על אלגוריתמים לעיבוד אותות קול ושיטות מתמטיות כדי לייצר בקרה על הסיגנלים והתפשטותם במרחב ולעומת זאת ישנן גישות חומרתיות שמבוססות על פיזיקה שונה מאשר החומרה הרגילה שמשמשים בה ביום יום, כאמור החומרה העיקרית בפתרון בעיה זו היא רמקולים.

5.1 הסבר על מבנה ופעולת הרמקול

רמקול בסיסי מורכב ממגנט ומסליל של מוליך חשמלי החופשי לנוע בתוך השדה המגנטי הקבוע שמייצר המגנט. אל הסליל מחוברת דיאפראגמה שהיא למעשה ממברנה דקה בצורה המזכירה חרוט רחב בסיסי. שולי הממברנה מחוברים אל מסגרת מתכתית בחיבור גמיש שאיננו מפריע לתנועתה. המגנט הקבוע מחובר אל המסגרת המתכתית בחיבור קשיח. כאשר זרם חשמלי עובר בסליל, נוצר שדה אלקטרומגנטי מהסליל שעוצמתו וכיוונו יחסיים לעומת הזרם החשמלי ולכיוונו. השדה גורם לתנועה של הסליל ביחס למגנט הקבוע, התנועה הזו נגרמת מהכוחות הפועלים בין השדה המגנטי הקבוע והשדה המגנטי המושרה מתוך הסליל. תנועת הסליל גורמת לכך שהממברנה המחוברת אליו תנוע, וזו דוחפת את האוויר שמסביבה בכיוון ובעוצמה יחסיים לכיוונו ולעוצמתו של הזרם החשמלי העובר בסליל. מכאן כאשר הזרם החשמלי העובר בסליל מתנדד, הוא יגרום לתנודה של הממברנה בקצב ובעוצמה התואמים לקצב ועוצמת תנודות הזרם, ולהרעדת האוויר סביב הממברנה. תנודות האוויר המופקות באופן זה הן, בעצם, גלי קול או צלילים.

אולטרה סאונד:

אולטרה סאונד (על שמע) הוא קול בתדירות גבוהה מזו שאדם יכול לשמוע (תחום השמע של האדם הוא מ-20 הרץ עד 20 קילו הרץ). ישנם בעלי חיים שיכולים לשמוע תדירויות גבוהות יותר מהאדם ולפיכך ביכולתם לשמוע אולטרה סאונד בתדירויות מסוימות.

5.2 שיטה חלופית - הגישה החומרתית לפתרון בעיית ההשתקה המרחבית

הגישה החומרתית לא מתעסקת בdata שמשודר ובוקע מהרמקולים, אלא באה להחליף את טכנולוגיית הרמקולים הקיימת לטכנולוגיה שיעודה הוא ליצור השתקה מרחבית. ע"פ גישה זו ניתן "לשדרג" את הרמקולים הקיימים היום והפועלים כפי שהוסבר לעיל, בכך שנוסיף להם גורם משדר נוסף -משדר אולטרה סאונד באמצעות משדר זה וביכולות בקרה ושליטה על הרמקול נוכל לגרום לכך שהסאונד הבוקע מהרמקול ישדר בתדרים גבוהים מאלו שהאדם מסוגל לשמוע ובכך לגרום לתחושה של השתקה מרחבית, בפועל הסיגנל קיים והוא מתפשט כרגיל במרחב

אך מאזין שיעמוד בנקודות בחדר בהן האותות הם אולטרה סאונדיים לא ישמע דבר בגלל מגבלות המקלט הביולוגי שלו (כפי שהוזכר האוזן האנושית מסוגלת לשמוע עד 20 קילו הרץ).

דוגמא ליישום שיטה זו ניתן לראות במוצר קיים בשם "Acouspade™ directional speaker" של חברת ULTRASONIC. המוצר הוא למעשה רמקול עם יכולות הכוונה והשתקה של שמע במרחב. [1] הטכנולוגיה עלייה המוצר מתבסס נקראת מערך פרמטרי [2,3] בעולם הסאונד והאקוסטיקה. מערך פרמטרי יודע לייצר אות צר סרט בתדר נמוך על ידי ריבוב של אותות בעלי תדר גבוה. טכנולוגיה זו מחייבת שהאותות הנשמעים לאוזן האנושית מהרמקולים המיוחדים האלו הם למעשה תוצר של הפרש של אותות שאינם יכולים להשמע, לדוגמא [3] מערכת בעלת טכנולוגיה כזו תכיל צמד רמקולים, רמקול אחד משדר בתדר של [kHz] 200 רמקול שני ישדר בתדר של [kHz] 201 כמובן שאותות אלו אינם נשמעים לאוזן האנושית אך ההפרש ביניהם [kHz] 1 כן. שרשור של צמדים כאלו לפי פריסה של המרחב לזוויות נותן לנו שליטה ובקרה על השמע.

יתרונות השיטה:

- 1-דיוק – שידור גלי קול אולטרה סאונד הוא למעשה מדויק יותר מאלגוריתם תוכנתי היות והוא מסתמך על מגבלות האוזן האנושית, אם נשדר בתדר שהאוזן לא יודעת לקלוט -השתקנו את המרחב.
- 2-סיבוכיות נמוכה לשידור real time – היות והפתרון הזה לא מעבד את הסיגנל שמשדרים, אין צורך באלגוריתמים כבדים שצריכים לדעת לטפל במידע שזורם למערכת בזמן אמת, הסיגנלים המשודרים הם אותם סיגנלים שהיינו משדרים ברמקול רגיל – החומרה היא זו שעושה את ההבדל ו"דואגת" שההשתקה המרחבית תתבצע.

החסרונות של השיטה הזו הם :

- 1-הבעיה לא באמת נפתרת מהשורש – האותות הבוקעים מן הרמקולים מתפשטים בכל המרחב, הפתרון החומרתי משתמש במגבלות המקלט (האוזן האנושית) על מנת ליצור תחושת השתקה, אם ננסה למדוד את הסיגנלים הם עדיין יהיו בכל המרחב ובעוצמה חזקה, אבל לא נצליח לשמוע כלום.
- 2-עלות – משדר אולטרה סאונד עלול לייקר את המערכת מאחר ומדובר בתוספות חומרה ואינטגרציה שלהן לטכנולוגיה קיימת שידור אולטרה סאונד מרמקולים הוא לא דבר נפוץ.
- 3-שינוי חומרה קיימת וטכנולוגיה קיימת – פתרון חומרתי לבעיה אוטומטית דורש מאמץ גדול יותר, פתרון תוכנתי-אלגוריתמי עשוי לחסוך את ההסבה של החומרה הקיימת לא רק בעלות אלא גם מבחינת זה שניתן להימנע מלהגביל את יכולות הרמקולים כדי לממש אפליקציה מסוימת. בפתרון תוכנתי האילוצים החומרתיים הם לא רבים ולכן הדרישות למימוש סבירות יותר.

4-סכנות בריאותיות- שימוש בגלי אולטרה סאונד בקרבת רקמות אנושיות ידוע כגורם שעלול לפגוע בהן, כמו כן עלול לפגוע גם באוזניים, לא נכנס לסיבות הביולוגיות אך נציין שתופעות אלו תלויות בפרמטרים של עוצמה, זמן חשיפה ותדר.

5.3 הגישה התכנותית-אלגוריתמית

בגישה שבה אנחנו משתמשים בעבודה זו אנחנו מתבססים על חומרה קיימת (רמקולים פשוטים) ולא נדרשים להוסיף רכיבים מיוחדים שמשנים את פעולת הרמקולים. למעשה הפתרון שלנו עובד על dataן ולא על הפיסיקה של התפשטות הגלים. אנחנו מחשבים את ההשהיות שחוזה כל סיגל בדרך למאזן והאלגוריתם אחראי בין היתר "ליישר קו" בין הסיגלים ובכך לגרום לתופעות התאבכות. נפרט על שלבי הפתרון בגישה זו:

בגישה זו אנו פורסים מערך רמקולים בסידור מעגלי (ניתן לסדר באופן שונה אך נתמקד בסידור המעגלי) הפריסה הזו מחלקת לנו את החלל שבו המערכת נמצאת לאזורים, בין כל שני רמקולים יש זווית שמגדירה את האזור. כך אנחנו מגדירים כל אזור לפי הזווית שלו ביחס לראשית צירים שנקבעת מראש.

הגדרת המרחב ע"י הזוויות שבין הרמקולים מסייעת לנו לקבוע באלגוריתם אילו אזורים יושתקו ולא יושתקו.

בשלב הבא אנחנו משתמשים בהגדרות הגיאומטריות של מערך הרמקולים (הזוויות והמיקומים במרחב) כדי לקבוע את המיקומים שבהן חזיתות הגלים מכל אחד מהרמקולים נפגשים. כעת, בהינתן אזור במרחב שברצוננו לא להשתיק אנחנו מחשבים את ההשהיות בהגעה של כל חזית גל מכל רמקול לאזור הרצוי. חישוב ההשהיות נועד ליצירת מסננים אופטימליים שבסוף יגרמו להתאבכויות הורסות בכל מקום מלבד האזור הרצוי.

ע"י שיטת ריבועים מינימליים שהוצגה בהרחבה בחלקים הקודמים אנחנו מחשבים את המסנן שיקרב אותנו למסנן אידיאלי שעל ידו כל האזורים בחלל מושתקים מלבד אזור אחד, בשיטת ריבועים מינימליים אנחנו מחשבים את המרחק בין המסנן הרצוי למסנן המצוי ומנסים להביא אותו למינימום.

חסרונות:

1-חוסר אידיאליות\חוסר דיוק אם מספר הרמקולים נמוך – כפי שנציג בתוצאות הסימולציה בהמשך, מדובר בפתרון תוכנתי אלגוריתמי שלא משתיק לגמרי את האזורים הרצויים להשתקה, תנאי הסביבה עלולים להשפיע מאוד על התוצאות אם זה התווך, ההדהוד שבמרחב ומאפייני הרמקולים.

2-תלות בגיאומטריה של המערכת (פריסת הרמקולים) - הפתרון שאנו מציגים בסימולציות תלוי באופן חזק בתצורת המערכת, לפריסה מעגלית יש חישוב שונה של ההשהיות הסיגלים מפריסה לינארית, ופריסות שונות ואחרות ידרשו התייחסות שונה גם כן.

יתרונות:

1-ההשתקה טוטאלית – הבעיה כפי שהוצגה הציפה מטרה להשתיק אזורים בחלל מסוים, האלגוריתם שלנו יוצר תופעות התאבכות בין הסיגנלים, אם ננסה למדוד המצאות של סיגנלים בחדר הפעלת האלגוריתם נצפה שלא נמצא כאלה כי סיגנל אחד התאבך עם השני (התאבכות הורסת), הפתרון אינו מסתמך על מגבלות המקלט (האוזן האנושית).

2-עלות – הפתרון התוכנתי לא דורש חומרה יקרה שבדרך כוללת גם תמיכה רחבה במקרים של תקלות או בלאי.

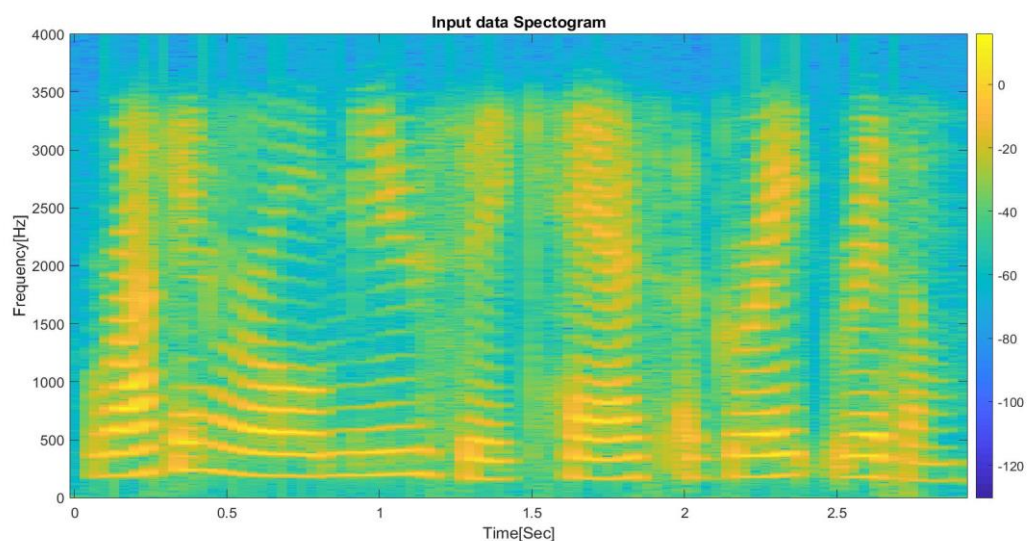
3-פתרון גנרי- כשהפתרון מבוסס על תוכנה ועל קוד ניתן לשנות אותו בקלות ולהתאים לתנאי סביבה שונים או למערכי רמקולים שונים.

6. סימולציה ותוצאות

כאמור , בנינו אלגוריתם שמטרתו לסנן את ה data באמצעות שיטת הSTFT. בחרנו לבצע סימולציה של שני מבנים שונים של מערכי רמקולים, האחד מבנה מעגלי ואילו השני מבנה לינארי. את שני המבנים בחנו במספר מקרים שונים על מנת לאמוד את איכות ה Beam Forming .

הסימולציה אותה בנינו מאפשרת למשתמש לבחור מגוון רחב של פרמטרים שמקנים גנריות למודל. הפרמטרים אותה ניתן לקבוע לשם בחינה הינם: כיוון שמע רצוי (במעלות), מספר הרמקולים, מספר הגזרות המרחביות המחלקות את המרחב, מספר דגימות להתמרה, גאומטריה של המערך, רדיוס או מרחק בין הרמקולים.

בתור data בחרנו שיר שהתמרת הSTFT שלו הינה



הערה: הסקלה של הגרף לעיל הינה dB .

כמו כן , בחרנו כי כיוון השמע הרצוי נבחר להיות 45° , מספר הגזרות המרחביות הינו 24 , מספר הדגימות לביצוע התרמת STFT הינו 1024 . להלן הגדרת המשתנים כפי שהיא מוגדרת באלגוריתם :

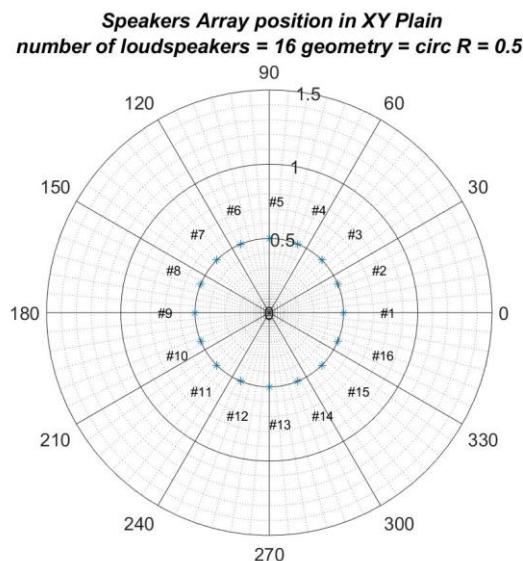
```
%% definition of the problem variables and constants
close all
clc

fs = 16e3;
c = 343; % Sound velocity (m/s)
N = 32; %number of array loudspeakers
Q = 24; %number of grid lines in dividing the azimuth angle theta [0,2pi]
Number_of_Radiuses_to_sweep = 50; %Number Radiuses to sweep of the speakers array
Num_of_freq_point = 1024; %number of frequency points
k = 0 : 1 : Num_of_freq_point-1;
direction_q_selected = 4; %select the azimuth of hearing
direction_to_plot = 14; %select the azimuth of testing
speaker_to_plot = 4; %select the speaker of interest
geometry = 'circ';
```

6.1 מבנה מעגלי

6.1.1 מבנה בעל 16 רמקולים

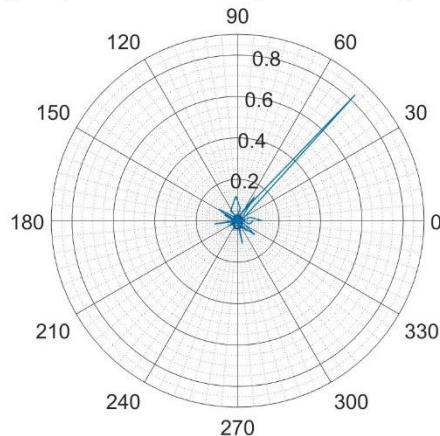
נציג את מיקומי הרמקולים במרחב אשר בחרנו בגרף הפולארי הבא לו צירים של רדיוס וזווית במעלות. ניתן לראות כי המקרה הנוכחי הינו סימולציה של מערך בעל רדיוס של 0.5 מטר.



על מנת לבחון את מערכי הרמקולים השונים בחרנו במגוון מדדים שונים שיעזרו לנו לנתח את ביצועי המערך.

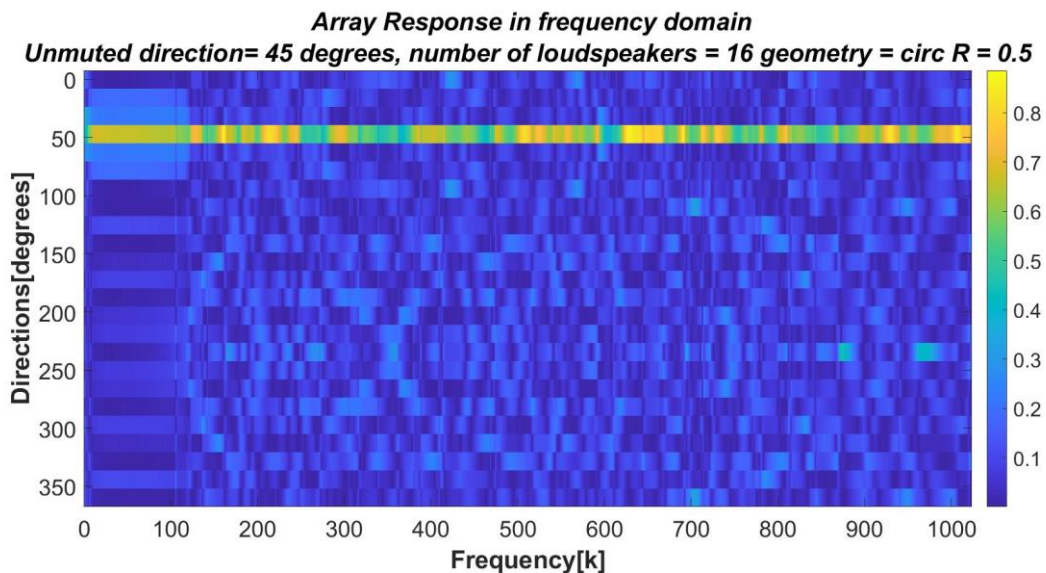
הראשון , הינו גרף פולארי של תגובת המערך המתבסס על ההנחה שמערך הרמקולים שלנו הוא מקור נקודתי (קירוב שדה רחוק) . הגרף הבא מציג את תגובת המערך לכל הכיוונים ואת מידת העברת השמע בסקאלה לינארית על גבי תדר דיגיטלי בודד אותו בחרנו.

Array Response in frequency domain for all direction
Unmuted direction = 45 degrees, number of loudspeakers = 16 geometry = circ R = 0.5 Bin = 640



אלומת השמע כמעט ומושקת לכלל הכיוונים מלבד הכיוון הנבחר. עוד נשים לב כי ישנה העברה של פחות מ-1 לכיוון הרצוי מה שמראה על חוסר אידאליות המבנה אך יחס העברה לכיוון הרצוי עם שאר הכיוונים הוא בקירוב 1:9 (בהתאם) בסקאלה לינארית.

שנית, הגרף הבא מציג תגובת מערך הרמקולים לכל תדר ולכל כיוון במרחב על מנת לבחון לעומק יותר את ביצועי המבנה.

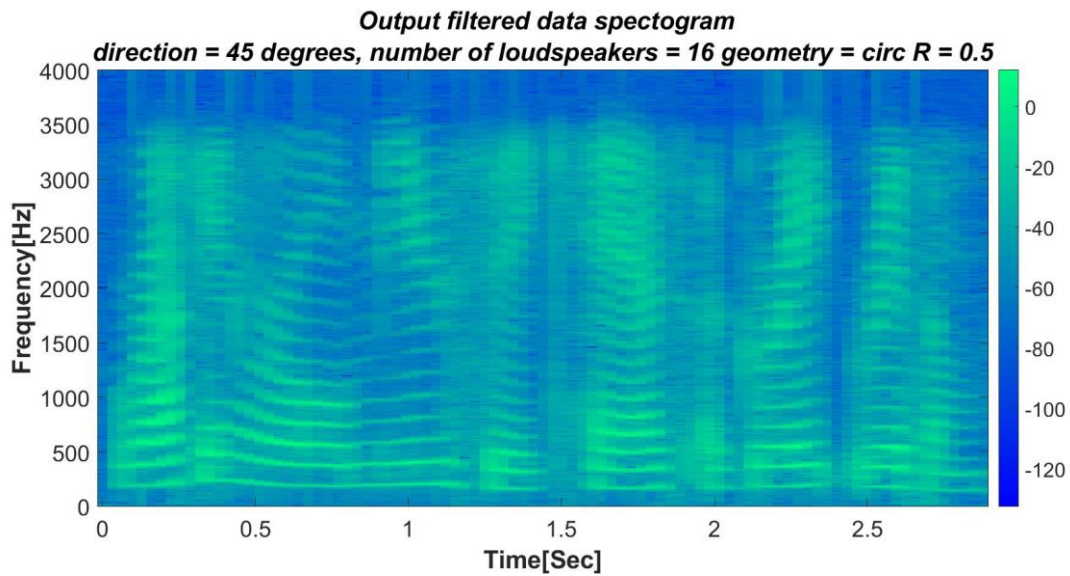


ניתן לראות כי בתדרים נמוכים מאוד המערך עדיין אינו מבצע השתקה ומנחית את השמע לכלל הכיוונים. לאחר מכן ישנה חוסר אחידות ביכולת של המערך להשתיק את הכיוונים הרצויים באותה מידה וניתן להבחין כי בתדרים מסוימים ישנה יכולת ביצוע טובה יותר מהאחרים. כך למשל עבור התדר הדיגיטלי $k = 650$ המערכת מעבירה בצורה טובה יותר את הסיגנל לכיוון 45 מעלות מאשר בתדר הדיגיטלי $k = 380$. עוד נאמר כי בהסתכלות כללית כן ישנה העברה משמעותית לכיוון 45

מעלות ביחס לשאר הכיוונים במרחב. לגבי ההשתקה ה"פגומה" בתדרים הנמוכים אנחנו מסיקים שזה נובע מכך שבתדר נמוך האורך גל גדול יותר מבתדר גבוה על כן האות עצמו הוא כמעט ישר (בקירוב DC) ולכן אין משמעות להזזה בזמן, שעליה האלגוריתם מבוסס ולכן ההשתקה נפגמת.

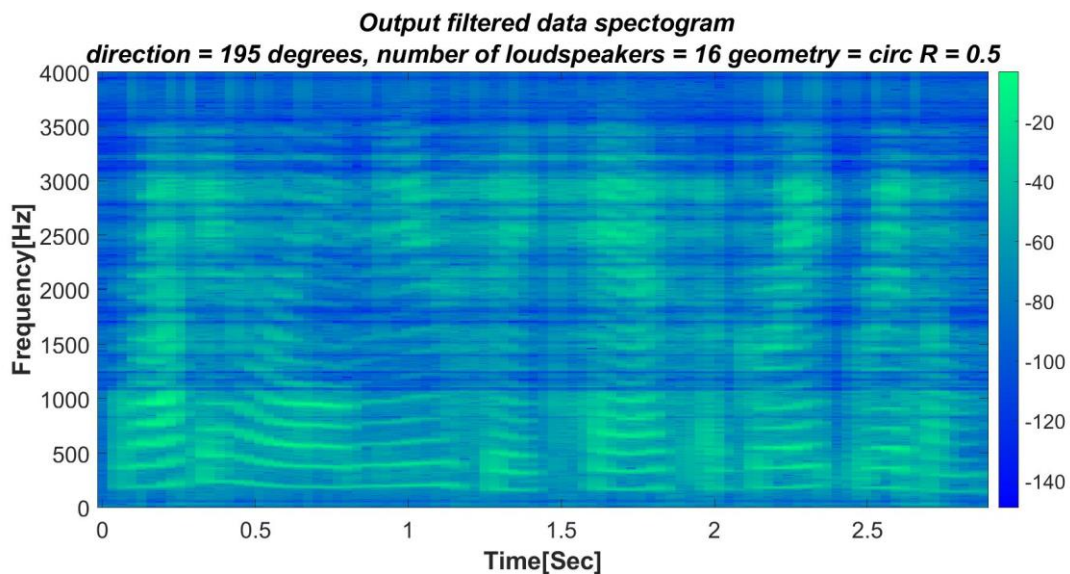
כעת בחנו את המבנה על ידי הכנסת אות שמע כמתואר בתחילת פרק הסימולציה ובחנו את התמרת STFT של המוצא לשני כיוונים שונים, האחד כיוון אותו נרצה להעביר והשני כיוון מושתק של 195° .

בכיוון 45° ,



התמרת STFT בכיוון הרצוי לשמע נשמרת ודומה למדי להתמרת הdata בכניסה,

בכיוון של 195° .

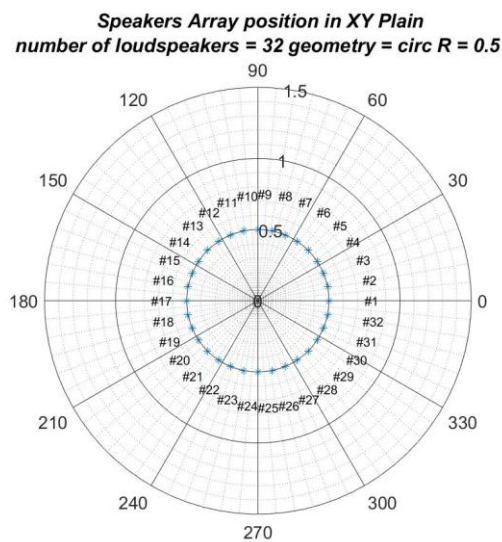


מעניין לראות כיצד פסי התדר האנלוגיים המונחתים תואמים לפסי התדר הדיגיטליים המונחתים כפי שמוצג לפני שני גרפים, בגרף תגובת המערך לכל פסי התדר ולכל כיוון. כמוכן שפסי התדר האנלוגיים המתאימים לתדר הדיגיטלי קשורים בזה על ידי תדר הדגימה לפי הקשר הבא:

$$\omega_{analog} = \omega_{sample} \cdot \frac{k}{N(= 1024)}$$

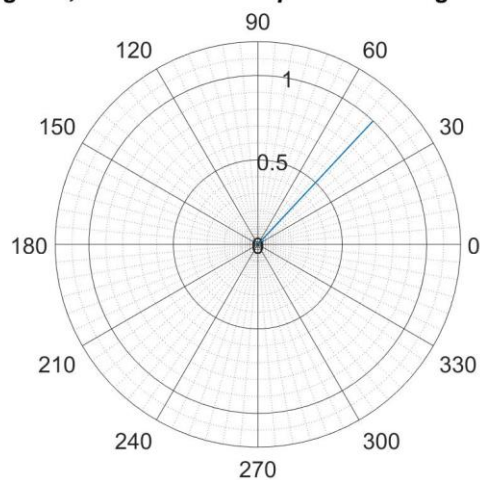
6.1.2 מבנה בעל 32 המקולים

על מנת לבחון את השינוי של ביצועי המערך עם עלייה במספר המקולים בחרנו לבדוק מקרה נוסף עבורו מספר המקולים הינם 32 ואילו רדיוסם עדיין 0.5 מטר. כמו בפעם הקודמת, נציג את מיקומי המקולים במרחב,



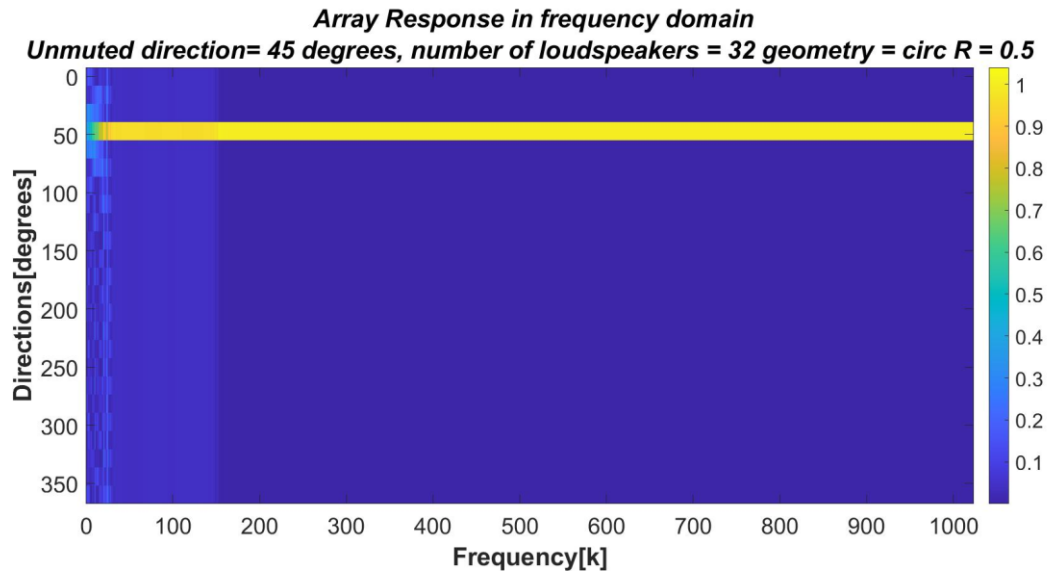
תגובת המערך בגרף פולארי

Array Response in frequency domain for all direction
Unmuted direction = 45 degrees, number of loudspeakers = 32 geometry = circ R = 0.5 Bin = 256



ניתן לראות כי כעת האנרגיה של האות מתרכזת באופן מיטבי יותר לכיוון ההשמעה הנבחר ולא זולגת לצדדים. כמו כן הכיווניות של האלומה מדויקת יותר.

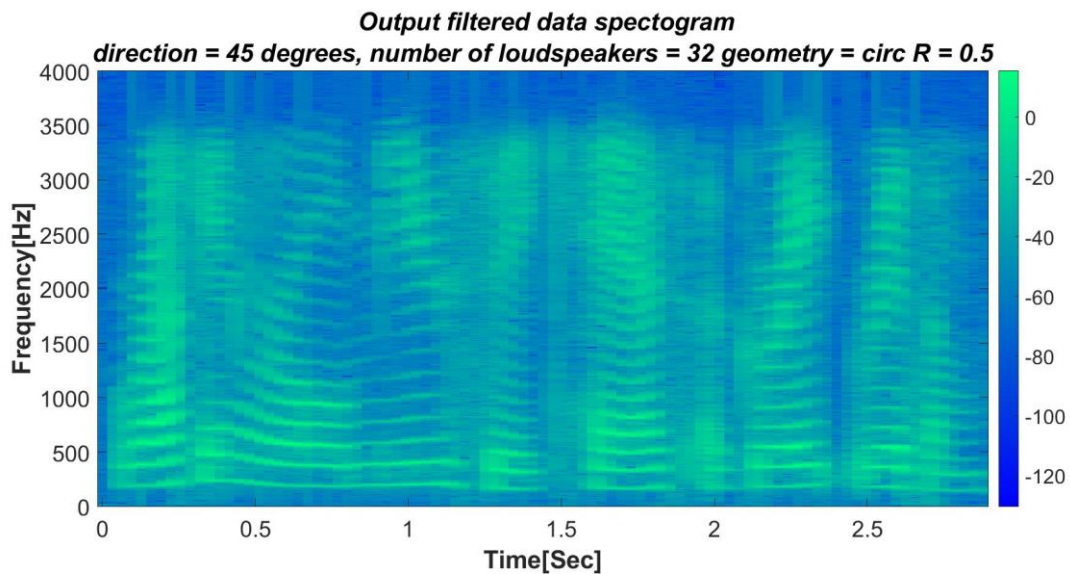
תגובת המערך בתדר לכל כיוון הינה



בתוצאות הגרף הנ"ל ישנן שתי מסקנות עיקריות.

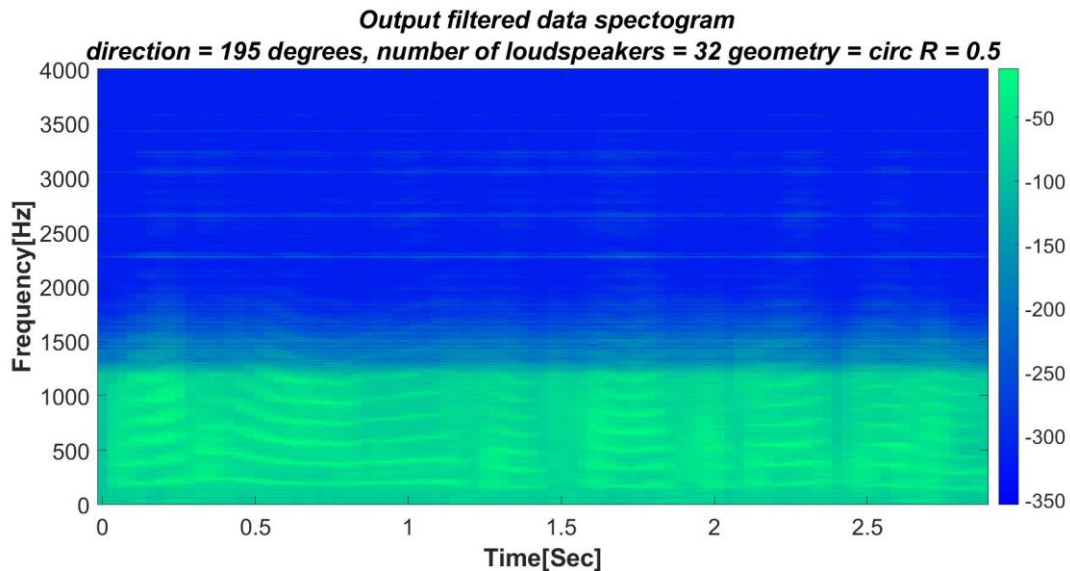
1. ביצועי המערך השתפרו באופן משמעותי עם עלייה במספר הרמקולים.
2. המערך מתפקד כמעט כמסנן אידיאלי שמעביר באופן מושלם את השמע בכיוון 45 מעלות ומשתיק אותו לשאר הכיוונים למעט מהתדרים הנמוכים שם ניתן להבחין בחוסר אחידות ההעברה.

להלן התמרת ה-STFT בכיוון השמע הרצוי



הערה: ניתן לראות בגרפים שמציגים את ההתמרה בכיוון הרצוי לשמע שיש טיפה הנחתה (גם עבור 16 רמקולים), אנחנו מסיקים שזה נובע מחוסר האידיאליות של המערכת. הסימולציה לוקחת בחשבון מאפיינים גיאומטריים וחומרתיים שעלולים להשפיע על האפקט שאנחנו מצפים לו.

בכיוון של 195 מעלות מתקבלת ההתמרה :

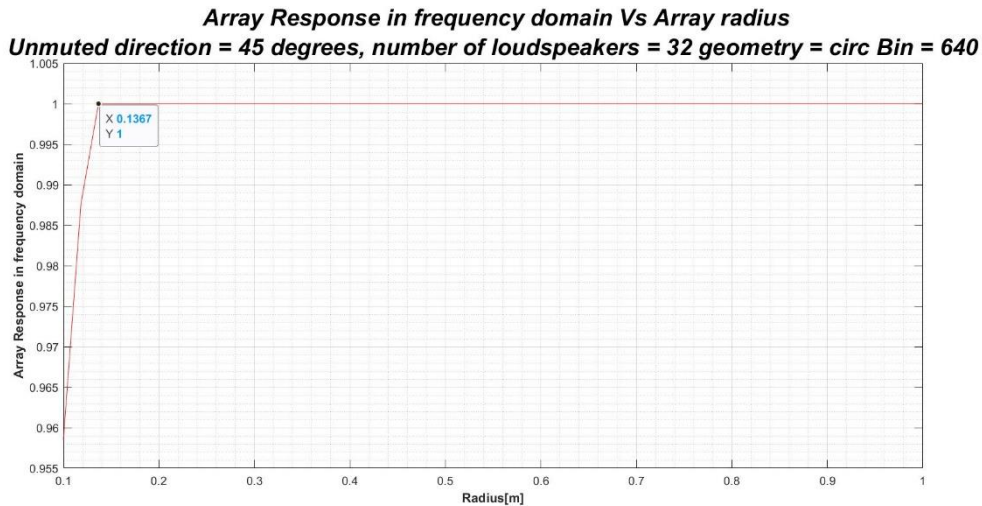


כמצופה , עם עלייה בפקטור 2 של מספר הרמקולים משתפרת ההנחתה לכיוונים שבהם נרצה להשתיק. ניתן לראות בגרף לעיל כי לכיוון של 195° מתקבלת הנחתה של -300db (הרקע הירוק) בתדרים הגבוהים ואילו בתדרים הנמוכים (הרקע הכחול) ההנחתה פחות משמעותית. שוב, פסי התדר האנלוגים שמועברים/מונחתים תואמים את פסי התדר הדיגיטליים שראינו שני גרפים קודם לכן. ניתן לראות שהחל מתדר של [Hz] 1500 ההנחתה נהיית משמעותית יותר, אנחנו מסיקים שמתחת לתדר זה בקונפיגורציה זו המערכת "מתקשה" לייצר את האפקט הנדרש, בהשוואה לקונפיגורציה של 16 רמקולים אנחנו מבינים שמספר הרמקולים כפרמטר משפיע על תדר הסף שהחל ממנו נקבל הנחתה משמעותית.

6.1.3 בחינת השפעת רדיוס המעגל

על מנת לבחון את השפעת פרמטר הרדיוס על יכולות המערך שלנו בתצורה מעגלית הרצנו sweep על 50 רדיוסים שונים בטווח שבין 0.1 – 1 m .

להלן התוצאות ,



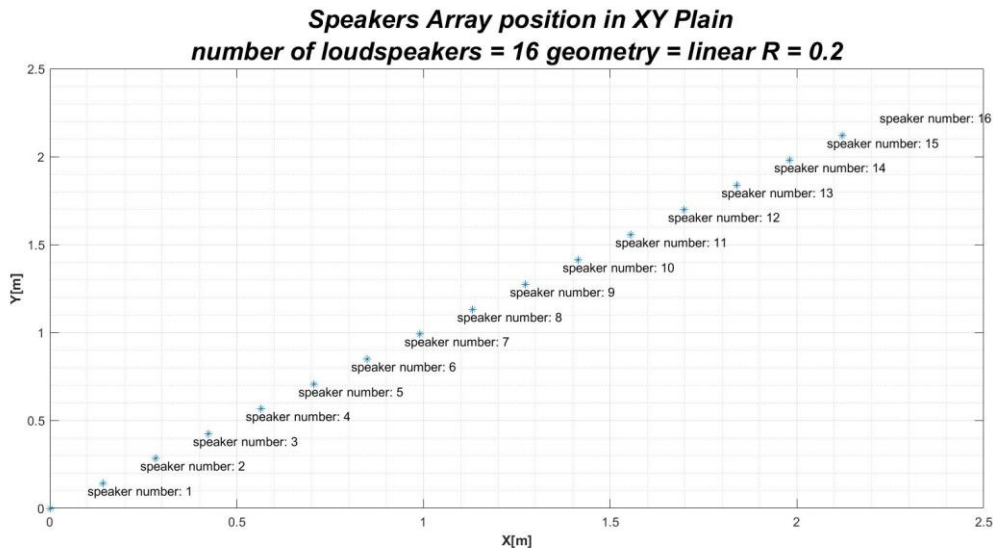
ניתן לראות עלייה של ארבעה אחוזים ביכולת ההעברה של השמע בכיוון הרצוי עד להגעה להעברה אידיאלית של פקטור 1 ברדיוס של 0.13 כפי שמסומן על הגרף. לכן אנחנו מסיקים שאיכות העברת השמע בכיוון הרצוי תלויה במרחק של הרמקולים ממרכז המערך. כמו כן אנחנו רואים מהגרף שמרדיוס נמוך יחסית בנוסף החל מרדיוס הסף אנחנו רואים תגובה יציבה לשינוי ברדיוס בהקשר להעברת האות בכיוון הרצוי.

6.2 מבנה לינארי

בחרנו לבחון קונפיגורציה נוספת של מערך רמקולים הפעם מסודרים בקו לינארי ישר עם שיפוע ומרחק בין הרמקולים לבחירה. בחרנו מרחק בין הרמקולים להיות 0.2 m ושיפוע של $m = 1$ (מעלות).

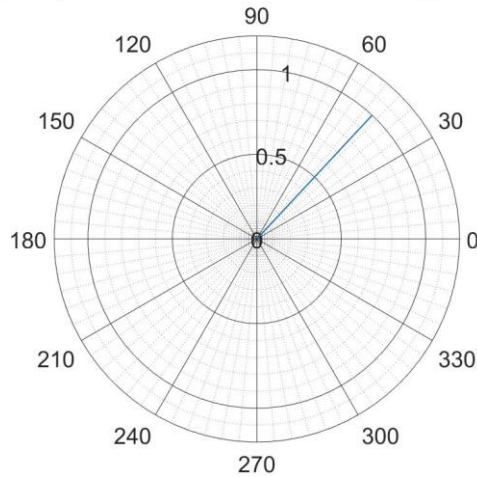
מבנה בעל 16 רמקולים

מיקומי הרמקולים במרחב על פני צירי XY :



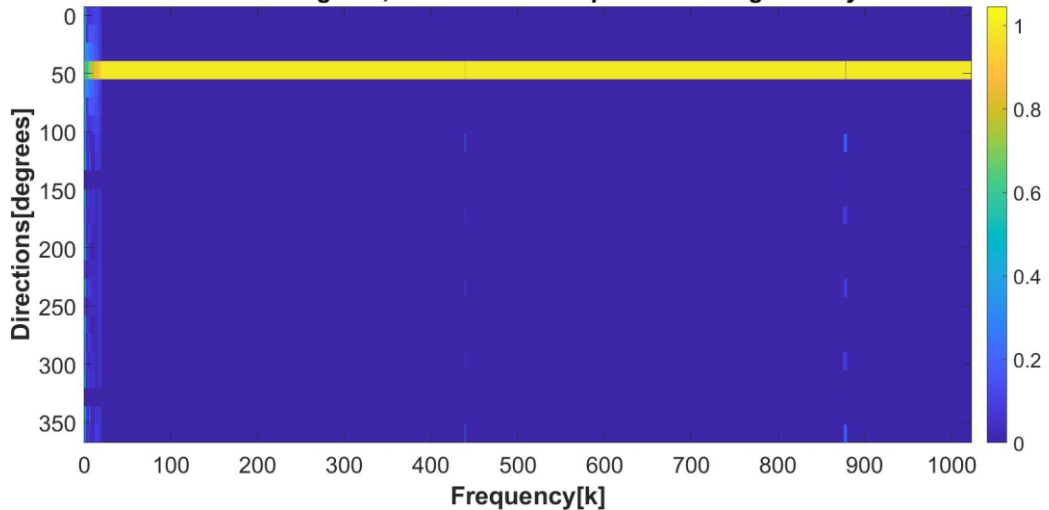
בגרף פולארי תגובת המערך הינה

Array Response in frequency domain for all direction
Unmuted direction = 45 degrees, number of loudspeakers = 16 geometry = linear R = 0.2 Bin = 256



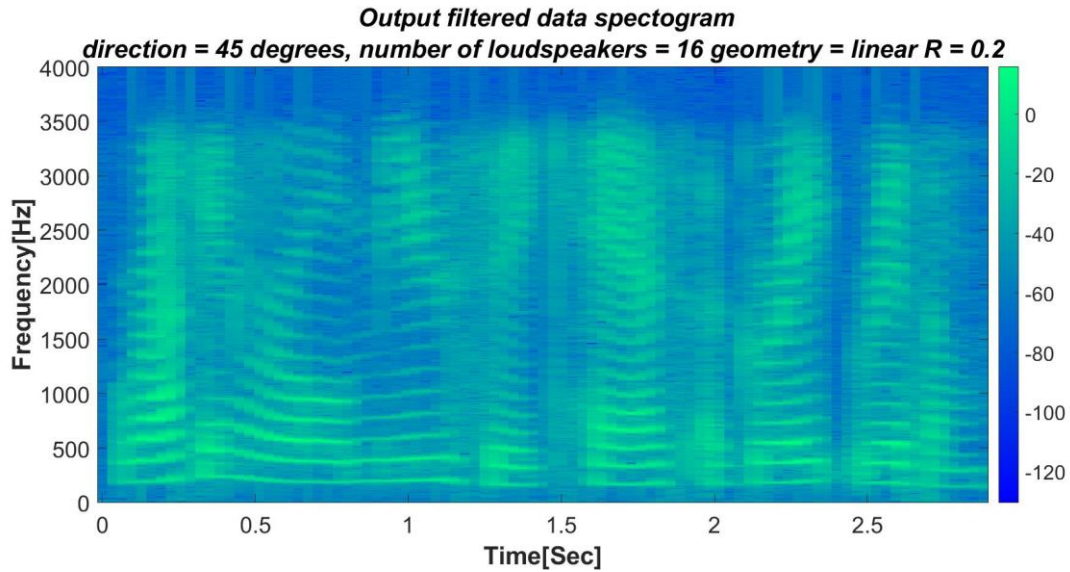
לא ניתן לראות הבדל משמעותי בין מבנה זה למבנה הקודם אותו ראינו בפרק. כעת נתבונן בתגובת המערך לכל כיוון במרחב ולכל תדר,

Array Response in frequency domain
Unmuted direction= 45 degrees, number of loudspeakers = 16 geometry = linear R = 0.2

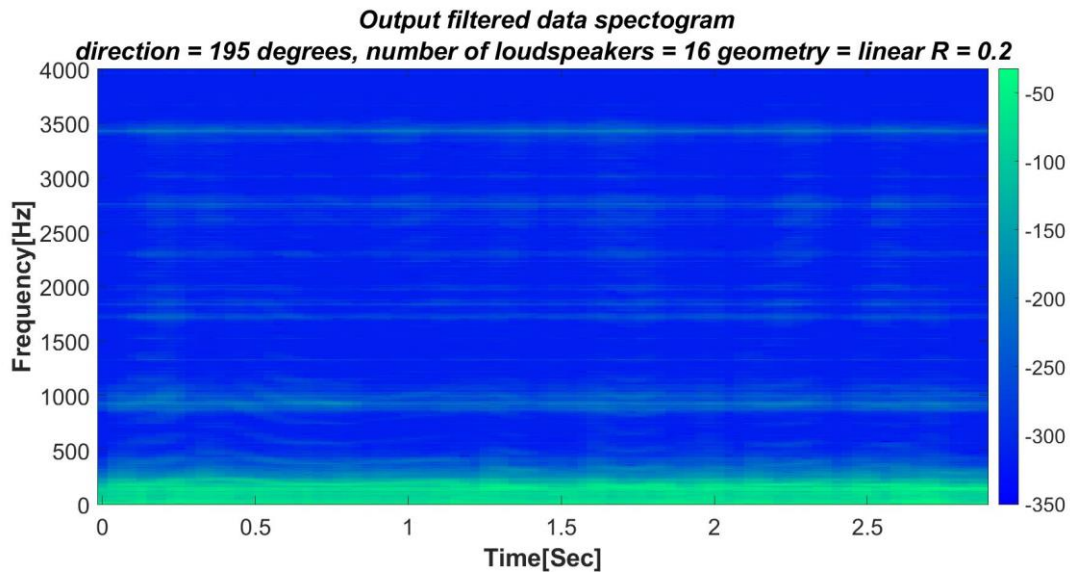


עם 16 רמקולים המערך הלינארי מתפקד באופן טוב יותר לעומת 2 התרחישים אותם הצגנו קודם לכן. את הקביעה הזו ניתן לבסס על פי התמרות הSTFT של אות שמע במוצא,

להלן התמרת הSTFT בכיוון השמע הרצוי



ואילו בכיוון מנוגד נקבל



אפשר לראות כי הביצועים של המערך הלינארי משמעותית טובים יותר. ההשתקה בכיוון שאינו בכיוון השמע הינה -350db כמעט בכל התדרים מלבד התדרים הנמוכים. ישנם פסי תדר שעדיין עוברים אך הם ברוחב קטן ופסי התדר הנמוכים שכן עוברים את המסך שלנו הינם בתחום מצומצם בהרבה מזה של המבנה הקודם.

המבנה מתפקד באופן כמעט מושלם מבחינת הסימולציה אך חשוב לזכור כי אנו בחרנו מקרים ספציפיים לבחינה וקביעה כללית ומבוססת יותר דורשת סימולציה רחבה בהרבה ומעמיקה עם שילוב של מימוש חומרתי וניסויים.

7. סימולציית חדר באמצעות (RIR) room impulse responses :

הסבר כללי

בסימולציה זו הוספנו לקוד יכולת להשתמש בפרמטרים של החדר כמו ממדים, פרמטר הדהוד וכמו כן פרמטרים של מערך הרמקולים ועל ידם לחשב את תגובת ההלם של החדר. המשמעות היא שהסימולציה תיצמד יותר למקרה אמיתי של מערך רמקולים בחדר, זאת מכיוון שחלל חדר אמיתי כולל החזרות מהקירות שמפריעות לפעולת המסננים שיצרנו והופכת את המערכת לפחות אידאלית. יכולת זו תיתן תוצאות שאמורות לשקף יותר את המציאות מאשר התוצאות שראינו בחלקים הקודמים שחושבו תחת תנאים של Free Space.

סימולציית הRIR כוללת את הפרמטרים הבאים :

- מיקום המיקרופון בחדר אשר קובע את הנקודה שבה תחושב תגובת ההלם. בסימולציה שלנו בנינו מערך מעגלי של מיקרופונים סביב מערך הרמקולים שלנו על מנת לסמלך את תגובת ההלם לכל כיוון במודל שלנו.
- ממדי החדר בקוארדינטות XY
- מיקום מערך הרמקולים בתוך החדר בקוארדינטות XY
- מספר דגימות מבוקש של תגובת ההלם של החדר. על מנת לקבל תגובת הלם מהפונקציה יש לבחור מספר סופי של דגימות.
- זמן הדהוד כתוצאה מהחזרי הקירות.
- סוג וכיווניות הרמקול

משמאל ניתן לראות את ערכי הפרמטרים

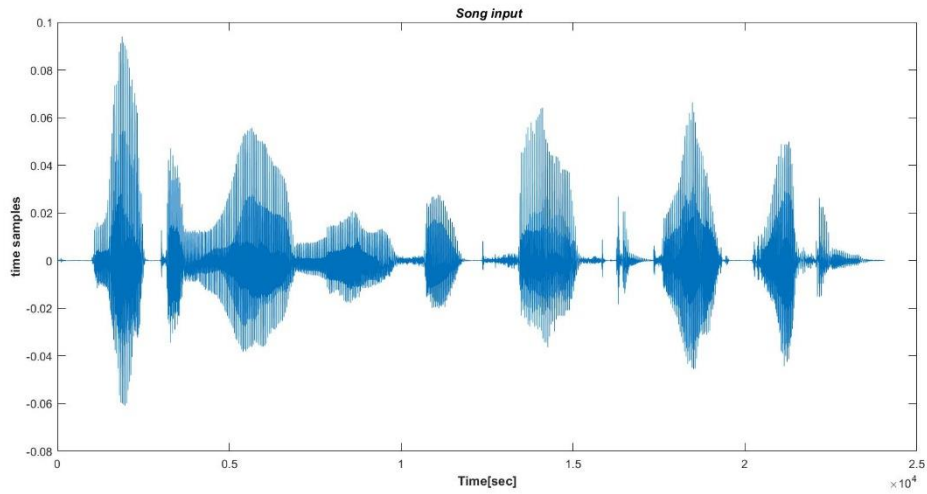
```
%room simulation variables
c = 343; % Sound velocity (m/s)
loudspeakers_height = 1.8;
|
s = [3 3 loudspeakers_height]; % Source position [x y z] (m)
L = [6 6 3]; % Room dimensions [x y z] (m)
n_rir = 8000; % Number of samples
beta = 0.3; % Reverberation time (s)
mtype = 'omnidirectional'; % Type of microphone
order = -1; % -1 equals maximum reflection order!
dim = 3; % Room dimension
orientation = [pi/2 0]; % Microphone orientation (rad)
hp_filter = 0; % Disable high-pass filter
```

שבחרנו כתלות בתרחיש שלנו

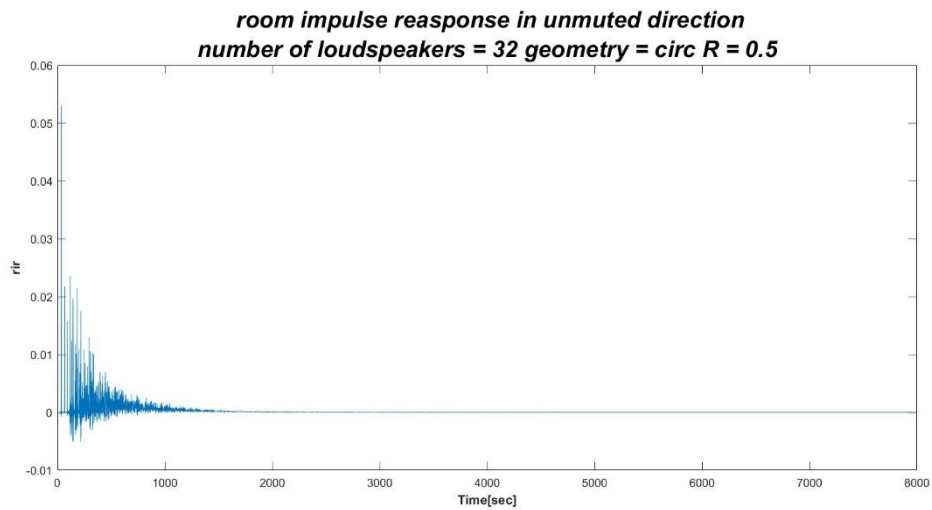
לשם הצגת התוצאות שכוללות את תגובת הלם של החדר אנחנו נתמקד במערך בעל מבנה מעגלי עם 32 רמקולים ברדיוס 0.5m. רדיוס מערך המיקרופונים הינו פי 3 מרדיוס המערך ועל כן כל מיקרופון במרחק של מטר ממערך הרמקולים, מרחק סביר אשר אדם היה עומד בו. תחילה לשם השוואה ועל מנת שנראה את פעולת הRIR נתבונן בתוצאות של אות המוצא בכיוון שבו אנו מעוניינים להשתיק וגם בכיוון שלא מעוניינים להשתיק.

7.1 תוצאות סימולציית *rir*

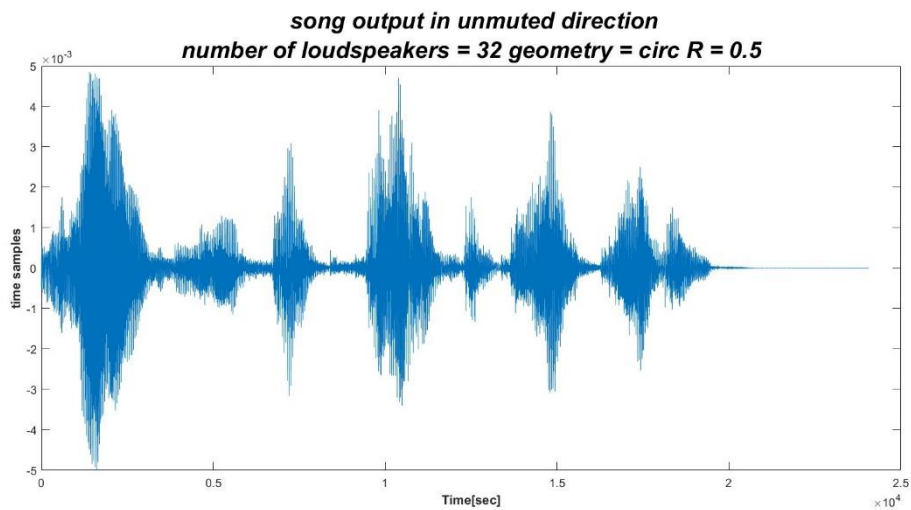
נציג את אות הכניסה שלנו בזמן (קטע סאונד שכולל את דיבור):



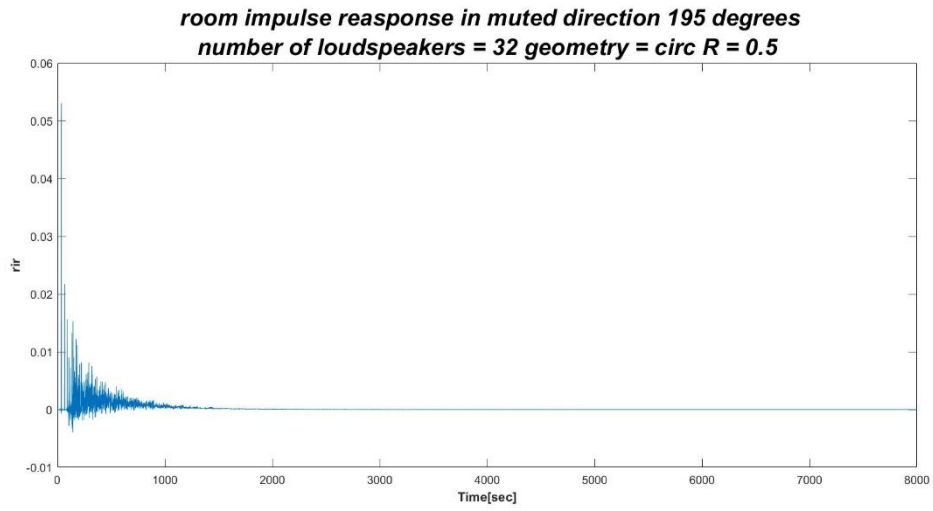
כעת נציג את התגובה להלם של החדר בכיוון שבו אנחנו לא מעוניינים להשתיק (45 מעלות) אשר חושבה על ידנו:



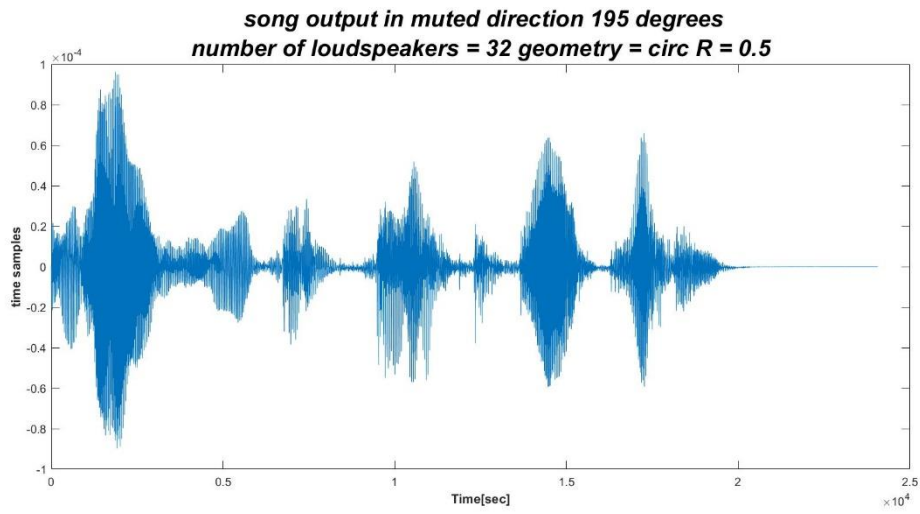
מוצא המערכת בכיוון 45° (אזור שאינו מושתק בחדר):



כעת , נביט בכיוון של 195° . תגובת ההלם של החדר בכיוון זה:



מוצא המערכת בכיוון 195° (אזור מושתק בחדר):

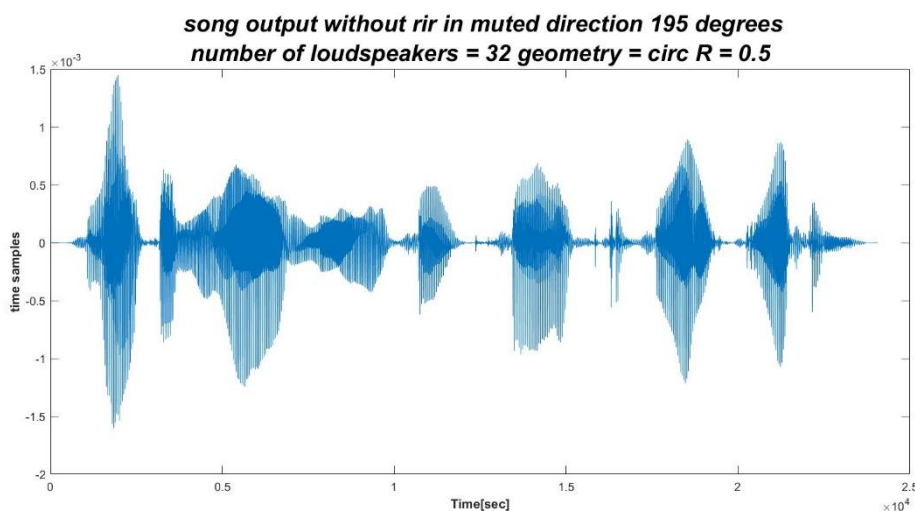
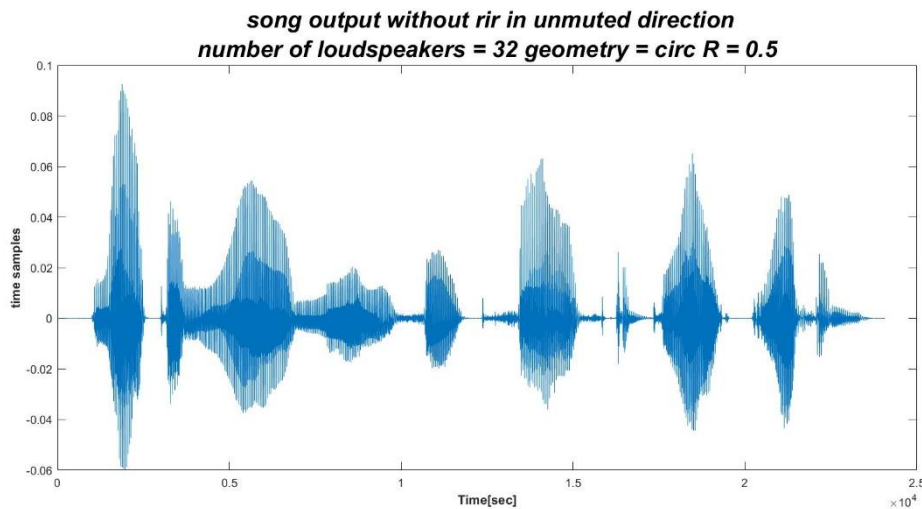


7.2 ניתוח תוצאות סימולציית *rir*

תחילה, נציין כי ניתן לראות שאין השפעה מהותית על כיוון ומיקום המיקרופון בחדר לתגובת ההלם שלה שכן ניתן לראות את הדמיון המובהק בין הגרפים של תגובת ההלם בחלק הקודם. קשה לראות את ההבדל ממבט גס, כיון שההבדל הוא בעיקר בזמני ההגעה של החזרות.

על מנת לבצע ניתוח תוצאות של סימולציית *rir* נוכל להשוות את מוצא המערך למקרה הראשון בו אנחנו בסימולציית Free Space.

נציג את הגרפים של מוצא המערך בשני הכיוונים המדוברים לשם השוואה:



כפי שניתן לראות תגובת ההלם בצורה של אות דועך, לכן כשנעביר את הכניסה (מוצא מערך הרמקולים) במערכת (החדר) האות שלנו ידעך.

ניתן לראות דעיכה בעוצמת הסיגנל בפקטור של 10^{-2} כתוצאה מהדהוד וחזרות בחלל בו נעשה הניסוי.

8. מסקנות

לאור המצב כתוצאה ממגיפת הקורונה הפרוייקט נעצר לפני שלב החומרה שתוכנן להתבצע בחדר האקוסטי בפקולטה, לכן עבודה זו מבוססת בעיקר על סימולציה רחבה.

מתוך התוצאות שהצגנו בשלב הסימולציה ושלבי המחקר הקודמים לה אנו מסיקים את המסקנות הבאות:

1. ישנן גישות שונות לפתרון הבעיה, בגישה החומריתית עלינו הרחבנו נתקלים בבעיות שונות מאלו שאנחנו נתקלנו בהם במהלך העבודה. הבעיות נובעות מהחיסרונות של המערכת אותם ציינו בפרק 5.2.

2. גיאומטריית המערך משפיעה רבות על התוצאה של פעולת ההשתקה, במערך שיכול להשיג רמת דיוק גבוהה ברוב המרחב אנחנו עלולים לשלם ברמת דיוק נמוכה בנקודות קצה (תדרי פעולה של מערך מעגלי מול לינארי).

3. נתוני החומרה, כמו למשל מס' רמקולים ומאפיינים פנימיים שלהם משפיעים על עוצמת הסיגנל ורמת ההנחתה.

4. המרחק של הרמקולים זה מזה יכול להשפיע על טיב המערכת עד לערך 90 ס"מ שממנו צפינו בהתייצבות.

5. סימולציית החדר לימדה אותנו כי על מנת להתגבר על הפגיעה בטיב המערכת כתוצאה מהידהוד ומעבר תווך לא חופשי אנחנו צריכים לייצר מסנן שתואם לתגובת הלם של החדר בכל כיוון לכל רמקול.

מהתהליך המחקר, הלמידה מהספרות האקדמית ומאמרים קודמים בתחום וכן מתהליך התכנון וביצוע הסימולציה השתכנענו מאוד כי התיאוריה שאנחנו מתבססים עלייה יכולה להוביל לניסוי חומרתי מלא שצפוי לעבוד ברמת סבירות גבוהה. יצרנו אלגוריתם שלוקח בחשבון הנחות הנדסיות מבוססות שלמדנו בתואר, ירדנו לעומק הפרטים של הבעיה וחישבנו מגוון מקרים וקונפיגורציות כולל סימולציית חדר שמשמשת לפרוייקטים רבים במסלול עיבוד אותות- דבר המעיד על אמינותה. תוצאות הסימולציה תאמו לצפיותינו ולכן אנחנו מאמינים שהשגנו את מטרת הפרוייקט.

9. סיכום

הפרויקט התחיל בתחילת השנה הרביעית והאחרונה לתואר ונמשך עד לסיומה. לאחר למידה ארוכה ופורייה של קורסים בתחום עיבוד האותות צברנו את הידע הנדרש בכדי לבצע את הפרויקט על הצד הטוב ביותר.

נדרשנו בתחילת הפרויקט במידול ותרגום הבעיה פיזיקלית אל מתמטיקה שתאפשר לנו לבנות את האלגוריתם. בהמשך מימשנו שלל מקרים וקונפיגורציות שונות לבעיה שלנו תוך כדי יישום הידע שלנו מהמסלול וראינו את השפעתם. למדנו והעמקנו בתחום עיבוד האות הדיגיטלי והכרנו טוב יותר את עולם הסימולציה ותוכנת ה-MATLAB. שיפרנו כאמור את יכולות הניתוח המתמטיות שלנו ויישומן. עוד ראינו את השפעת פרמטרים שונים על התוצאות אותן קיבלנו ויכולות המבנה שלנו.

לאור התוצאות, צבירת הידע והכלים שרכשנו במהלך הפרויקט רצינו להודות שוב לפרופ' שרון גנות - ראש מסלול עיבוד אותות, שבמהלך השנה השלישית והרביעית לתואר לימד בצורה יוצאת מן הכלל חלק מהקורסים שהיוו ידע בסיסי וחשוב שבלעדיהם לא היינו יכולים לבצע את הנדרש בפרויקט.

בנוסף, אנו מודים לעופר שוורץ, המנחה שלנו שליווה אותנו במשך כל הפרויקט, נפגש עימנו מידי שבוע והיה קשוב לבעיות שצצו בדרך ונתן פתרונות או רעיונות שעזרו לפתור בעיות אלו. למדנו ממך המון כיצד לבצע חשיבה נכונה, להיות יסודיים ולהתמודד עם תכנות אלגוריתמים עם כלים שהקנת לנו מעולמך האישי ונסיונך.

לסיום, נהנינו לבצע את הפרויקט הנ"ל, התמודדנו עם קשיים רבים במהלך הפרויקט מה שגרם לנו לחשוב על פתרונות יצירתיים ולרכוש ניסיון נוסף בתחום ולמדנו המון מסגל עיבוד האותות של בר אילן ומלמידתנו העצמית.

10. אלגוריתם

definition of the problem variables and constants

```
close all
clc

fs = 16e3;
c = 343; % sound velocity (m/s)
N = 32; %number of array loudspeakers
Q = 24; %number of grid lines in dividing the azimuth angle
theta [0,2pi]
Number_of_Radiuses_to_sweep = 50; %Number Radiuses to sweep of the speakers array
Num_of_freq_point = 1024; %number of frequency points
k = 0 : 1 : Num_of_freq_point-1;
direction_q_selected = 4; %select the azimuth of hearing
direction_to_plot = 14; %select the azimuth of testing
speaker_to_plot = 4; %select the speaker of interest
geometry = 'circ';

%chirp variables
t = 0 : 1/fs : 3;
f0 = 100;
f_final = 8e3;

%creating data from audio file
[time_sampeld, fs] = audioread('fdmy0-sx297.wav');

% time_sampeld = chirp(t,f0,3,f_final);
% pspectrum(chirp,1/fs)
```

creating the array response

```
G_sweep = zeros(Number_of_Radiuses_to_sweep,Q,N,Num_of_freq_point);
h_sweep = zeros(Number_of_Radiuses_to_sweep,N, Num_of_freq_point);
f_sweep = zeros(Number_of_Radiuses_to_sweep,Q,Num_of_freq_point);
i=1;

for Radius = linspace(0.1,1,Number_of_Radiuses_to_sweep)

%creating free space propogation matrix and filter
[G,h,f,speakers_position] = create_filter(Q, N, Num_of_freq_point, fs,
direction_q_selected, Radius, geometry);

G_sweep(i,:,:,:) = G;
h_sweep(i,:,:,:) = h;
f_sweep(i,:,:,:) = f;

i = i + 1;

end
```

filtering an audio file using STFT. sound output in chosen q direction

```
close all
clc

%WOLA variable definitions
Lh = 1024; %window length

Num_of_freq_point = Lh;
R = fix(Lh/4); %Jump (down sampling)

[time_filtered_wola,f_stft] =
Data_Filtering(time_sampeld,N,Q,G,h,Num_of_freq_point,R,Lh);
```

plot section

```
close all
clc

%%%%%%%%%%%% Spectrogram Plot %%%%%%%%%%%%%%

T = (0:size(f_stft,2)-1)/fs*R;
F = (0:Num_of_freq_point/2)*fs/2/(Num_of_freq_point/2);
figure('Name', 'Spectrogram in selected direction')
% surf(T,F,20*log10(abs(f_stft(1:Num_of_freq_point/2+1, :,speaker_to_plot)+eps)))
% hold on
imagesc(T,F,20*log10(abs(f_stft(1:Num_of_freq_point/2+1, :,direction_to_plot)+eps)))
xlabel('Time[Sec]','fontweight' , 'bold')
ylabel('Frequency[Hz]','fontweight' , 'bold')
set(gca,'fontsize',18)
title({'\it Output filtered data spectrogram';['direction = ',
num2str((direction_to_plot-1)*(360/Q)),...
' degrees, number of loudspeakers = ', num2str(N),' geometry = ',geometry, ' R = ',
num2str(Radius)]},...
'fontsize',20)
axis xy
colorbar
colormap(winter)

filename = ['Output filtered data spectrogram direction = ',
num2str((direction_to_plot-1)*(360/Q)),...
' degrees, number of loudspeakers = ', num2str(N), ' ', geometry, ' R = ',
num2str(Radius)];
saveas(figure(1),[pwd ['\figures\Spectrogram Plot\'],filename,'.fig']);

%%%%%%%%%%%% Array Response Polar Plot %%%%%%%%%%%%%%

figure('Name', 'Array Response Polar Plot')
polarplot(linspace(0,2*pi,Q),f(:,640),'Linewidth',1);
set(gca,'fontsize',18,'GridAlpha',1,'MinorGridAlpha',0.5)
title({'\it Array Response in frequency domain for all direction';['Unmuted direction
= ',...
num2str((direction_q_selected-1)*(360/Q)), ' degrees, number of loudspeakers = ',
num2str(N), ...
' geometry = ', geometry, ' R = ', num2str(Radius), ' Bin = ', num2str(640)]},
'fontsize',20)
grid on
```

```

grid minor

filename_1 = ['Array Response Polar Plot Unmuted direction = ',
num2str((direction_q_selected-1)*(360/Q)),...
' degrees, number of loudspeakers = ', num2str(N),' geometry = ',geometry, ' R = ',...
num2str(Radius), ' Bin = ', num2str(640)];
saveas(figure(2),[pwd ['\figures\Array Response Polar Plot\'',filename_1,'.fig']]);

%%%%%%%%%%%% Array Response in frequency domain %%%%%%%%%%%%%%

figure('Name', 'Array Response in frequency domain')
imagesc(k, linspace(0,360,Q), abs(f));
set(gca,'fontsize',18)
title({'\it Array Response in frequency domain';['Unmuted direction= ',
num2str((direction_q_selected-1)*(360/Q)),...
' degrees, number of loudspeakers = ', num2str(N),' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
xlabel('Frequency[k]', 'fontweight', 'bold')
ylabel('Directions[degrees]', 'fontweight', 'bold')
colorbar

filename_2 = ['Array Response in frequency domain Unmuted direction = ',
num2str((direction_q_selected-1)*(360/Q)),...
' degrees, number of loudspeakers = ', num2str(N),' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(3),[pwd ['\figures\Array Response in frequency domain\'',filename_2,'.fig']]);

%%%%%%%%%%%% Speakers Array position %%%%%%%%%%%%%%

switch geometry
case 'circ'
figure('Name', 'Speakers Array position')
speakers_array = [0:2*pi/N:2*pi*(N-1)/N ;Radius*ones(1,N)];
polarplot(speakers_array(1,:),speakers_array(2,:), 'LineStyle','none','Marker','*')
set(gca,'fontsize',14,'GridAlpha',1,'MinorGridAlpha',0.5)
title({'\it Speakers Array position in XY Plain';['number of loudspeakers = ',
num2str(N), ...
' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
rlim([0 3*Radius])

grid on
grid minor

for i=1 : N
dr = Radius/2; % displacement so the text does not overlay the data points
line = ['#', num2str(i)];
text(speakers_array(1,i), speakers_array(2,i)+dr, line);
end

case 'linear'
figure('Name', 'Speakers Array position')
stem(speakers_position(1,:),speakers_position(2,:), 'LineStyle','none','Marker','*')
title({'\it Speakers Array position in XY Plain';['number of loudspeakers = ',
num2str(N), ...
' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
xlabel('X[m]', 'fontweight', 'bold')

```

```

ylabel('Y[m]','fontweight','bold')

grid on
grid minor

for i=1 : N
dx = 0.1; dy = 0.1; % displacement so the text does not overlay the data points
line = ['speaker : ',num2str(i)];
text(speakers_position(1,i)+dx, speakers_position(2,i)+dy, line);
end

end

filename_3 = ['Speakers Array position in XY Plain ', 'number of loudspeakers = ',
num2str(N), ...
' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(4),[pwd ['\figures\Speakers Array position\'',filename_3,'.fig']]);

%%%%%%%%%%%% Sound Plot %%%%%%%%%%%%%%

sound(time_filtered_wola(:,direction_to_plot));
str = ['C:\Users\ravid\Desktop\FinalProject\sound plots\sound simulated domain in
direction = ', num2str((direction_to_plot-1)*(360/Q)),...
' degrees, number of loudspeakers = ', num2str(N),' geometry = ',geometry, ' R =
',num2str(Radius),'.wav'];
audiowrite(str,time_filtered_wola(:,direction_to_plot),fs);

%%%%%%%%%%%% Radius sweep Plot %%%%%%%%%%%%%%
figure('Name', 'Radius sweep')
% unmuted direction
plot(linspace(0.1,1,Number_of_Radiuses_to_sweep),f_sweep(:,4,640),'color','r')
title({'\it Array Response in frequency domain Vs Array radius';['Unmuted direction =
',...
num2str((direction_q_selected-1)*(360/Q)),' degrees, number of loudspeakers = ',
num2str(N), ...
' geometry = ', geometry, ' Bin = ', num2str(640)]}, 'fontsize',20)
grid on
grid minor

filename_4 = ['Array Response in frequency domain Vs Array radius Unmuted direction =
', num2str((direction_q_selected-1)*(360/Q)),...
' degrees, number of loudspeakers = ', num2str(N),' geometry = ',geometry];
saveas(figure(5),[pwd ['\figures\radius sweep\'',filename_4,'.fig']]);

```

testing signals with room impulse response

```
[sound_filtered_rir,rir_array] = my_rir(time_filtered_wola,Radius,Q,fs);
```

evaluate performance of rir

```

figure('Name', 'song input')
plot(time_sampeld);
title({'\it Song input'})
xlabel('Time[sec]','fontweight','bold')

```

```

ylabel('time samples','fontweight','bold')

filename_5 = ['Song input ', 'number of loudspeakers = ', num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(1),[pwd ['\figures\rir plots\' ,filename_5, '.fig']]);

figure('Name', 'song output in unmuted direction')
plot(sound_filtered_rir(:,direction_to_plot));
title({'\it song output in unmuted direction';['number of loudspeakers = ',
num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
xlabel('Time[sec]','fontweight','bold')
ylabel('time samples','fontweight','bold')

filename_6 = ['song output in unmuted direction ', 'number of loudspeakers = ',
num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(2),[pwd ['\figures\rir plots\' ,filename_6, '.fig']]);

figure('Name', 'room impulse response in unmuted direction')
plot(rir_array(:,4));
title({'\it room impulse response in unmuted direction';['number of loudspeakers = ',
num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
xlabel('Time[sec]','fontweight','bold')
ylabel('rir','fontweight','bold')

filename_7 = ['room impulse response in unmuted direction ', 'number of loudspeakers
= ', num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(3),[pwd ['\figures\rir plots\' ,filename_7, '.fig']]);

figure('Name', 'song output in muted direction 195 degrees')
plot(sound_filtered_rir(:,14));
title({'\it song output in muted direction 195 degrees';['number of loudspeakers = ',
num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
xlabel('Time[sec]','fontweight','bold')
ylabel('time samples','fontweight','bold')

filename_8 = ['song output in muted direction 195 degrees ', 'number of loudspeakers =
', num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(4),[pwd ['\figures\rir plots\' ,filename_8, '.fig']]);

figure('Name', 'room impulse response in muted direction 195 degrees')
plot(rir_array(:,14));
title({'\it room impulse response in muted direction 195 degrees';['number of
Loudspeakers = ', num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
xlabel('Time[sec]','fontweight','bold')
ylabel('rir','fontweight','bold')

filename_9 = ['room impulse response in muted direction 195 degrees ', 'number of
Loudspeakers = ', num2str(N), ...
            ' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(5),[pwd ['\figures\rir plots\' ,filename_9, '.fig']]);

```

```

figure('Name', 'song output without rir in unmuted direction')
plot(time_filtered_wola(:,4));
title({'\it song output without rir in unmuted direction';['number of loudspeakers = ', num2str(N), ...
    ' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
xlabel('Time[sec]','fontweight','bold')
ylabel('time samples','fontweight','bold')

filename_10 = ['song output without rir in unmuted direction ', 'number of
Loudspeakers = ', num2str(N), ...
    ' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(6),[pwd ['\figures\rir plots\' ,filename_10, '.fig']]);

figure('Name', 'song output without rir in muted direction 195 degrees')
plot(time_filtered_wola(:,14));
title({'\it song output without rir in muted direction 195 degrees';['number of
Loudspeakers = ', num2str(N), ...
    ' geometry = ',geometry, ' R = ',num2str(Radius)]}, 'fontsize',20)
xlabel('Time[sec]','fontweight','bold')
ylabel('time samples','fontweight','bold')

filename_11 = ['song output without rir in muted direction 195 degrees ', 'number of
Loudspeakers = ', num2str(N), ...
    ' geometry = ',geometry, ' R = ',num2str(Radius)];
saveas(figure(7),[pwd ['\figures\rir plots\' ,filename_11, '.fig']]);

