סיכום קצרצר של הנאמר בזמן העברת המצגת:

שקף ראשון : הגדרות בסיסיות

תמונה שמכילה טקסט, תרשים, קו, צילום מסך

התיאור נוצר באופן אוטומטי

**PDM בראשי תיבות זה (pulse density modulation ) – אפנון צפיפות פולסים.**

**זו שיטה דיגטלית (עובדים עם 1- 1 בלבד) לאפיון אות שהוא אנלוגי, לפי צפיפות הפולסים.**

**בPDM האות האנלוגי נדגם בקצב גבוה (תדר גבוה).**

**דוגמא: ככל שאמפליטודה גבוהה יותר, כך הדגימות 1 שנלקחו הן יותר צפופות.**

**יתרונות: קל ליישם מבחינת חומרה, קל להחזיר חזרה לאות אנלוגי בעזרת מסננים פשוטים.**

**חסרונות: שימוש בקצב דגימה גבוה יותר, שדורש יותר זכרון וגם רוחב פס גדול יותר.**

**PCM**, או בשמו המלא Pulse Code Modulation, הוא שיטה נפוצה לייצוג דיגיטלי של אותות אנלוגיים, במיוחד אותות שמע.

בשיטת PCM, האות האנלוגי נדגם במרווחי זמן קבועים, וכל דגימה מיוצגת על ידי ערך מספרי. הדיוק של הייצוג תלוי במספר הביטים המשמשים לכל דגימה.

PCM הוא הבסיס לפורמטים שמע דיגיטליים נפוצים, כמו קבצי WAV ו-CD אודיו.

היתרונות העיקריים של PCM כוללים:

איכות שמע גבוהה, ללא איבוד מידע משמעותי בתהליך הדגימה.

תאימות רחבה - רוב מערכות השמע והתוכנות יכולות לעבוד עם אותות PCM.

קלות בעיבוד - פורמט זה מאפשר ביצוע מניפולציות שונות על האות בקלות יחסית.

חשוב לציין כי בעוד ש-PDM מספק דגימה בקצב גבוה מאוד, PCM משתמש בדרך כלל בקצבי דגימה נמוכים יותר, אך עם רזולוציה גבוהה יותר לכל דגימה."

**המרת PDM to PCM- הוא תהליך של שינוי מאפיון לפי צפיפות הפולסים לשיטה של אפיון לפי** הדגימות של האמפליטודה.

שקף שני: למה שנעשה את ההמרה?

תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי

**אינטגרציה עם מערכות שמע:**

רוב מערכות השמע והתוכנות עובדות עם פורמטים של PCM כמו WAV (שמע). המרה ל-PCM מאפשרת שימוש חלק של אותות PDM במערכות אלו, ללא צורך בשינויים מהותיים.

**יכולת שמע:**

אותות PDM מיוצגים ע"י תדר גבוה שבני האדם לא יכולים לשמוע אותו על ידי מערכות שמע סטנדרטיות או על ידי האוזן האנושית כי הן בתדר גבוה. המרה ל-PCM הופכת את האות לפורמט שניתן להשמעה ועיבוד ישיר.

**הפחתת רעשים:**

תהליך ההמרה מ-PDM ל-PCM כולל שלב של סינון. סינון זה עוזר להפחית רעשים בתדרים גבוהים שנוצרים כתוצאה מהדגימה בקצב גבוה של PDM, ובכך משפר את איכות השמע.

**יעילות אחסון:**

אותות PDM מכילים כמות גדולה של נתונים בגלל קצב הדגימה הגבוה שלהם. המרה ל-PCM, הכוללת תהליך של דצימציה (הורדת קצב הדגימה), מפחיתה משמעותית את כמות הנתונים אבל עדיין שומרת על המידע הנחוץ לשמיעה. זה מוביל לחיסכון משמעותי בשטח אחסון.

שקף שלישי: שלבי תהליך ההמרה:

תמונה שמכילה טקסט, צילום מסך, תרשים, גופן

התיאור נוצר באופן אוטומטי

תהליך ההמרה מ-PDM ל-WAV מורכב מכמה שלבים עיקריים:

**קריאת קובץ PDM:**

השלב הראשון הוא קריאה של קובץ הטקסט המכיל את האות PDM.

זה בעצם קובץ טקסט עם אחד ומינוס אחד.

אנחנו קוראים את הנתונים וממירים אותם למערך NUMPY של מספרים שלמים.

**כעת יופיעו שני השלבים העיקריים בתהליך ההמרה:**

**LPF (Low Pass Filter - מסנן מעביר נמוך):**

השלב הבא הוא הפעלת מסנן LPF. זה חיוני להסרת רעשים בתדרים גבוהים שנוצרים כתוצאה מהדגימה בקצב גבוה של PDM. אנו משתמשים במסנן באטרוורת' מסדר 8 לצורך זה.

LPF הוא רכיב קריטי בתהליך ההמרה מ-PDM ל-PCM. תפקידו העיקרי הוא לסנן את התדרים הגבוהים מהאות, ולכן השם 'מעביר נמוך' - הוא מעביר את התדרים הנמוכים ומחליש את הגבוהים.

למה זה חשוב?

1. הסרת רעשים: אותות PDM מכילים רעשים בתדרים גבוהים כתוצאה מהדגימה בקצב גבוה. ה-LPF מסיר רעשים אלה.
2. מניעת אליאסינג: לפני הדצימציה (הורדת קצב הדגימה), חיוני להסיר תדרים גבוהים כדי למנוע עיוותים בתחום התדר הנשמע.
3. שחזור האות: ה-LPF עוזר לשחזר את צורת הגל המקורית מזרם הפולסים של ה-PDM.

בקוד שלנו, אנו משתמשים במסנן באטרוורת' (Butterworth) מסדר 8. מסנן זה נבחר בשל התגובה השטוחה שלו בתחום המעבר ודעיכה הדרגתית בתחום החסימה.

הפרמטרים של המסנן נקבעים כך:

**תדר הניתוק הוא 0.5 חלקי פקטור הדצימציה. זה מבטיח שנסנן את כל התדרים מעל מחצית מקצב הדגימה החדש.**

**סדר המסנן (8) קובע כמה חדה תהיה הירידה בתגובת התדר מעבר לתדר הניתוק.**

חשוב לציין שבחירת הפרמטרים הנכונים ל-LPF היא קריטית לאיכות ההמרה הסופית, ועשויה להשתנות בהתאם לאופי האות המקורי ודרישות היישום הספציפי.

**דצימציה:**

לאחר הסינון, אנו מבצעים דצימציה - תהליך של הפחתת קצב הדגימה. זה מפחית את כמות הנתונים אבל עדיין שומר על המידע החיוני של האות ולכן לא מאבד את האופי העיקרי שלו.

למה אנחנו עושים דצימציה ?

אנחנו רוצים להוריד את קצב הדגימה. הדבר הזה יאפשר לנו לעבור לאפיון של האות בתדרים גבוהים לאפיון באמצעות נמוכים וכך נוכל לבצע את ההמרה לPCM

זה גם מתקשר ליתרון הקודם שנאמר, אנחנו רוצים לחסוך במקום – הדצימציה מקטינה את כמות המידע שאנחנו צריכים לשמור. ככה המחשב יכול לעבוד על הצליל מהר יותר.

איך זה עובד? אנחנו בחרנו את פקטור הדצימציה להיות 64. כלומר אנחנו שומרים את המידע של האות כל פער דגימה של 64. בחירת הפקטור לא היתה קלה- בחרנו את הפקטור הזה לאחר התייעצות עם צוות המחקר, וצוות דצימציה ואינטרפולציה. לאחר ניסוי וטעיה, הצלחנו לבצע התאמה של זמן הקול בפלט שלנו, לבין הזמן הרצוי (קובץ השמע שהועלה ללמדה) , וכך ידענו שבחרנו בפקטור נכון עבור המשימה שלנו.

חשוב לזכור שלפני שעשינו דצימציה חייבים לסנן את הצלילים הגבוהים כדי שלא יאבד מידע חשוב ונקבל רעשים.

**בשלב זה נוצר לנו PCM**

עליו נבצע את ה..:

**נרמול:**

לאחר מכן אנחנו מנרמלים את האות PCM כדי להתאים אותו לטווח של מספרים שלמים באורך 16 ביט. איך זה עובד?