```

[Published with MATLAB® R2020a](#)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by : Ravid Avraham & Leroy Dokhanian
% Last update : 25/05/2020
% Function name : create_filter
% inputs : Q, N, Num_of_freq_point (frequency resolution), fs, direction_q_selected
% (chosen direction of hearing), Radius, geometry
% outputs : G , h , f , speakers_position
% Description : this function first sets up an array of chosen number of
% loudspeakers and calculate the time delay between them for all directions
% selected q. second the function calculate a filter h so that the array
% output will be a function of choise f_ideal. finally the function calculate
% and return the propogation matrix G along with the filter h and array response and
% speakers position in XY.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [G,h,f,speakers_position] = create_filter(Q, N, Num_of_freq_point, fs,
direction_q_selected,Radius, geometry)

switch geometry

    case 'circ'

%Definition of an array of N loudspeakers in a geometry of a circle with
%radius of choise

R = Radius; %radius of the speakers array

```

```

%position matrix (2D) of each loudspeaker in polarian cooardination
speakers_array = [R*ones(1,N) ; 0:2*pi/N:2*pi*(N-1)/N];

%position matrix (2D) of each loudspeaker in XY cooardination
speakers_position = [speakers_array(1,:).*cos(speakers_array(2,:)) ;
speakers_array(1,:).*sin(speakers_array(2,:))];

    case 'linear'

        speakers_position = zeros(2,N);

        m = 1;

        for i = 1 : N-1

            speakers_position(1,i+1) = ((speakers_position(1,i) +
m*speakers_position(2,i))...
            + sqrt((speakers_position(1,i)+m*speakers_position(2,i))^2-
(1+m^2)*(speakers_position(1,i)^2 ...
            + speakers_position(2,i)^2-Radius^2)))/(1+m^2);

            speakers_position(2,i+1) = m*speakers_position(1,i+1);

        end

    end

end

%finding time delay between loudspeakers
[Time_Delay_Matrix] = Find_Time_Delay(speakers_position ,N ,Q ,fs);

%defenitions
f_ideal = zeros(Q,Num_of_freq_point); %vector of the desierd direction
f_ideal(direction_q_selected,:) = ones(1, Num_of_freq_point);

h = zeros(N, Num_of_freq_point);
G = zeros(Q,N,Num_of_freq_point);

%calculation of the speakers filterd output using least mean squares
for k=1 : Num_of_freq_point

    G(:,:,k) = exp(-1i.*(2*pi*((k-1)/Num_of_freq_point)*Time_Delay_Matrix)); %creating
propogation matrix in freq plain

    h(:,k) = ((G(:,:,k)'*G(:,:,k))+trace(G(:,:,k)'*G(:,:,k))*(10^-
1))\G(:,:,k)'*f_ideal(:,k); %calc of the filter

    f(:,k) = G(:,:,k)*h(:,k); %estimated filterd spectrum by least mean squares

end

end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by : Ravid Avraham & Leroy Dokhanian
% Last update : 08/01/2020
% Function name :Find_Time_Delay
%inputs : speakers_position ,N (number of array speakers) ,Q (number of
%directions) ,fs (frequency of sampling)
%outputs : Time_Delay_Matrix
% Description : this function calculate the time delay of each loudspeaker
% in direction q selected and returns a QxN matrix of delays (in samples), delay for
% all speakers and directions.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Time_Delay_Matrix] = Find_Time_Delay(speakers_position ,N ,Q ,fs)

c_sound = 343; %speed of sound [m/sec]
Time_Delay_Matrix = zeros(Q,N); %matrix of delay in time from loudspeaker reference

%setting reference speaker
reference_position = [0 0] ;

%calculation of time delays
for i = 0 : N-1
    for j = 0 : Q-1

        current_angle = j*2*pi/Q ; %current angular direction q
        beta = myatan(speakers_position(1,i+1),speakers_position(2,i+1)) + (3*pi/2-
current_angle); %calculation of the the angle between speaker and the direction
tangent
        L = sin(beta)*pdist2(speakers_position(:,i+1).',reference_position,
'euclidean');
        Time_Delay_Matrix(j+1,i+1) = (L/c_sound)*fs;

    end
end
end

```

[Published with MATLAB® R2020a](#)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by : Ravid Avraham & Leroy Dokhanian
% Last Date of update : 08/01/2020
% Function name :myatan
%inputs : x,y (positions in the euclidean space)
%outputs : angle
% Description : this function calculate angle in radians of (x,y) point using the the
angles from 0 to 2*pi.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function angle = myatan(x,y)

if x>0 && y>=0
    angle=atan(y/x);
end

```

```

if x<0
    angle=pi+atan(y/x);
end

if x>0 && y<0
    angle= 2*pi+atan(y/x);
end

if y>0 && x==0
    angle=pi/2;
end

if y<0 && x==0
    angle=-pi/2;
end

if y==0 && x==0
    angle = 0;
end

end

```

[Published with MATLAB® R2020a](#)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by : Ravid Avraham & Leroy Dokhanian
% Last update : 27/06/2020
% Function name : Data_Filtering
% inputs : time_sampeld, N, Q, G, h, Num_of_freq_point, R, Lh
% outputs : time_filtered_wola, f_stft
% Description : this function filters the data using STFT and the filter h
% and reconstruct the data using ISTFT.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [time_filtered_wola,f_stft] =
Data_Filtering(time_sampeld,N,Q,G,h,Num_of_freq_point,R,Lh);

w_analysis = hamming(Lh); %window of analysis
w_synthesis = synthesis_win(w_analysis, R);

Spectrogram = my_stft(time_sampeld, w_analysis, R ,Num_of_freq_point,N); %spectrogram of
the all array
frames = size(Spectrogram,2) ;

h = repmat(h,[1 1 frames]);
h_permute = permute(h , [2 3 1]);
G = repmat(G, [1 1 1 frames]);
f_stft = zeros(Q, Num_of_freq_point, frames); %array response in all direction in
freq domain using STFT

Spectrogram_Filtered = h_permute.*Spectrogram;
Spectrogram_Filtered = permute(Spectrogram_Filtered, [3 1 2]);

for r=1 : frames
    for k=1 : Num_of_freq_point

```

```

        f_stft(:,k,r) = G(:, :,k,r)*Spectrogram_Filtered(:,k,r); %estimated filtered spectrum
        by least mean squares
    end
end

f_stft = permute(f_stft, [2 3 1]);
time_filtered_wola = real(my_istft(f_stft, w_synthesis, R));

end

```

[Published with MATLAB® R2020a](#)

```

function spectrogram = my_stft(s, w_analysis, R ,NFFT,N)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%coefficients%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = 1; %descrete time
Lh = length(w_analysis);%length of analysis window
L_signal = length(s);%length of data
h = flip1r(w_analysis);% h(n) = h(-n)
M = NFFT; %number of freq points

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%analysis%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if Lh>M

for r=1 : R : L_signal-Lh+1

    sbuf(1:Lh,1) = s(r:r+Lh-1,1);

    tempbuf = h.*sbuf;

    tempbuf_paded = padarray(tempbuf,mod(length(tempbuf),M), 'post');

    tempbuf_1 = reshape(tempbuf,[M,length(tempbuf_paded)/M]);

    conved_signal = sum(tempbuf_1,2);

    spectrogram(:,n) = fft(conved_signal,M);

    n = n+1;

end

else

for r=1 : R : L_signal-Lh+1

    sbuf(1:Lh,1) = s(r:r+Lh-1,1);

    tempbuf = h.*sbuf;

    tempbuf_paded = padarray(tempbuf,(M-Lh), 'post');

    spectrogram(:,n) = fft(tempbuf_paded,M);

end

```

```

    n = n+1;

end

end

k=0:M-1;

n=0:floor((L_signal-Lh)/R);

spectrogram=spectrogram.*exp(-1i*2*pi*k' .*n*R/M);

spectrogram= repmat(spectrogram, [1 1 N]);

end

```

```

function signal_reconstruction = my_istft(Spectrogram, w_synthesis, R)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%coefficient%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[M, frames, Q] = size(Spectrogram);

k = 0:M-1;
n = 0:frames-1;
Spectrogram = Spectrogram.*exp(1i*2*pi*k' .*n*R/M);
Lf = length(w_synthesis);
hbuf = fliplr(w_synthesis);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%synthesis%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:Q

    r = 1;

    for m=1:frames

        sbuf=Spectrogram(:,m,i);
        sbuf(1,1) = 0 ;
        s1=M*(ifft(sbuf,M)); %performing M*IFFT
        s_per = [repmat(s1, floor(Lf / M), 1); s1(1:mod(Lf, M))];%periodic replication
        tempbuf=hbuf.*s_per;

        if m==1
            signal_reconstruction_i=tempbuf;

        else %overlap & add
            signal_reconstruction_i=[signal_reconstruction_i; tempbuf(Lf-R+1:end,1)];
            signal_reconstruction_i(r:r+Lf-R-1,1)=signal_reconstruction_i(r:r+Lf-R-1,1)+tempbuf(1:Lf-R,1);

        end

        r=r+R;
    end
end

```

```

end

signal_reconstruction_i=signal_reconstruction_i/Lf;

if i==1
    signal_reconstruction = signal_reconstruction_i;
else
    signal_reconstruction = [signal_reconstruction signal_reconstruction_i];
end
end
end
end

```

[Published with MATLAB® R2020a](#)

```

function [ws] = synthesis_win(w, L)
N = length(w);
ws = w;
w2 = zeros(N, 1);
for n=0:N-1
    qn = floor(n/L);
    qm = floor((N-1-n)/L);
    for q=-qn:qm
        w2(n+1) = w2(n+1)+w(q*L+n+1)^2;
    end
end
ws = flip(w./w2, 1);

```

[Published with MATLAB® R2020a](#)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written by : Ravid Avraham & Leroy Dokhanian
% Last update : 3/08/2020
% Function name : my_rir
% inputs : time_filtered_wola, Radius, Q, fs
% outputs : sound_filtered_rir, rir_arrayn
% Description : this function define a room parameters to simulate and calculate
% the room impulse response (rir). seconde the function filters input datat
% with the rir and returns the filtered data along with the array of microphones
% tested.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [sound_filtered_rir,rir_array] = my_rir(time_filtered_wola,Radius,Q,fs)

Radius_of_reciver = 3*Radius; %radius of microphone

%position matrix (2D) of each microphone in polarian cooardination
microphone_array = [Radius_of_reciver*ones(1,Q) ; 0:2*pi/Q:2*pi*(Q-1)/Q];

%position matrix (2D) of each microphone in XY cooardination
microphone_position = [microphone_array(1,:).*cos(microphone_array(2,:))+3 ;...
    microphone_array(1,:).*sin(microphone_array(2,:))+3];

%room simulation variables
c = 343; % Sound velocity (m/s)
Loudspeakers_height = 1.8;

```

```

s = [3 3 loudspeakers_height];% Source position [x y z] (m)
L = [6 6 3];                % Room dimensions [x y z] (m)
n_rir = 8000;                % Number of samples
beta = 0.3;                  % Reverberation time (s)
mtype = 'omnidirectional';  % Type of microphone
order = -1;                  % -1 equals maximum reflection order!
dim = 3;                     % Room dimension
orientation = [pi/2 0];     % Microphone orientation (rad)
hp_filter = 0;               % Disable high-pass filter

sound_filtered_rir = zeros(size(time_filtered_wola,1),Q);
rir_array = zeros(n_rir,Q); % microphone position [x y z] (m)
for q = 1 : Q

    r = [microphone_position(1,q) microphone_position(2,q) loudspeakers_height]; %
    Receiver position [x y z] (m)
    rir_array(:,q) = (rir_generator(c, fs, r, s, L, beta, n_rir, mtype, order, dim,
    orientation, hp_filter))';
    sound_filtered_rir(:,q) = conv(time_filtered_wola(:,q),rir_array(:,q),'same');

end

end

```

[Published with MATLAB® R2020a](#)

11. מקורות

- [/https://ultrasonic-audio.com](https://ultrasonic-audio.com) .1
- <https://patents.google.com/patent/EP1175812B1/en> .2
- https://en.wikipedia.org/wiki/Parametric_array .3