קודם כל, אנחנו מזיזים את הצליל כך שהממוצע שלו יהיה אפס (נדנוד סביב האפס).

אחר כך, אנחנו מגדילים את עוצמת הצליל כך שהערך הגבוה ביותר יהיה בדיוק בקצה של הטווח המותר שלנו, שנגיע הכי גבוה שאפשר.

לבסוף, אנחנו ממירים את המספרים למספרים שלמים כך שיהיה אות שמסוגלים לשמוע ולעבד בתוכנות.

למה אנחנו עושים את זה?

להתאים לטווח: אנחנו צריכים להתאים את הצלילים לטווח שהאוזן האנושית יכולה לשמוע ומחשב מבין (בדרך כלל מספרים בין -32,768 ל-32,767).

למנוע עיוותים: אם הצליל חזק מדי, הוא יישמע מעוות. אם חלש מדי, לא נשמע אותו טוב.

לנצל את כל הטווח: אנחנו רוצים להשתמש בכל הטווח האפשרי כדי לקבל איכות צליל טובה, למקסם את הביצוע.

נרמול בשפה קצת יותר פשוטה- זה כמו לקחת תמונה ולכוון את הבהירות והניגודיות שלה כך שתיראה הכי טוב על המסך שלנו. רק שאצלנו- זה עבור אות 😊

**המרה לקובץ WAV:**

לבסוף, אנו כותבים את האות המעובד לקובץ WAV באמצעות הפונקציה wavfile.write של SciPy. זה יוצר קובץ שמע סטנדרטי שניתן להשמעה במגוון רחב של מכשירים ותוכנות.

כל השלבים הללו מתבצעים ברצף בפונקציה הראשית pdm\_to\_wav, שמקבלת כקלט את קובץ ה-PDM ומייצרת כפלט קובץ WAV."

שקף רביעי: דוגמת הרצה (העלנו כסרטון בקובץ נפרד, יש גם קישור במצגת עצמה)

שקף חמישי: קשיים ביישום

תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי

בתהליך ההמרה מ-PDM ל-WAV נתקלנו במספר אתגרים משמעותיים. נסקור אותם:

1. חומר לא מוכר: PDM הוא פורמט שלא היינו מורגלים בו, ולא נלמד בהרצאה או בתרגול בזמן בו ביצענו את המשימה. לכן, היה עלינו ללמוד את עקרונותיו ואת הדרכים הנכונות לעבד אותו. זה דרש מחקר מעמיק והתנסות מעשית כדי להבין את ההתנהגות של האותות ואת הדרך הטובה ביותר לטפל בהם.
2. בחירת קצב הדגימה: קצב הדגימה של ה-PDM המקורי אמור להיות גבוה מאוד. היה עלינו להחליט מהו קצב הדגימה הסופי המתאים ביותר לקובץ ה-WAV. זו החלטה קריטית שמשפיעה על איכות השמע הסופית ועל גודל הקובץ. לאחר התיעצות עם צוות המחקר, וצוות דצימציה ואינטרפולציה, ובנוסף נסיון להסתכל על הפער בין זמן השמע שקיבלנו וזמן השמע שהועלה בלמדה- הגענו לפקטור הרצוי.
3. פקטור דצימציה: קביעת פקטור הדצימציה הנכון היה אתגר. פקטור גבוה מדי עלול לגרום לאובדן מידע חשוב, בעוד שפקטור נמוך מדי לא יפחית מספיק את כמות הנתונים. היה עלינו למצוא את האיזון הנכון.
4. **למה לא השתמשנו במסנן CIC: מסנן CIC (Cascaded Integrator-Comb) הוא פתרון נפוץ בהמרות PDM. למרות זאת, בחרנו להשתמש במסנן באטרוורת'. הסיבה לכך היא שמסנן CIC, למרות יעילותו החישובית, יכול לגרום לעיוותים בתגובת התדר, וזה לצערנו מה שגרם לנו. המסנן שבחרנו מספק תוצאות איכותיות יותר, אם כי במחיר של מורכבות חישובית גבוהה יותר. אך מאחר ומסנן CIC לא נתן לנו תוצאות תקינות, בחרנו לחשוב באופן יצירתי כמו מהנדסים אמיתיים, ולהשתמש במסנן חלופי.**

שקף שישי: מה למדנו?

**תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי**